Visual Analysis of Data for Criminal Investigation

Masaryk university,
Faculty of Informatics

Kristína Zákopčanová
Master thesis

2018

Declaration

I declare that this Master thesis is my original work and that I have written it independently. All sources and literature that I have used during elaboration of the thesis are correctly cited with complete reference to the corresponding sources.

Acknowledgement

First of all, I would like to thank my friend and superviser **Bára Kozlíková** for all she has done for me throughout my master studies. During this time, she has been my great inspiration and influence, not only in the academic field. She provided me with many opportunities that helped me grow, this thesis being just one of them. It was also thanks to her that I enjoyed the master studies so much.

I would like to thank my boyfriend **Jakub** for his endless support and patience with which he kept listening to me all these months when I spoke of almost nothing but my thesis.

I would also like to thank my great friend **Petr Vacek** whose design advice helped me make the thesis look as beautiful as I had dreamt of and who supplied me with sweets anytime I felt down when working.

Last, but not least, I am immensely grateful to my parents, **Marián** and **Lora**. The importance of thier support and love cannot be described in words.

Abstract

Visualization provides a way to effectively understand data, especially when dealing with big amounts of heterogenous data that form an inevitable part of the criminal investigation domain. It helps the criminal analysts discover both the expected and unexpected through the use of interactive visual representations of data.

This thesis forms a part of the research project Complex Analysis and Visualization of Heterogenous Big Data, funded by the Ministry of the Interior of the Czech Republice. Within the visualization part of this project, we focus on the design of a system for visual analysis of multidimensional data which are analyzed in criminal investigations. This thesis first reviews the existing approaches to visualization of such data. Then, it summarizes the requirements on the system for visual analysis, along with requirements on the network visualization which forms the core of the data representation in the proposed system. Based on these requirements, it proposes the high-level design of the system for visual analysis and detailed design of the network visualization. Last, it presents the prototype of the network visualization which presents the basic functionality of the proposed tool.

Keywords

information visualization, interactive visualization, network visualization, visual analysis, multidimensional data visualization, criminal investigation, big data, D3.js, force-directed layout

Contents

Introduction

Latest technology advancements allow us to collect large amounts of complex data from various domains. Since it often provides us with useful information, it is only natural that big data processing and visualization became a present topic that seeks for new approaches, which help us understand the data, especially the amounts which would not be readable without visual support. So, a lot of useful tools have been developed to analyze specific types of data or data from specific domains. However, sometimes it is necessary to analyze large amounts of data that come from various domains and that can be of any type. This is a challenge faced especially in the criminal investigation domain, where the analysts usually need to work with big amounts of heterogenous data that ask for efficient and complex analysis in order to draw necessary conclusions that help in finding relations in the data and in final decision making. Creating such a tool is a goal of the research project called Complex Analysis and Visualization of Heterogenous Big Data, further called Analýza, and that is tightly connected with this thesis.

As the title partly suggests, the primary goal of the project is to design a distributed system which allows for complex analysis of heterogenous big data for the purposes of criminal investigation.

The Analýza system consists of three main, interconnected components. Each component provides a specific functionality which is fundamental for effective complex analysis of data. The components are the following:

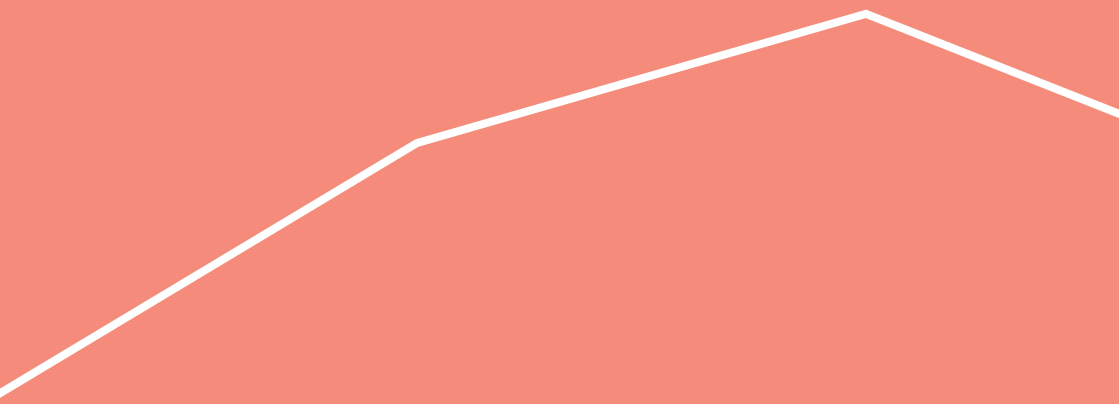
Data storage, ensuring a central data storage of the available data, their manipulation, such as addition or deletion of data entries, multicriteria selection of data, and definition of analytical operations on data.

Transformation and analysis of data, which is the main computational part providing a set of transformation modules executing analyses of data, or transformation of data stored in the data storage.

Data visualization, the key component for the presentation of data to criminal analysts in the form of a network visualization and providing extensive techniques and a set of specialized visualizations for visual analysis methods for multidimensional data that are analyzed during criminal investigations.

The goal of this thesis is to analyze requirements on the visualization component, design its architecture, and create a detailed design of the network visualization which forms the core of the visual analysis. The system needs to allow analysts to intuitively explore and analyze big data sets at various levels of cardinality and abstraction. Then, as a part of this thesis, a web-based prototype of the network visualization will be created.

The first chapter reviews the existing approaches to visualization of data related to criminal investigation with focus on network visualization. In the second chapter we present summary of the theory behind information visualization and design principles that ensure that the outcome visualization communicates information in an effective manner. The third chapter focuses on the analysis of requirements on the visualization component of the Analýza tool and proposes a design of its architecture and typical workflow. The fourth chapter reviews the requirements on the network visualization and presents a detailed design of data representation and interactions in the network visualization, which is the primary visualization in the system. Details regarding the implementation of the prototype are stated in the fifth chapter. The conclusion presents the results along with ideas and plans for future extensions that will be accomplished as a part of the Analýza project.



1 Related Work

In this chapter we discuss the existing approaches to visualization of data related to criminal investigations. We primarily focus on network visualization as this type of visualization forms the core of this thesis. First, we discuss several research publications related to visualization and visual analysis of network-based datasets. Then we list several related existing tools, along with their benefits and drawbacks in relation with our case.

1.1 Networks in Security Visualization and Analysis

Security analysis and visualization is a field with extensive research as the amount of data to be analyzed and their diversity is so vast that the visual support for analysis is a necessity. This is confirmed also by the fact that there are several events related to security visualization each year, the largest is the IEEE Symposium on Visualization for Cyber Security[1].

Van der Hulst (van der Hulst, 2009) published a general introduction into social network analysis (SNA) which facilitates understanding of criminal behavior through systematic analysis of criminal networks. It presents an overview of key concepts and applications of network analysis and concisely summarizes analytical purposes of social network analysis and its several aspects. It also provides a detailed description of building network visualization as a tool for investigative analysis. Then, it proposes a protocol draft for data handling and coding which requires careful planning, attention and accuracy, since any mistake made during the collation of data can obscure conclusions drawn from the analysis, and thus significantly influence the case resolution.

Another interesting study by Strang (Strang, 2014) was published as a part of book called Networks and Network Analysis for Defence and Security, which is a collection of articles that "discuss relevant framework and applications of network analysis in support of the defence and security domains." (Networks and Network Analysis for Defence and Security 2014). Strang discusses network analysis approaches applied in criminal investigations that are used to organize data, reveal patterns and relationships between data points, and identify culprits. He presents network visualizations as means for eloquent visualization of both qualitative and quantitative information and lists features that are useful in the visualization for the purposes of network analysis. The paper reviews properties of networks of criminals, such as patterns of self-organization and leaderships, or types of organizational structures in organized crime. It also describes in detail social network analysis concepts that offer

1 http://vizsec.org/

valuable input for investigation analysis, such as centrality, equivalence, density, strong and weak ties, and cut points.

Many proposed visualizations are tightly related with a specific domain or task. Gutfraind and Genkin (Gutfraind and Genkin, 2017) proposed a framework for visualizing the terrorist network on the 2015–2016 attacks in Paris and Brussels. The framework is based on the graph database theory and the analyzed relationships are visualized as network with different color coding for nodes. However, the proposed solution is suitable only for small datasets. Hughes et al. (Hughes et al., 2017) presented a network visualization enabling the analysis of Australian poly-drug trafficking networks. The proposed solution is highly specific to the input dataset and suffers from the same problem as the method by Gutfraind and Genkin – it is not designed to work efficiently for large networks.

Another type of existing solutions utilizes the combination of network visualization with other visualization types to a visual analysis tool. Tsigas et al. (Tsigas et al., 2012) proposed a visual analysis tool containing a network visualization designed specifically to overcome the scalability and comprehensibility of the existing solutions. They demonstrated the usability of the proposed tool on a large corpus of spam emails. The solution displays the network using abstract graph visualization. However, on the lower levels of abstraction, the method suffers from node and label overlaps. Similarly, Liao et al. (Liao et al., 2010) proposed a solution for large network visualization, related to the enterprise network security and management. Their solution is based on the hierarchical structure of similarity or difference visualization in the context of heterogeneous graphs. The similarity graphs are supported by the additional bipartite graphs showing the traffic flow. Harrison et al. (Harrison et al., 2010) proposed a system for interactive detection of network anomalies using coordinated multiple views. The network graph visualization is accompanied with the visualization of the spectral space, product of the

spectral analysis method. Selection performed in this view enables to interactively filter the graph representation which should be explored in more detail. The last part of the tool contains a time histogram, enabling to detect and select suspicious time ranges. Angelini et al. (Angelini et al., 2015) introduced PERCIVAL, a visual analytics environment enabling to monitor security events and understand the network security status.

Interesting alternative solution to visualization of network traffic for security administration was proposed by Ball et al. (Ball et al., 2004). The proposed solution enables to observe the communication patterns between home networks and external hosts.

1.2 Network Analysis Tools for Criminal Investigation

In this section, we present the most relevant tools used by the analysts which are able to operate with our type of data.

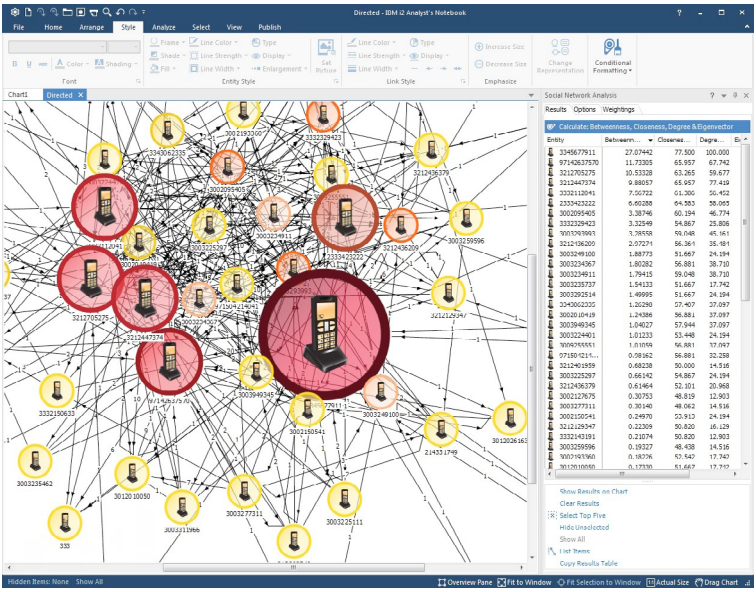
1.2.1 IBM i2 Analyst's Notebook

IBM i2 Analyst's Notebook 9[2] is a commercial tool for visual analysis of multidimensional data with focus on analysis of criminal networks and identification of key people and events. It provides visualization environment offering multiple analysis views into data, including network visualization, temporal and spatial views, or statistical ones, with goal to facilitate deeper understanding of data. It also provides integrated social network analysis capabilities to help reveal structures, hierarchy, and methods of operation of criminal networks.

Even though IBM i2 Analyst's Notebook is a very effective tool for analysis of multidimensional data, it has also few drawbacks. The major drawback is its computational complexity which limits the amount of data that can be effi-

2 https://www.ibm.com/us-en/marketplace/analysts-notebook

ciently analyzed in the tool. Unfortunately, it is quite common that criminal investigators work with big data. The tool also lacks options for effective connections with other systems. From the visualization point of view, the tool offers very poor visual representation of data with frequent overlays of displayed information, especially when analyzing bigger datasets, leading to unreadable visualization. It also provides a limited number of interactions which are often not very user friendly.



↑ Figure 1.1: Network visualization in IBM i2 Analyst's Notebook

1.2.2 Valcri

Valcri[3] is an Integrating Project funded by the European Commission that creates a system for visual analytics for sense-making in criminal intelligence. It is a semi-automated analysis system that helps reveal connections in data from mixed-format sources and displays its results in interactive, visualization-based user interface.

↑ Figure 1.2: Logo of the Valcri project.

Valcri aims to provide streamlined and effective visual analysis. It is based on cutting edge features like real-time search, historic knowledge extraction, analysis of textual and multimedia data, or crime situation reconstructions.

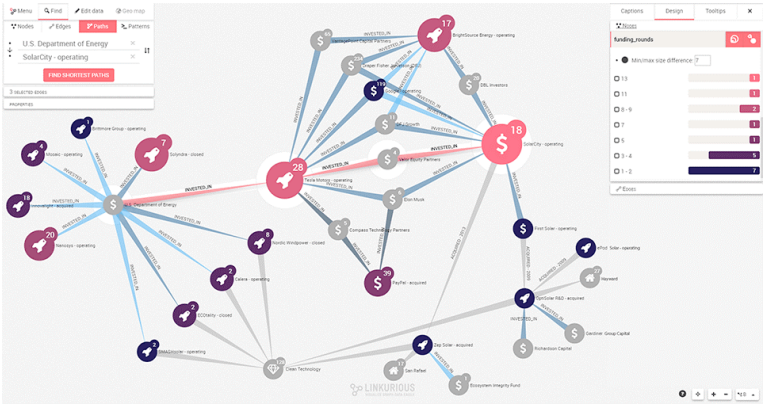
1.2.3 Linkurious

Linkurious[4] is a tool with a mission to democratize graph visualization. This software provides graph visualization techniques to allow to extract specific insight from graphs. They develop a product called Linkurious Enterprise and they offer the Linkurious SDK (Software Development Kit) that offers functionalities to build, test, and deploy own web applications for graph analysis. They put an emphasis to their visual representation which allows for easier orientation in displayed data.

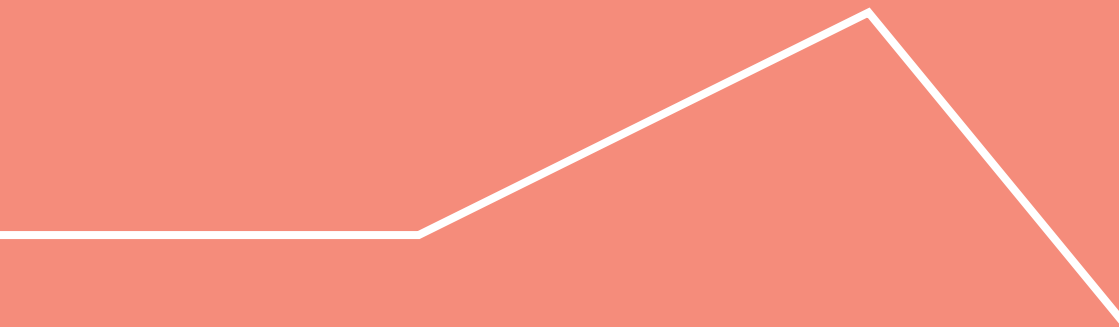
Figure 1.2: Logo of → the Valcri project.

3 http://valcri.org
4 https://linkurio.us

The Linkurious Enterprise solution provides visual analytics through graph visualization which can comprise large datasets of billions of nodes. It is also designed for multi-users collaboration, real-time exploration of data, and includes graph analysis methods to easily investigate connections and identify complex patterns. However, it does not offer any analysis for processing heterogenous data.



↑ Figure 1.4: Graph visualization from the Linkurious tool.



2 Information Visualization

Nowadays, most of visualization papers, articles, and books stress the fact that we live in data-rich world, feel overwhelmed with information and often do not know how to dive in and extract useful information without becoming lost. Though we would not think of contradicting such an obvious fact, we can look at the problem from a different viewpoint. As Stephen Few wrote:

> "We don't have too much information. Its quantity and rapid growth is not a problem. In fact, it represents a wealth of potential." (Few, 2009)

So, what makes one either feel overwhelmed with the amount of information or be comfortable and ready to use its potential? The answer simply depends on whether you can make sense of such big amounts of data or not. One of the ways to help you do it is to use data visualizations which provide a visual representation of information, and together with interactive manipulation and analysis can make big amounts of data suddenly accessible and understandable. Few also stated that:

> "What numbers could not communicate when presented as text in a table, which our brains interpret through the use of verbal processing, becomes visible and understandable when communicated visually. This is the power of data visualization." (Few, 2013)

In this chapter, our goal is to explain what information visualization is and present relevant topics for its design. We first discuss the definition of information visualization, then we present visual language and principles that are crucial for designing visualizations that communicate data accurately, intuitively and meaningfully. And lastly, we describe how interactivity allows for more effective visualizations which help us detect the expected and discover the unexpected.

2.1 Definition of Information Visualization

It is a difficult task to precisely define what information visualization is and what it is not, because its boundaries are often vague. However, there has been many attempts to define it and we would like to present to you few of them that we have found useful to provide you with as clear idea as possible.

To understand what information visualization means, we introduce other two terms commonly used in the context of visualization: data visualization and scientific visualization. The most general term, **data visualization**, describes any visual representation of data that supports exploration, examination, and communication of data. (Few, 2009) Then, based on the character of data, we can distinguish between two basic types of data visualization: information and scientific visualization.

Scientific visualization deals with visual representation of data that is usually physical in nature, such as X-ray, climate, or protein-protein interactions data. That means that creating a visual representation of such data is to some extent intuitive, since it is often based on the corresponding realistic physical form it takes in nature. This helps make the visualization easy to understand and thus allows for effective exploration of additional information present in data. (Few, 2009)

On the other hand, **information visualization** helps us to make sense of nonphysical information, such as financial data, business information, and abstract conceptions. We can generally call these “abstract data” in contrast to “scientific data”. However, since abstract data cannot benefit from having physical representation in nature, there is no obvious spatial mapping from data to visual terms. (Card et al., 1999)

Card, Mackinlay and Shneiderman define information visualization as follows:

> "The use of computer-supported, interactive, visual representations of abstract data to amplify cognition." (Card et al., 1999)

While all the characteristics are important,

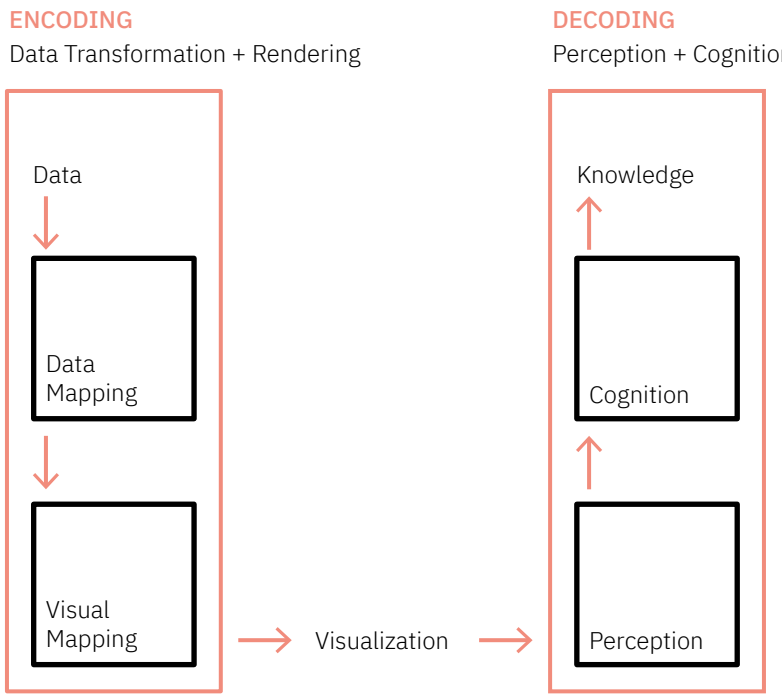
- the use of computers and visual artifacts helps the mind in the same way as cars enhance our ability to move in comparison to using only our feet,
- the interaction allows for manipulation with visualization, such as filtering of data or displaying details on demand,
- visual representation allows our perception system to process the information faster than textual representation and notice patterns, trends, or anomalies;

the last characteristics, amplification of cognition, describes the core goal of information visualization – to provide insight, which then allows for further discoveries, decision making, and explanations. (Card et al., 1999) (Few, 2009)

To be able to create effective visualizations that provide faithful insight into data, we first need to understand how the transformation of data into knowledge works.

2.2 Encoding and Decoding of Information

Generally, we can describe the process of retrieving knowledge from data through visualization as a two-stepped process. First, data is transformed (encoded) into visual representation, which is then interpreted (decoded) by human perception and cognitive system. The two-stepped process is also illustrated in Figure 2.1, which is inspired by Korsara (Korsara, 2018).



↑ Figure 2.1: Encoding and decoding of Information

Since reading visualizations relies on our perception and cognitive system, in other words, how we see and think, we need to find a visual language that can be used to encode data in such a way that it can be later correctly interpreted.

2.2.1 Visual Language

The first person to have laid the foundation of encoding information by visual representation was Jacques Bertin in his book Semiology of Graphics, first published in 1967 (Bertin and Berg, 2011). He studied how visual perception operated and formulated rules which describe how to express information visually so that it represents data intuitively, clearly, accurately and efficiently (Few, 2013). He formed a classification of all graphic marks and described in what way they can be used to express data visually. (Ware, 2013). Bertin also introduced **eight visual variables** of marks (Figure 2.2) that form the world of images and that a visualization designer can use. These are classified into the following three categories:

- **Plane variables:** Position
- **Retinal variables:** Size, Value, Texture, Color, Orientation, Shape
- **Motion**

Bertin also presented that each of these variables has different properties; therefore, each of them is suitable for displaying a certain type of information. Even though we have eight variables that are capable of displaying information in visualization, it is quite impossible to include them all in visualization at the same time. Otherwise, each of them would lose its expressive power and the resulting visualization would become too complicated and much more difficult to read. (Bertin and Berg, 2011)

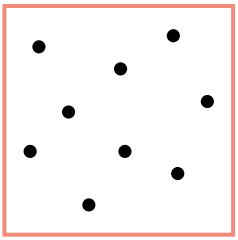
However, formulating a visual language is just the first step to designing successful visualizations.

2.2.2 Visual Perception

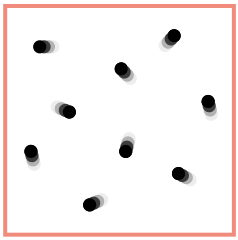
To build visualizations that effectively translate abstract data into visual language which is then correctly interpreted, we need to understand how visual

PLANE VARIABLES

Position

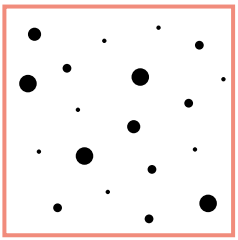


MOTION

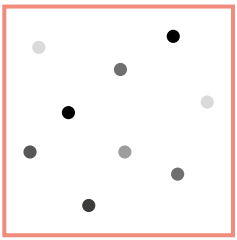


RETINAL VARIABLES

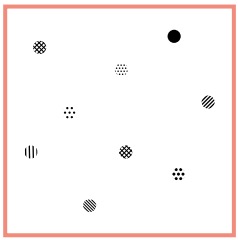
Size



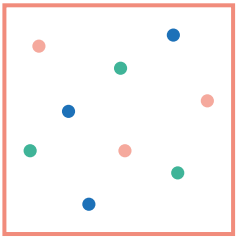
Value



Texture

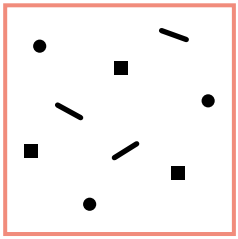


Color



Orientation

Shape



↑ Figure 2.2: Visual variables defined by Jacques Bertin

perception and cognition work and derive design principles from this knowledge. These then help to create visualizations with visual representation of data that is easily, efficiently, and correctly decoded.

2.2.2.1 Gestalt Theory of Visual Perception

One of the first studies of visual perception was introduced by the German school of experimental psychology at the beginning of the 20th century and is called **Gestalt psychology**. It studied principles of how human mind perceives pattern, form, and how it tends to organize visual objects into groups. We present six basic principles from the Gestalt psychology (Few, 2013). Each of these principles is also illustrated in Figure 2.3.

Proximity: Objects that are placed close together are perceived as a group.

Similarity: Objects that are similar in some visual attributes are perceived as a group.

Enclosure: Objects that are placed inside a boundary are perceived as a group.

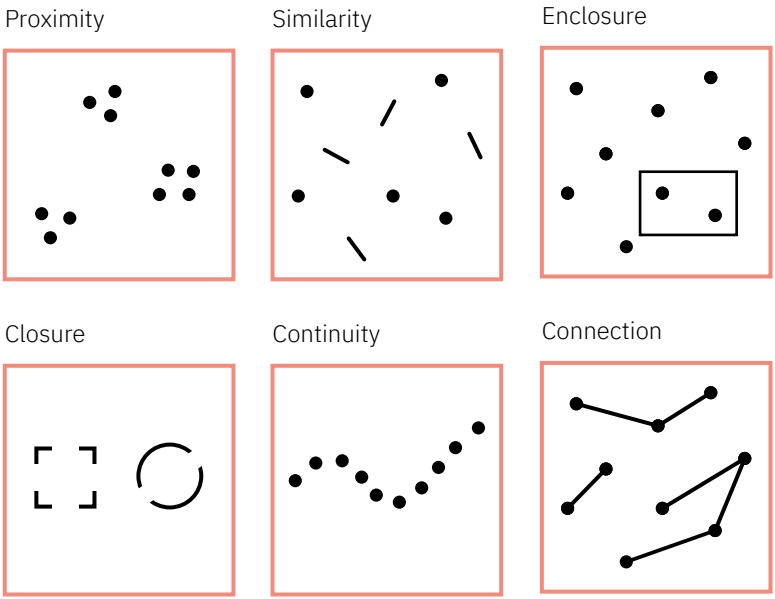
Closure: Objects such as shapes, letters, or pictures, are perceived as a whole, even if they are incomplete or some part is missing.

Continuity: Objects that are aligned along a path are perceived as a group.

Connection: Objects that are smoothly connected, e.g. by a line, or that are continuous are perceived as a group.

2.2.2.2 Preattentive Processing

Another principles used in visualization design are based on the speed of processing visual information in human perception system.



↑ Figure 2.3: Gestalt principles of visual perception

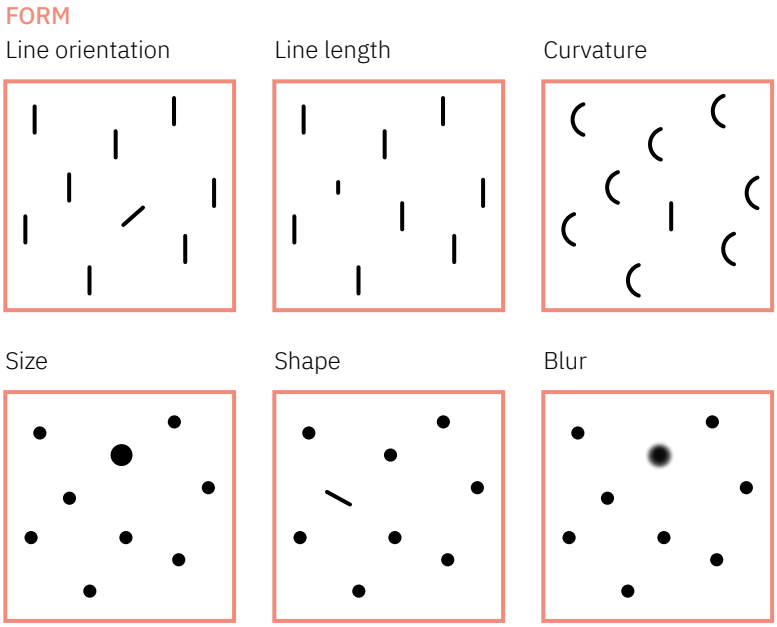
As Kosara et al. (Kosara et al., 2002) wrote, “Visualization is so effective and useful, because it utilizes one of the channels to our brain that have the highest bandwidths: our eyes. But even this channel can be used more or less efficiently. One special property of our visual system is preattentive processing.”

Preattentive processing of visual information happens automatically, prior to conscious attention, and it allows people to detect basic features of displayed objects, such as color, closure, contrast, or size. These are then joined in the conscious attention and form coherent objects.

Since preattentive processing happens very quickly and effortlessly, understanding what is processed preattentively and including these principles in the design of data visualization can greatly enhance its ability to communicate data intuitively and efficiently, and draw one's attention to the target by using unique visual feature. Generally, using preattentive processing usually leads to a more natural way of acquiring information. ("Preattentive processing", 2018) (Treisman, 1986) (Ware, 2013) (Healey, 2018)

Based on the classification of Colin Ware, we can organize the preattentively processable features into the following categories:

Form: line orientation, line length, line width, line collinearity, size, curvature, shape, special grouping, blur, added marks, and numerosity. (see Figure 2.4)



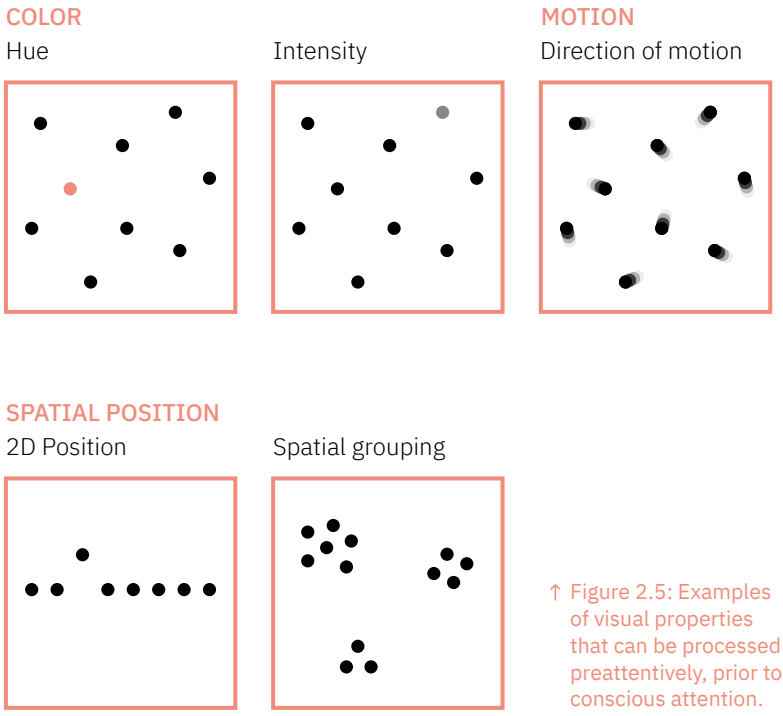
↑ Figure 2.4: Examples of visual properties that can be processed preattentively.

Color: hue and intensity

Motion: flicker and direction of motion

Spatial position: 2D position, and special grouping

Some of these features are illustrated in Figure 2.5.

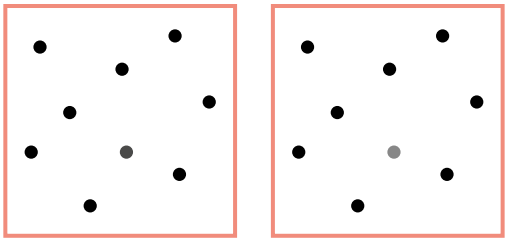


↑ Figure 2.5: Examples of visual properties that can be processed preattentively, prior to conscious attention.

However, in order to make something to be processed preattentively, there are two important factors that influence its effectiveness that need to be taken

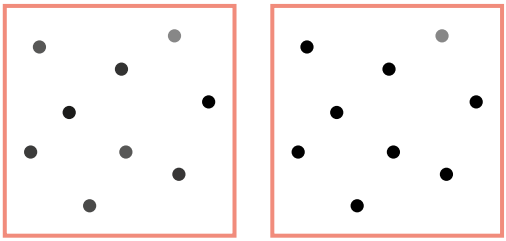
into consideration. These were introduced by Duncan, Humphreys and Quinlan (Duncan and Humphreys, 1989) (Quinlan and Humphreys, 1987).

1. **The degree of difference of the target from the nontargets.**
   Since this can be best explained visually, Figure 2.6 illustrates two cases, each presenting a different degree of difference. Image (a) displays the target object with little difference in intensity compared to the remaining, nontarget, objects, while image (b) displays objects with much higher difference in intensity. As we can clearly see, the second image can be much more effectively preattentively processed.



↑ Figure 2.6: The degree of difference of the target object from nontarget objects influences the effectiveness of preattentive processing.

2. **The degree of difference of the nontargets from each other.**
   Figure 2.7 illustrates how higher difference in preattentively processed property present in the nontarget objects complicates preattentive processing of the target object. In image (a) the difference in intensity is quite high, making it more difficult to notice the target object in the upper right corner, while in image (b) the target is easily noticeable since the difference in intensity among nontarget objects is very low.



↑ Figure 2.7: Effectiveness of preattentive processing is influenced by the degree of difference among the nontarget objects.

2.3 Visual Analysis

Having a static visualization, no matter how effective and rich its communication of data is, it can provide answers only to those questions it was designed for. However, often during the exploration of visualization, many questions that appear can be answered only by interacting with the data. An effective information visualization is built on two blocks: clear and intuitive data representation, and interactivity allowing to explore and understand information presented by the data.

So far, we have studied aspects we needed to understand in order to create effective, clear, and intuitive data representations. Now, we will reveal how visual analysis and exploration help to gain insight into data.

Visual analytics is a multidisciplinary field that focuses on extracting information and deriving insight from big multidimensional data, allowing to detect the expected and discover the unexpected (Thomas and Cook, 2005).

Thomas and Cook (Thomas and Cook, 2006) define following focus areas as parts of visual analytics:

- specific techniques providing deep insight, assessment, planning, and decision making
- visual representations and interaction techniques enabling the user to see, explore, and understand large amounts of information simultaneously
- specific data representations and transformations converting conflicting and dynamic data to representations supporting visualization and analysis
- techniques supporting production, presentation, and dissemination of results of analysis and convey them to a variety of audiences

2.3.1 Analytical Navigation and Exploration

The process of visual analysis is composed of many steps, decisions and possible paths that lead us to knowledge. A person who introduced us to the potential of data visualizations through exploration was John Tukey. He laid the foundation to a new statistical approach called **exploratory data analysis**. He also wrote a very apt description of data analysis:

> "Data analysis, like experimentation, must be considered as an open-ended, highly interactive, iterative process, whose actual steps are selected segments of a stubbily branching, tree-like pattern of possible actions." (Tukey and Wilk, 1966)

In exploratory visual analysis, there is no predefined way which is followed to gain insight into data. Rather, it is a natural process of navigating through visualization in an undirected way which is influenced by the data itself, searching

for answers to questions which appear during the analysis, and with only general guidelines at hand.

When navigating through visualization analytically, we can distinguish two approaches: **directed and exploratory analysis**. Directed analysis is applied when we search for answers to specific questions or we have certain hypothesis we want to confirm or disprove. On the other hand, exploratory analysis starts from no question or hypothesis, rather it is studying data generally, trying to find out whether there is something interesting or meaningful that we may explore further. Exploratory analysis often leads to directed analysis, once we notice an interesting pattern or anomaly and we start asking specific questions.

Any analytical navigation is possible thanks to a wide range of interaction techniques.

2.3.2 Analytical Interaction Techniques

Generally, there are many possible ways of interacting with data; however, some of them are especially useful for the design of tools for visual analysis. We will describe these in closer detail based on classification used by Few (Few, 2009).

Comparing: As we have previously discussed in the Section 2.2.2, a human mind tends to automatically look for similarities and differences. Comparison is a frequent, important and very effective way used in visual analysis. Therefore, information visualization should be designed for easy comparison of relevant patterns and data, and include means that allow to place information that we want to compare simultaneously on the screen.

Sorting: Even though simple, sorting is a very powerful feature that provides various views on the same data, it offers orderings of it based on chosen at-

tributes and thus assigning meaning to individual data entries compared to the whole. Moreover, it benefits from human mind's natural ability to compare values that are placed close together, therefore multiple resorting allows to relate one value to many others. In visualization, it is important to provide not only means for quick sorting of data in visualizations based on available attributes, but also to allow sorting of several visualizations of various data in the same way, thus allowing to see differences between different data sets.

Adding variables: Examining data through visualization may rise questions to which the answers are not available in the current visualization. Therefore, an option to display additional variables in visualization, or remove no longer needed ones, is essential for visual analysis.

Filtering: Filtering is a fundamental interaction technique of visual analysis, allowing for display and examination of only a subset of data that is currently needed and clear away unnecessary data, so it does not distract from the current task. Filtering is a very effective and powerful means of interaction, so information visualization tools should include extensive support for interactive controls for filtering (indirect interactivity) that are quickly accessible, easy to use and immediately adjust the visualization view. However, those tools should also support manual application of filters through manipulation with data representations in visualizations (direct interactivity).

Highlighting: Another way of leading our focus to interesting data is by highlighting it. In contrast to filtering, which displays only filtered data, highlighting allows a particular subset of data to stand out while keeping the context. This allows observing relationships between a highlighted subset of data and a whole dataset. It is also important to provide extensive support for highlighting, similar to the one of filtering, offering both direct and indirect means to highlight elements in visualization.

Highlighting a set of items not only in one graph, but in multiple simultaneous visualizations that share the same data set, is a very important interaction technique used in visual analysis, often referred to as brushing.

Aggregating: Aggregating and disaggregating information allows applying various levels of detail to data, offering thus either summarization or generalization, or more detailed information. This is very useful for adjusting visualization views so that they provide individual information at the levels of detail that is necessary for the current task of visual analysis. Aggregation and disaggregation can be again executed in direct and indirect manner, often in combination with filtering.

Remapping: Sometimes, it might be useful to provide a different view on data by expressing them in different units or percentage, or by changing their visualization, not only in the means of visual representation, but also by changing properties that are visualized. For example, instead of presenting values of two comparable variables, it might be useful to visualize the difference of these values.

Zooming and Panning: Zooming and panning are essential interaction technique of visual analysis. They allow studying individual parts of the whole visualization from a closer look. Zooming copies the functionality of magnifying glass and panning determines where exactly the magnifying glass is applied, that means which part of the whole visualization is scaled. These can be used either to allow study of individual values or range of values, or to provide more detailed information, when combined with various levels of detail.

Accessing details on demand: A key interaction technique of visual analysis tools is accessing details on demand. Often, a single view visualization cannot encompass all information that is available in data or the visualization's goal is to provide an overview of data. Technique of accessing details on demand allows visualizing only a subset of information by default, but anytime the de-

tails are needed, they can be instantly displayed. And once they are no longer necessary, they can be again removed.

Annotating: During visual analysis, it is often useful to be able to make notes and write down thoughts and questions related to the analysis. Annotating allows analysts to put notes directly into the visualization and to individual data entries, offering thus the most effective way of keeping notes related to the visualization. These notes can be either private to a user or serve as means for collaboration.

Bookmarking: The process of visual analysis does not follow any sequence of defined steps. Rather, it quite naturally tries to find answers to questions that may rise during the analysis itself. It often happens that we come to an interesting discovery and we want to save the particular view with all filters, sorts, aggregations and other mechanisms applied to the visualization, we may want to trace it back, or return to some of previous views. Bookmarking allows saving individual steps that were took during visual analysis and that brought us to the current view. We can then later access them, return, or just apply all the steps again to the same or distinct set of data. This feature is especially important in exploratory visual analysis, allowing to save multiple views, each providing an answer to a different question.

2.3.3 Shneiderman's Mantra

Now when we think of it, interactive information visualization and our perception system are a perfect combination for exploratory data analysis, since, as we have studied in previous parts, our eyes naturally notice patterns and anomalies, which are exactly the features we are looking for in exploratory analysis.

However, as usually, there are more and less effective ways to reach a discovery. Even though there is no correct way to navigate through information, Ben

Shneiderman formulated a simple, yet powerful, guideline for effective data exploration:

Overview first, zoom and filter, then details-on-demand.

Overview first is the first key step of visual analysis that focuses on detecting overall patterns and anomalies while reducing search. It helps analysts to choose their next step.

Zoom and filter is the second step of visual analysis. Once we have found interesting patterns, we can study them individually from a closer look. We can filter these to remove extraneous and distracting information.

Details-on-demand allows exploring the points of interest with further details that are stored in data, but often not explicitly displayed in the visualization or they are only represented approximately.

3 Visualization Component

As it was mentioned in the Introduction, the Analýza tool, designed for complex analysis of big data sets of multidimensional data, is composed of three components: (1) data storage, (2) transformation and analysis of data, and (3) visualizations. The visualization component could be seen as a face of the whole Analýza tool, providing to an analyst a visual representation of data sets from data store and presentation of results from the analysis component. However, above all, it is a tool for complex visual analysis of big data supporting the exploration, examination, and communication of data, helping the analyst in the decision making.

In this chapter, first the analysis of requirements on the visualization component is presented and then based on the requirements, the design for the component is introduced.

3.1 Analysis of Requirements

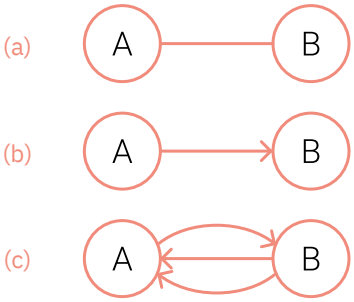
The first and inevitable step in every process of designing a new visualization tool is building knowledge of the domain it is intended for. In our case, it is the criminal investigation domain. To understand the domain as closely as possible, we combined various approaches, each providing a slightly different perspective, but together helping to gain more complex knowledge. We started with studying introductory literature into criminal investigation and exploration of existing tools. That helped us to create a better image of the domain and introduced relevant topics. Then we analyzed the structure of investigation data in detail. And lastly, but most importantly, we conducted many interviews with the future users of the Analýza tool, who helped us understand the value of their data for criminal investigation, learn their typical workflow of analyzing data in investigation, and identify advantages and drawbacks of their current approach. In the end, all of this together allowed us to form a list of key requirements on the visualization component.

3.1.1 Data

Data used for the criminal investigation contains all sorts of information, such as information about people, organizations, properties, telecommunications, financial transactions, or whole file systems.

This data is stored in the data storage in the form of data entries. Data entries either take form of an entity (for example **entity** representing a person, a phone device, or an address) or of a **link** representing a relationship between two entities (examples of types of links are a relative, ownership, or usage). Figure 3.1 displays possible types of direction of links.

The link can represent a relationship which is non-directional, or reciprocal (Figure 3.1 a). This means that both entities represent the same role in the relationships, such as “relatives”, or “friends”.



↑ Figure 3.1: Types of direction links.
(a) a non-directional link, also called reciprocal;
(b) a single-directional link;
(c) a multi-directional link.

The link can define a direction indicating which entity is a source and which one is a target of the relationship. For example, when a person A calls a person B, A is the source and B is the target of the link. This example is represented in Figure 3.1 b.

It is also important to note that between two entities, there can be multiple links of the same or distinct types. An example of such multiple link is when a person A called a person B, who then transferred some money to person A and called to person A to confirm the transaction. This situation is illustrated in Figure 3.1 c.

Both entities and links can have assigned an arbitrary number of attributes providing additional information about the data entry. Probably the most important attribute of each entity and link is the attribute called type which indicates what is the character of data, and thus predefining a set of expected attributes that the entity or link can have. For example, an entity of type person will most likely have attributes such as a name, last name, and date of birth. However, it is important to mention that none of the attributes is obligatory, not even the type attribute.

Apart from these attributes providing the additional information, each data entry also contains the corresponding case number and can contain a timestamp of data entry creation.

3.1.2 Typical Workflow of the Analyst

When data is analyzed for investigation purposes, it often needs to be studied from various points of view and contexts, each helping to reveal different connections and conclusions. As simple as it sounds in theory, in practice it is usually a lengthy process of analyzing various datasets from a number of sources with distinct data formats in a wide range of specialized tools. This leads to quite crumbled workflow. Analysts need to have a deep understanding of a multitude of tools and use them simultaneously in order to gain desired insight into data.

So generally, analysts lack a unifying tool where they could combine outcomes from various tools and analyze them in further detail. We could find an analogy in real word where we would not have general practitioners, only specialized doctors and there would be no one to combine all the specific information about our health and draw necessary conclusions.

The current crumbled workflow is one of the main reasons that calls for a new tool for criminal investigation. This tool would be designed to help facilitate the process of analysis in the way the data is stored, analyzed, both computationally and visually, and how it is then visualized so that it is an intuitive and understandable representation.

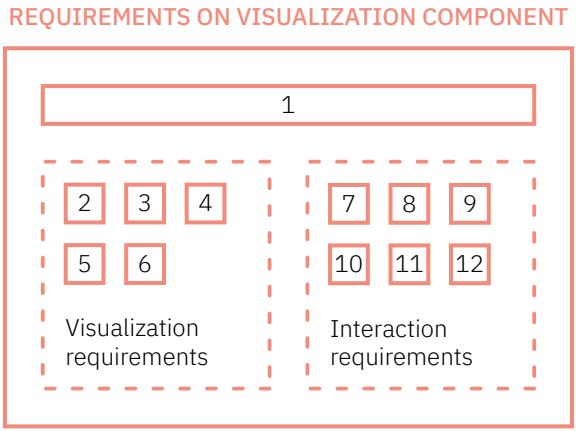
Therefore, the overall goal of the Analýza tool is to maximize the integration of the functionality the analysts need during criminal investigation by providing:

- central data storage,
- a set of data transformations and analyses helping to find connections difficult to draw from raw data and speeding up tedious data processing commonly carried out by humans (e.g., face recognition in photos),
- **a visualization tool allowing to effectively read and explore big data.**

The primary focus of this thesis is to create the design of the visualization tool.

3.1.3 Requirements on the Visualization Component

Based on the study of the current workflow described in the previous section, combined with the knowledge in the visualization domain and the sense of intuition, the following list and Figure 3.2 summarize twelve high-level requirements on the visualization component.



↑ Figure 3.2: Classification of twelve requirements on the visualization component.

REQUIREMENTS ON THE VISUALIZATION COMPONENT

1. **Multiple Views on Big Data and Multiple Levels of Detail:** Design of visualization component providing multiple views on data offering several levels of detail of presented information. These are adjustable based on the analyst's need and on the amount of displayed data. Available views range from the most high-level overview visualization displaying thousands to hundreds of thousand data entries, to low-level view presenting the details of an entry's attributes and relationships.

VISUALIZATION

2. **Network Visualization:** The primary visualization of the visualization component presents the entities and relationships between them and serves as an interface between the data storage and the analyst.
3. **Set of Specialized Visualizations:** A set of specialized interactive visualizations provides answers to specific types of queries and is interconnected with the network visualization, and optionally with other specialized visualizations.
4. **Spatial and Temporal Analysis:** A set of visualizations is designed for extensive analysis of data according to time or space with intuitive representation and focus on interaction with the timeline.
5. **Report Visualization:** Analysts can export analytical results presented in the appropriate context to a variety of audiences, such as the court.
6. **Intuitive and Understandable Data Representation:** Visualizations are designed in such way that their data representation is intuitive and understandable to the analyst. The data representation is derived from both understanding how human perception works, and visualization conventions used in criminal investigation.

INTERACTION

7. **Real-time Exploration:** Visualization component allows for real-time exploratory analysis of data with focus on data filtration and accessibility of specialized visualizations.
8. **Advanced Interaction Techniques:** Combination of navigational and interaction techniques facilitates orientation in visualizations and enhance the visual analysis.
9. **Local Data Manipulation:** Analysts can introduce additional information into a visualization by creating new data entries or modifying the

existing ones, and observe how these affect the visualization without affecting the primary data storage.

10. **Customization:** Analysts can customize the visualization component in terms of attribute mapping preferences, visualization settings, environment settings, template creation etc.
11. **Workspaces:** Each visual analysis takes place in a visualization document which stores current progress of analysis and all settings of the visualization component.
12. **Teamwork:** Analysts can share their workspaces, notes, and their modifications of data.

3.2 Design of Visualization Component

So far, we have looked at bits and pieces from the visualization theory, analyzed input data and formed requirements on the visualization component. At this point, we know everything we need to in order to create a design for the visualization component, which we will present in this section.

As we have already mentioned, the primary goal of the Analýza tool is to create a tool unifying functionality that the analysts usually need during criminal investigation. The same goal applies to the visualization component, which also aims at providing its functionality and features in an intuitive way which is designed with respect to analysts' needs.

3.2.1 Visualization Workflow

In the designed visualization component, an analyst's first step of analyzing any data stored in the central data storage is creating a new **case analysis**. A case analysis consists of its own local data storage and a set of **visualization**

documents used for individual analyses. The local data storage is filled with data from the central data storage based on the analyst's filter query. This data is then available for future visual and computational analyses in the case analysis and is shared among all case analysis' visual documents.

Based on a task the analyst wants to solve and the number of data entries, the visualization component offers a number of visualizations which are divided into three levels, each serving a different purpose and offering different views on data.

1. High-level visualization
2. Network visualization
3. Visualization modules

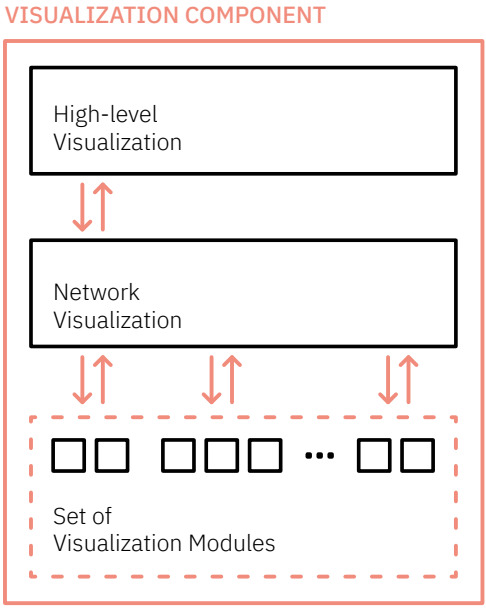


Figure 3.3: Visualization component diagram displaying three visualization levels and communication between them. →

The first two levels provide a general view on data through an interactive network diagram and are displayed in the visualization document. They offer a wide range of functionality for visual analysis. The last level, visualization modules, consists of modules containing specialized visualizations that can be invoked directly from the network visualization or from the menu of visualization component. Visualization modules provide additional insight into data and sometimes, their results can be also displayed as a part of the network visualization.

Individual levels of the visualization component and their communication is displayed in Figure 3.3.

3.2.1.1 High-level Visualization

Visualization that is usually used first from the visualization component is the high-level visualization. It presents big data in a network diagram in the most general and abstracted way. The high-level visualization can be seen in Figure 3.4.

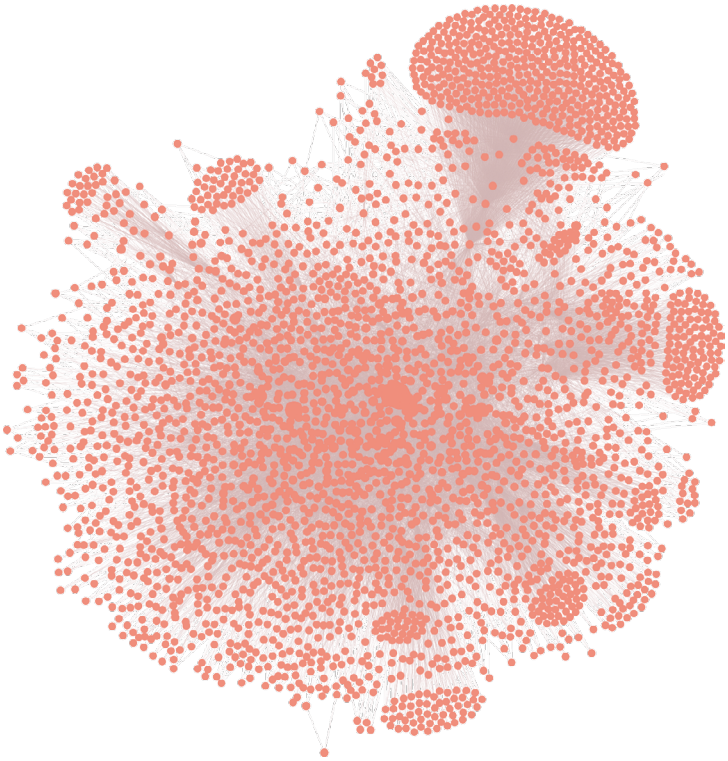
Its main purpose is to be able to comprise huge amounts of data, containing thousands to hundreds of thousands of data entries, and present structures that appear among them, such as communities, its centers, or global edges. This provides analysts a means to gain overview of data and thus helps them to choose a subset of dataset that is of interest.

For the high-level visualization, there is a set of predefined network analyses that can be called on the data and its results are then displayed in the high-level visualization. There is also an option to reduce the size of the visualization by decreasing the number of nodes and links displayed in it. Irrelevant nodes and links are removed from the visualization according to conditions that are depending on the type of applied analysis.

At this scale, node and link representations are as simplified as possible to allow our perception system focus on identifying structures. These structures

can help find suspicious activities, reveal key players, or just allow to explore data in more structured way.

Nevertheless, the visualization still offers details-on-demand of any data entry. These are visualized in object's overview card which are described in the Section 4.2.2.1.

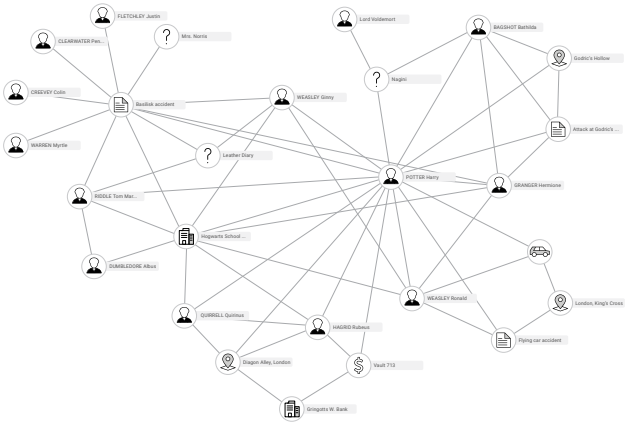


↑ Figure 3.4: Illustrative image of the high-level visualization (Bernhard, 2018).

With visualization's support for filtering, analysts can, based on structure analysis combined with filtering queries, gain overall insight into the data and identify a subset of data of interest, that they wish to explore further in more details.

3.2.1.2 Network Visualization

Once analysts have chosen a subset of data that are of moderate size and that they want to further explore, they can invoke the network visualization, which is illustrated in Figure 3.5.



↑ Figure 3.5: The network visualization example.

The network visualization is the primary visualization of the component which forms the core for visual analysis. It provides a concise and intuitive visual representation of data entries, allowing the analyst to interactively explore any information about data and study the relationships to others in detail as well as study properties of the whole network and its structure. It also offers a wide

range of functionalities that together form a powerful tool for effective visual analysis, such as advanced interaction and navigational techniques, manipulation of data, or support for customization.

The design of network visualization is described in detail in the Section 4.2, Design of Network Visualization.

3.2.1.3 Visualization Modules

When analysts explore data through the network visualization, they may need to look at the data from a different perspective, which is not available through the node-link point of view. That is why, as a part of the visualization component, we included a set of modules containing wide range of specialized visualizations which are interconnected with the network visualization. Visualization modules and their interactivity with the network visualization are designed to improve the visual analysis, help understand the context, and discover additional information, connections, and patterns in data.

Generally, visualization modules are designed with one of the two following goals in mind:

1. To provide a specific view on data that helps to find answers to analysts' questions. For example, visualization for telecommunication traffic which allows to easily explore characteristics of telecommunication. Often, the gained information can be saved to particular attributes of data entries, which can be later accessed from the network visualization.
2. To provide overview information that helps identify a subset of data that needs to be further explored, usually by using an external visualization tool that specializes on the given domain, such as complex space-temporal analysis. Providing such overview is often very useful,

since a common problem of external tools is their incapability to effectively analyze big datasets.

3.2.2 Visualization Document

When the analyst displays data through the visualization component, all his or her work takes place in a visualization document. As it was already mentioned, the visualization document allows to display data through two basic visualizations, offering a different level of detail, the high-level visualization, and the network visualization. The document stores a current state of all visualization settings, i.e., any change the analyst made to the visualization since the initial state, such as relocation of nodes, aggregation of nodes into groups, and other operations that influence the visualization view.

Visualization documents can be saved at any point and loaded later. They also provide a way to share the work with other analysts, as it will be described in the following section.

There is no limitation on the number of visualization documents that the analyst may create in a case analysis. Generally, this is very useful, especially for creating multiple views of the same data and comparing them together.

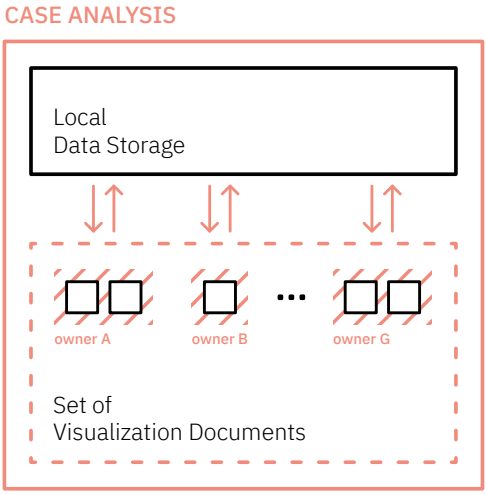
3.2.3 Teamwork Mode

Since it is a common practice that a case is investigated by multiple analysts, it is necessary that the design of the visualization component incorporates an option for collaboration. However, the work of an analyst requires their strong concentration and even a slight distraction may completely break the analytical process. The collaboration mode needs to take that into consideration and be as unintrusive as possible. Therefore, it does not take the form of a common collaboration mode where several people access the same file and work on it at the same time.

Our collaboration mode consists primarily of the following possibilities:

- Accessing visualization documents of collaborating analysts (read only)
- Creating a new visualization document as a copy of a collaborator's visualization document
- Displaying differences between two and more collaborators' visualization documents
- Displaying data modifications done by other collaborators and saving them to your visualization documents
- Creating working notes which are accessible by all collaborators. These can be at either of the following three levels:

    1. A case analysis
    2. A visualization document
    3. Data entries (nodes and links)



← Figure 3.6: Case analysis diagram.

4 Network Visualization

As we have already mentioned, the network visualization represents the core of the whole visualization component. It presents a tool that the analyst can use to access data from the data storage, which is represented in an intuitive and understandable way, and to explore the data from various points of view. The visual representation of data allows for much faster information processing and pattern recognition. By extension this helps us gain insight into the data that leads to better explanation and decision making. With the wide range of built-in functionality, the network visualization forms a powerful tool for effective and efficient visual analysis.

In this chapter, we first analyze the requirements on the network visualization, and then we present its design from the representation and interaction point of view.

4.1 Analysis of Requirements

When we analyze the requirements on the network visualization, we can generally divide them into two categories, abstract and specific requirements. Abstract requirements define high-level properties that the visualization should have. Some of these were already listed in the requirements on the visualization component, in Section 4.1.3. On the other hand, specific requirements describe exact tasks the visualization should be able to execute.

In this section, we will focus primarily on the analysis of specific requirements, however, it is important that the design of the network visualization is based on both categories. Abstract requirements on the visualization component that also apply to the network visualization are the following:

- Intuitive and understandable data representation
- Real-time exploration
- Advanced interaction techniques
- Data manipulation
- Customization
- Workspaces
- Teamwork
- Report visualization

The analysis of specific requirements on the network visualizations was conducted by simultaneously applying two approaches. In the first approach, we focused on analysis of a commonly used tool for criminal investigation, IBM i2 Analyst's Notebook 9. The second approach consisted of numerous interviews that we conducted with analysts and future users of the Analýza tool.

In the following part, we use the IBM i2 Analyst's Notebook 9 as a point of reference for defining requirements on the network visualization. Primarily because the tool is commonly used and represents data in network visualization. However, it is important to mention that we are not trying to create just a better version of the tool. The goal of our tool is to create a unified environment for analysts, and the network visualization is just one of many parts it will include. So, we reference IBM i2 Analyst's Notebook 9 as an existing solution of network visualization that we can use as an inspiration in those parts that we find useful. Then we add other features that are not part of this tool, but we would like them to be part of our visualization.

We present a concise list of requirements divided into three categories, based on functionality of the IBM i2 Analyst's Notebook 9 that is either (1) provided and convenient to include in our tool, (2) that is part of the tool but we would like to provide it in a different way, or (3) that the tool lacks and we would like it to offer to the analysts.

What Analyst does well

- Complex multicriteria filtering
- Creation of new nodes and links
- Values changes of object and link attributes
- View zooming on selected nodes and links

What Analyst does, but could be improved

- Node representation with a set of icons that is not visually consistent
- Access of information about individual nodes and links
- Focus and context together

- Support for annotations, such as highlighting or creating working notes, allowing for more effective analysis
- Undo and redo operations

What Analyst lacks and is useful to include in our visualization

- Node collisions avoidance
- Aggregation of nodes
- Labelled selections
- Storing the hierarchy of applied filters

4.2 Design of Network Visualization

Design of an interactive data visualization can be generally divided into two parts, a static one, defining how data are visually represented, and a dynamic one, defining how a user can interact with the visualization.

In the design of data representation, we usually answer questions such as:

- How should the data be represented? What is the most suitable visual mapping?
- What information needs to be always present?
- What information is important, therefore needs to visually stand out?
- What information can be simplified into physical attributes of vision, such as shape, color, or size?
- What information needs to be textually represented?

In the design of interactions, we provide answers to questions such as:

- How can users interact with the visualization?
- What operations can they apply to data?
- How can they access information currently not present in the visualization?
- Can they observe visual representation changes based on evolution of certain attributes, such as time?
- What analyses over data can they invoke?

4.2.1 Data Representation

A network visualization, sometimes also called a node-link diagram, is built on two key elements, nodes and links. A node represents entities from the data storage and a link represents a relationship between any two entities. In this part, we describe their visual representation.

However, before we dive into the detailed description of their visual representation, we would like to mention two principles we kept in mind throughout the design of the network visualization.

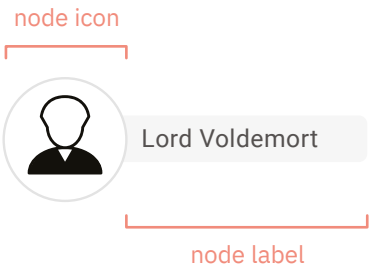
- **Make it as simple as possible:** This in practice means to use as little visual variables for the basic data representation as necessary. The remaining visual variables can be then used for interaction techniques and representation of additional information. This principle inspired us to create the basic layout of the network visualization only in grayscale.
- **Keep it as light as possible:** The second principle that played an important role throughout the design, was to use the values of visual variables wisely. By this we mean that once we have chosen a visual variable to represent a certain property, the value we assign to the variable

is very important and should be only as high as necessary. So for example, when we create a grayscale design of nodes and links, a choice to use only black and white colors would not be the best one, even though it can be perceived as simple as possible. If it was only black and white, already the basic view on the data would demand a lot of attention of our perception system, and quickly and easily tire us. That is why the basic design of nodes and links uses mostly light gray colors which still clearly indicate their boundaries.

4.2.1.1 It's All About Nodes

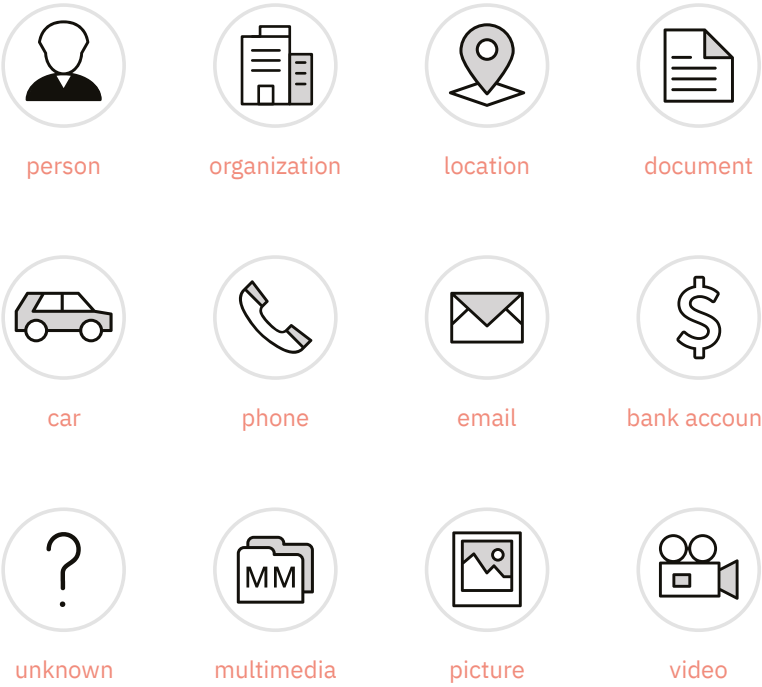
Each node is composed of two parts, a node icon and a node label.

A node icon is the smallest possible representation of any node. It consists of a circle outline containing an icon image which represents a type of an entity the node represents, as shown in Figure 4.1.



← Figure 4.1: Node representation, consisting of a node icon and node label.

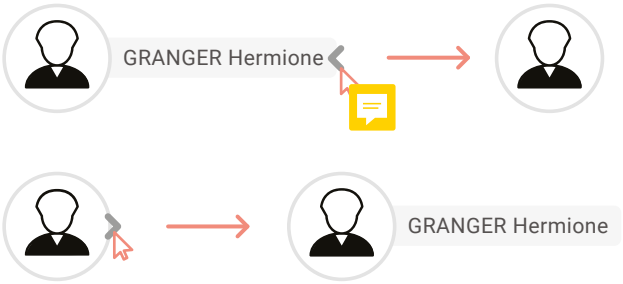
We have created a set of icon images for common entity types, such as a person, an organization, or a document. All icons were created based on a freeware icon set (Ozgur, 2018). We strived to create simple, line-based set of icons with similar visual weight, that means that all icons attract attention of a viewer with a similar intensity (see Figure 4.2). Though there is one exception to this rule and that is the icon representing the person because it contains black fill. Since entities of type person are usually the most important (as crimes revolve



↑ Figure 4.2: Basic set of icons that are frequently used in the network visualization.

around people), the icon is designed so that it visually stands out and can be thus preattentively processed first.

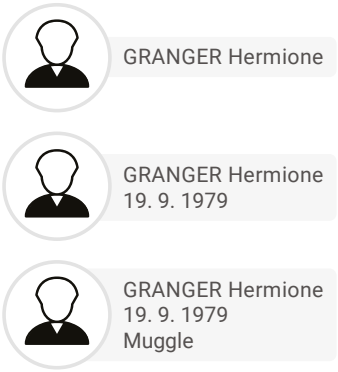
The second part of a node, a node label, contains a textual label identifying the node. The node label can be optionally hidden, as displayed in Figure 4.3. This can be useful either when having too many nodes and object identity is unimportant, or when an object is generally insignificant and we do not need to distinguish it from other objects of the same type, e.g. all employees of a company.



↑ Figure 4.3: Node labels can be hidden and opened by using arrow button which displays when user hovers over the node.

Since a unique identification of an object in the data storage is of a numerical value and this has little information value to an analyst, the textual label presents an information that usually serves as a clear identification of the given type of object. For example, a combination of the last name and first name serves as an identification of a person, or a vehicle registration plate of a car. Each entity type thus has defined an attribute (e.g., phone number) or a combination of attributes (e.g., last name + first name) that are used as its identification in the node label.

The lenght of the textual information displayed in the node label is limited by the width of the node label background. When the one-line text is longer, it is abbreviated and ellipsis (...) is added to indicate incomplete information. The position of ellipsis is based on the type of the abbreviated information and what part of it is the most important (e.g., my name can be shortened to "ZAKOPCANOVA Kri...", while a collapsed IP address "FE80::0202:B3FF:FE1E:8329" could be shortened to "...:B3FF:FE1E:8329".



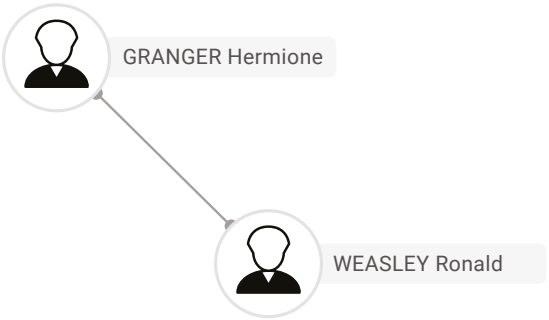
↑ Figure 4.4: Illustration of two- and three-lined node labels.

By default, the node label displays only the one-line identification information. However, the analyst can choose other attributes to appear inside the node label, usually, when these are necessary to be always visible at the first sight. Nevertheless, if it is not the crucial piece of information, it should not be displayed. Otherwise, the visualization would be easily overfilled with textual information, hiding the actual network structure and losing the power of visual data representation.

In the basic variants, we offer a one-, two- or three-lined textual label. These are presented in Figure 4.4.

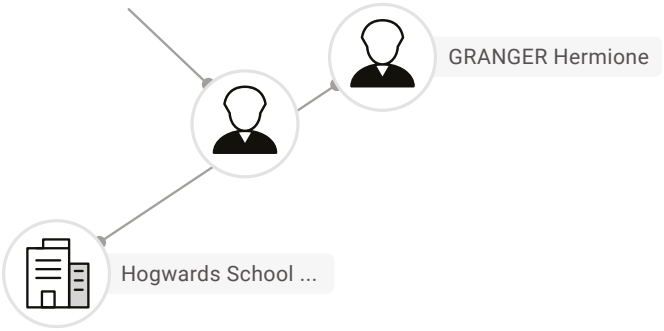
4.2.1.2 And Links Matter As Well

To express any connections between nodes in the network visualization, we use links which are represented by a simple line connecting two nodes (see Figure 4.5).



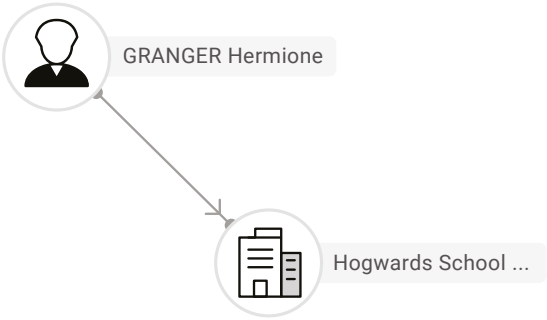
← Figure 4.5: Visual representation of a relationship between two nodes.

Since the network visualization can contain a large number of nodes and quite complicated structures, it can be difficult to visually connect a link to its two corresponding nodes. The difficulty can appear either because the link is too long and there is a lot of other links in its proximity making it difficult for eyes to track the link, or because a part of the link is occluded by another node and we then are not sure whether it just passes behind the node or whether it is



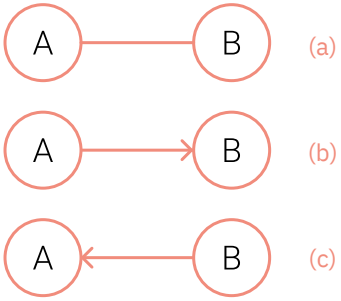
↑ Figure 4.6: Sticky paws help visually assign a link to the nodes it is connected with.

connected to that node. In order to help the analyst to connect the link to its corresponding nodes, we created a visual help that we call **sticky paws** of a link. Sticky paws are represented by two small circles, each one of them sticking to one of the two corresponding nodes, as illustrated in Figure 4.5. Sticky paws help to distinguish between nodes belonging to the link and nodes that only overlap the link. Example of such situation is depicted in Figure 4.6.



↑ Figure 4.7: Single directional link, indicating the source and the target of a given link.

Sometimes, a link can have defined direction of the relationship indicating what is the source and what is the target of the link. Therefore, we distinguish between two types of a single link, a nondirectional link and a directional link. The only difference of the directional link is that we use an arrow pointing in the direction from the source to the target, as we can see in Figure 4.7. The arrow is by default displayed next to the target node, thus between two nodes, there exist three types of a single link, which are displayed in Figure 4.8.



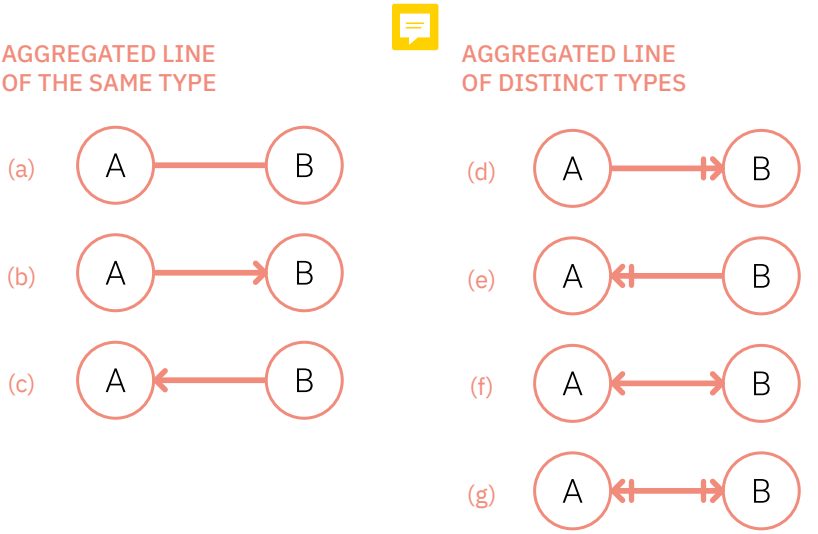
← Figure 4.8: Illustration of different types of link directionality:
(a) non-directional link,
(b) directional link, node A is the source and node B the target,
(c) directional link with node B as the source and node A as the target.

In addition, it is quite common that between two nodes, there are several links. The number of the links can be high, especially for certain types of links. For example, telecommunication traffic can contain a big number of links representing just phone calls, and there can be tens to hundreds of them. So, rather than representing every link individually (which would soon lead to crammed, difficult-to-read visualization), we represent all links between two given nodes as one aggregated link, called multilink. A multilink is visually distinguished from single links by using thicker line for its representation

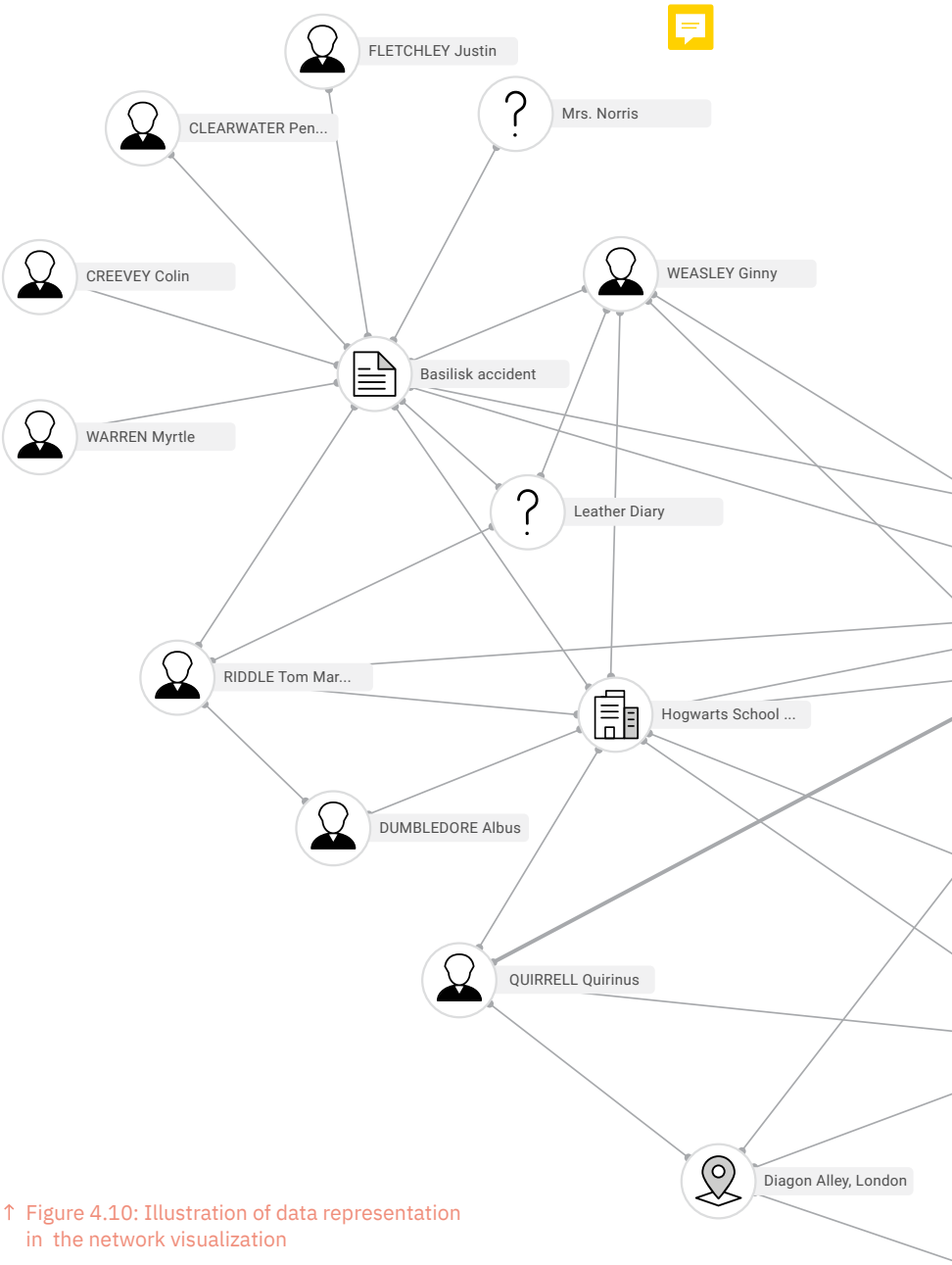
In Figure 4.9 we illustrate all possible types of multilinks. However, we would like to point out that when a multilink represents links with distinct types and one of them is nondirectional, we represent it by adding a simple line perpendicular to the link.

Data entries of links usually contain also a description of the link, but since these can be often very long and are not generally necessary to be visible all the time, they are by default hidden in the visualization. Nevertheless, when analysts need to have certain link labels explicitly visible, they can have them

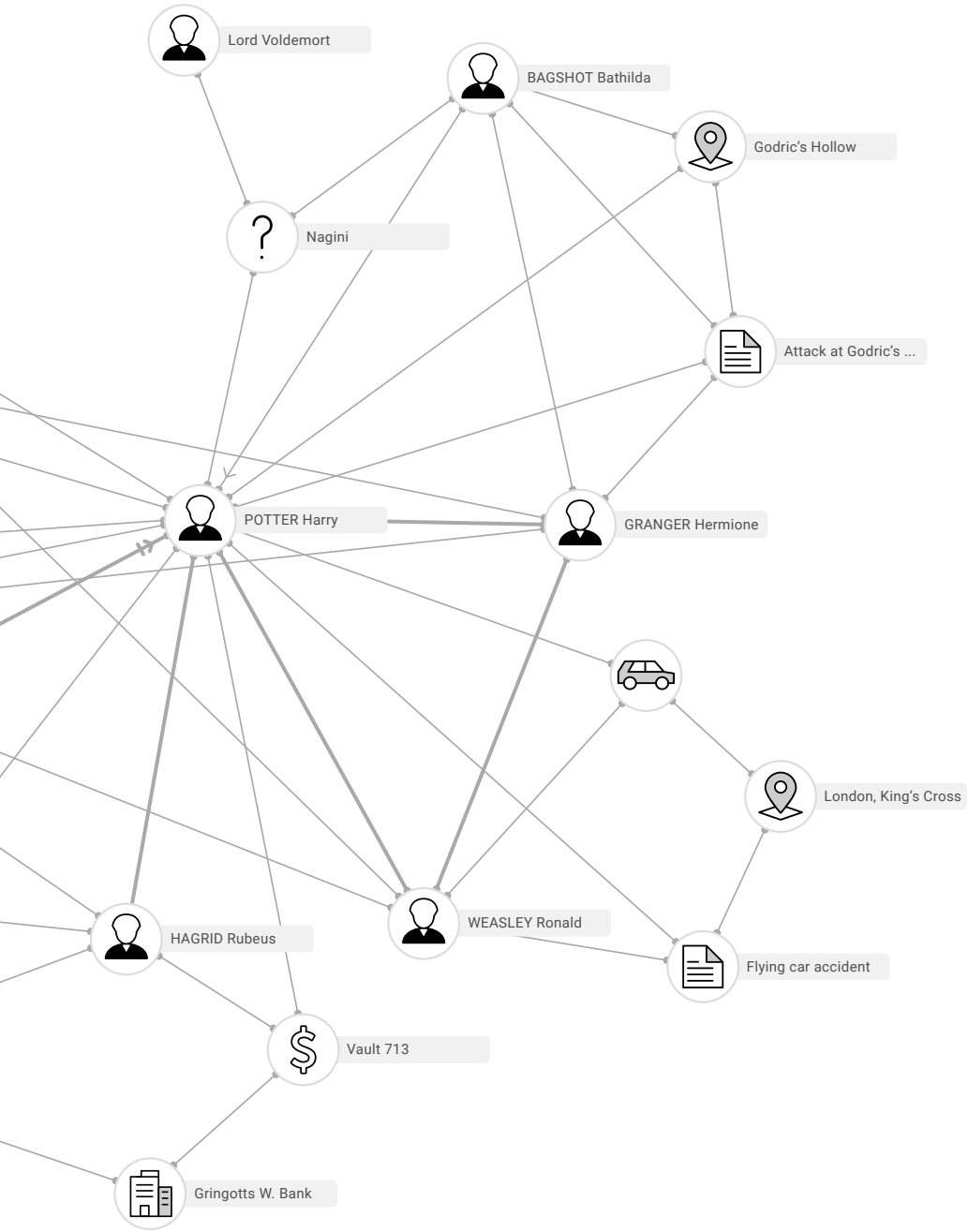
displayed on demand. However, we strongly discourage from having all available link descriptions displayed, since it can lead again to crammed and unreadable visualization with either frequent overlaps, hiding other information, or too big dimensions.



↑ Figure 4.9: Representation of possible types of multilinks.
We visually distinguish multilinks aggregating:
(a) only non-directional links,
(b) only directional links from node A to node B,
(c) only directional links from node B to node A,
(d) non-directional and directional links from A to B,
(e) non-directional and directional links from B to A,
(f) directional link from both A to B and B to A,
(g) non-directional and directional links from both A to B and B to A.



↑ Figure 4.10: Illustration of data representation in the network visualization


Lord Voldemort
BAGSHOT Bathilda
Godric's Hollow
Nagini
Attack at Godric's ...
POTTER Harry
GRANGER Hermione
London, King's Cross
HAGRID Rubeus
WEASLEY Ronald
Flying car accident
Vault 713
Gringotts W. Bank

4.2.2 Interaction

As we have seen in the previous part, data representation in network visualization takes a concise form, displaying the most necessary information. However, it does not present all information available in data. So if the presented data representation was the only representation available to the analyst, we would not use the potential neither of the data, nor of the visualization at all.

What makes visualizations a powerful tool for visual analysis is their wide support for interactivity. It allows for analytical navigation, real-time exploration of data, easier orientation in visualization, and interactive manipulation with the data and the visualization.

In the design of the network visualization, we included the following interaction techniques:

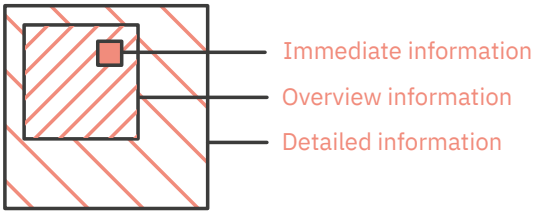
- **Local exploration of data** to access any detail about nodes and links saved in the data storage.
- **Focus and context together** to provide a quick visual overview of neighboring nodes, and facilitate analyst's orientation in the visualization.
- **Selections** providing an effective way to define selections of nodes and links.
- **Data manipulation** to create new nodes and links, including thus new information into the visualization.
- **Node aggregations** to organize nodes into groups, or hide them from the visualization.
- **Annotations** to include analyst's knowledge, observations, and questions in the form of tags, highlighting, and notes.
- **Bookmarking** to save a particular view, including filters, sorts, aggregations, and other interaction features, so that we can return to it later.

4.2.2.1 Local Exploration of Data

A goal of the local exploration of data is to provide a way for the analyst to access all available information regarding nodes and links. It is obvious that with such big data, it is impossible to display all available information at once, because it would make the visualization unreadable.

Since the importance and value of each information may differ, and so does the frequency of how often it needs to be accessed, we propose three levels of detail of the presented information. Individual information about nodes and links is assigned to one of the three levels based on their importance. There is a simple hierarchy structure among the three levels, which is illustrated in Figure 4.11.

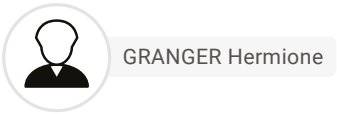
So, the analyst can start from the most important information, which is the fastest accessible one, and only dig deeper for additional information on demand, when necessary.



↑ Figure 4.11: Venn digram displaying the hierarchy of individual levels of provided information.

Instant information

Instant information provides minimal and crucial information that usually clearly describes an entity or a link between entities. Node's instant information is presented in the node label, as illustrated in Figure 4.12. Link's instant information is displayed in the link label, however, it is by default hidden.



↑ Figure 4.12: Node's instant information is displayed in the node label.

Overview information

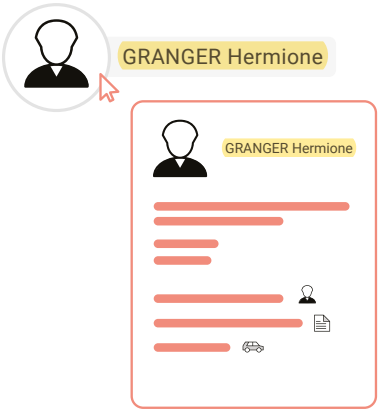
The second level of detail provides, as the name already suggests, an overview of important information. It is displayed in an overview card which is accessible by hovering over a selected object (see Figure 4.13).

Node's overview card usually contains:

- Important node's attributes (e.g., a person's name, birthdate, birthplace, and nicknames)
- List of all links connected to the node
- List of neighboring nodes, that means nodes that are connected to the node by a link
- Working notes associated with the node

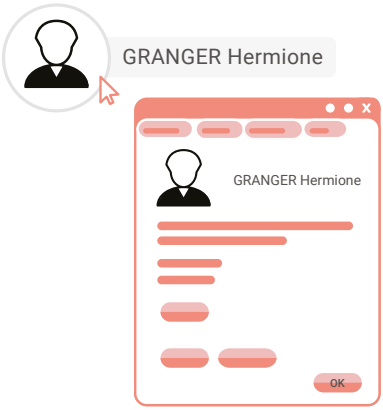
Link's overview card usually contains:

- Important link's attributes (e.g., type of the link, such as financial transaction, and amount of transferred money)
- Two nodes connected by this link
- Working notes associated with the link



← Figure 4.13: Node's overview information is displayed in the node's Overview card, which is accessible when hovering over the node.

Apart from the textual information, overview cards can also include some basic operations that can be executed over a node or link based on its type. In the future, we would also like to include some small and simple visualizations or statistics related to the node or link and their surroundings.



← Figure 4.14: Node's detailed information is displayed in the node's Detailed card, which is accessible on double click on the node.

Detailed information

In the last level of detail, the analyst can find everything that is known about the node or link, including all available operations defined on them, as well as individual settings for their display in the visualization. A detailed information is displayed in a detailed card (see Figure 4.14) which is invoked by double clicking on a node or link.

4.2.2.2 Focus and Context Together

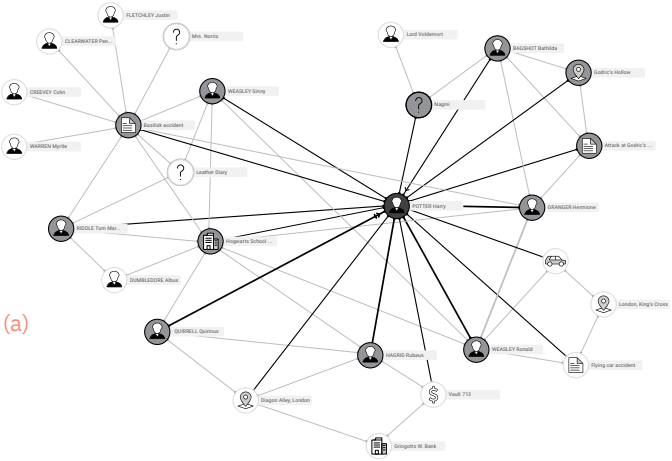
As network visualizations usually contain a high number of nodes and links which tend to cross over a lot, they are often difficult to visually orient in. Well, not unless there are just few nodes with few links or simple structure in data. Unfortunately, this is not our case. In criminal investigation, the network visualization commonly contains tens to hundreds of nodes linked together in an unorganized manner. We already introduced sticky paws, a means for helping the analyst to distinguish between a line connected to a node, and a line only intersecting a node, but not being connected to it.

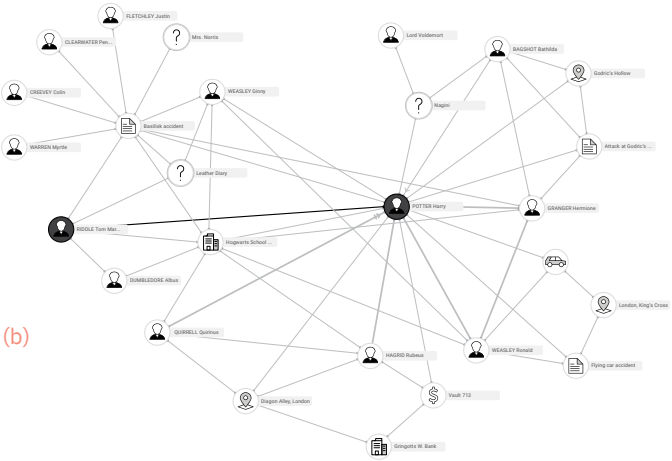
This technique is designed with two goals in mind:

- to provide a quick overview of connections of individual data entries,
- and to help find connected links and nodes, or neighboring nodes in the visualization.

This technique can be very useful at any moment of visual analysis; therefore, it is important that the analyst can use it easily. It can be applied by simple hovering over a node or a link and pressing a dedicated key (currently Ctrl key).

When a mouse hovers over a link, it highlights the link and the nodes it connects, as illustrated in Figure 4.15 a. While when a mouse hovers over a node, it highlights all links coming out of the node and all nodes connected by those links. This is shown in Figure 4.15 b.



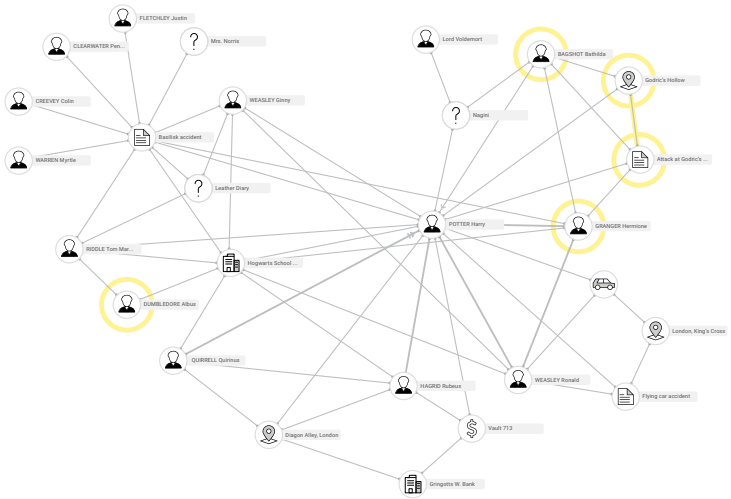


↑ Figure 4.15: Illustration of focus and context techniques
(a) illustration of link highlighting, (b) illustration of node highlighting.

4.2.2.3 Selections

A powerful, and most likely inevitable part of visual analysis, is the selection technique providing a way to define selections of links and nodes, typically based on advanced filtering. This technique is key in identifying sets that share some properties and is also the base for many operations that can be performed on groups of nodes or links, or generally, on the whole visualization.

Analysts can choose a group of nodes and links directly, either by selecting objects individually using their mouse, or by using the lasso tool, or they can choose objects indirectly, through advanced filtering. The set of selected nodes can be then given name and saved as a **labelled selection** in the visualization document.



↑ Figure 4.16: Illustration of the selection technique in the network visualization.

Advanced filters allow the analyst to define a filter query which can be composed of several filter conditions. An example of such query could be “Select all **nodes** of type **account** which has the **balance** smaller than zero and the **total sum of transfer payments** higher than one million euro”. The filter query is defined in a Filter card which lists the options the analyst can choose from. The analyst successively defines filter conditions with a set operator, such as union, intersection, or difference, that is applied to combine the results of the filter conditions. A filter condition definition starts from selecting an available attribute, then based on its type a selection of possible comparative operators is offered and an input text field to specify the value the attribute is compared to.

The selection of nodes and links is visually highlighted in network visualization by light outlines, as displayed in Figure 4.16.

4.2.2.4 Data Manipulation

An important part of a system for visual analysis is allowing the analyst to manipulate data that is displayed in the network visualization. This can be very useful, as it is common that throughout the process of analysis, new pieces of information appear, and these may influence the conclusions drawn from the analysis. Another important feature of data manipulation is to allow the analyst to mark selected data entries as non-relevant to the current analysis, so that these are not displayed and do not cause distraction from the relevant data. However, it is important to mention that these data manipulations take place only in the local data storage of the corresponding case analysis, they are not reflected in the central data storage.

As we have indicated, the analyst can add new nodes and links to a visualization and specify any of their attributes. Nonetheless, it is necessary to distinguish between data entries loaded from the central data storage and those created in the case analysis. Therefore, their visual representation in the visualization is distinct. While nodes and links loaded from the central data storage are drawn

with a full stroke, new nodes and links created by the analyst are represented by a dashed stroke, as you can see in Figure 4.17.

↑ Figure 4.17: Sketch of the visual representation of a new node and new link.

Each new data entry is stored together with an identification of the analyst who created it and a timestamp of creation. The collaborating analysts can then choose to display newly created data entries created either by specific collaborators or during a specific time.

4.2.2.5 Node Aggregations

During the exploratory analysis of data, aggregating a set of nodes is a very useful and important feature. It allows changing structure of the data, introducing logical structures, or can lead to a simplified view.

For the purposes of criminal investigation, we defined three types of aggregation that can be applied in the network visualization to enhance visual analysis and that are naturally reversible operations.

1. Alias aggregation
2. Bundling
3. General group

Each of these types are created for specific use cases and have a distinct visual representation. They are described in closer detail in the following parts.

However, before we focus on the description of individual types, we need to introduce a new term. So far, we have talked only about nodes which represented a single entity. But now, when we can aggregate several nodes together, we need to be able to distinguish between a node representing one entity and a node representing more entities.

- **A regular node:** A node which represents one entity.
- **A group node:** A node which represents multiple objects, both entities and links between them.

It is important to say that only the third type of aggregation, that is the general group, will produce a group node.

Alias aggregation

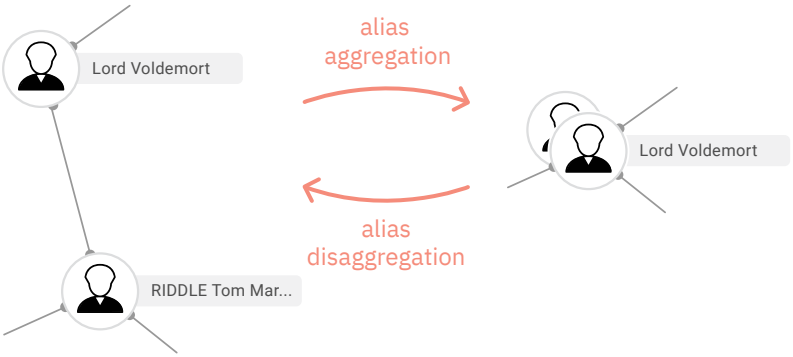
Alias aggregation is a type of aggregation which is designed for a special case, when analysts suspect, or have a proof, that two entities in the data represent the same object in the real world. In such case, they can apply alias aggregation to these nodes and create a new node as their combination.

Alias aggregation takes in two regular nodes and produces a new regular node with merged attributes and links of the original nodes. Any conflicts between merging attributes are displayed to analysts who solve them by either picking

↑ Figure 4.18: Sketch of the visual representation of an alias node.

one of the pieces of conflicting information or combining them together. The reason why the resulting node is not a group node, but a regular node, is that it is supposed to represent only one entity in the real world, even though it is technically created from merging two entities in the data.

Even though the resulted node is a regular node, it still needs to have a special visual representation, which clearly states that this node is created thanks to modification of the original data from the local data storage. The representation of alias aggregation can be seen in Figure 4.18, while Figure 4.19 presents an example from the network visualization. Different visual representation generally also indicates that there may be special attributes or operations connected to the node, such as the operation of reversing the aggregation, breaking it down back into the original data.

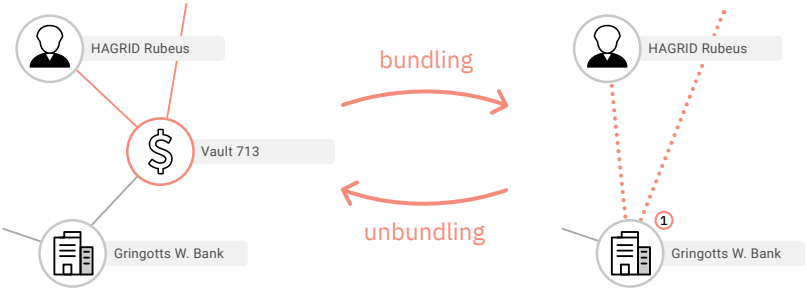


↑ Figure 4.19: Illustration of alias aggregation.

Bundling

The second type of node aggregation is called bundling. It is used for hiding currently unimportant nodes inside a neighbor node when there is logical or hierarchical connection between them. For example, when a person owns a car and the car is irrelevant for the current analysis, we can hide its node below the person who owns it. The person becomes a representative node of the car. When a node is representative, a little bubble appears next to its node icon with a number indicating how many nodes are currently hidden inside the node, as illustrated in Figure 4.20.

↑ Figure 4.20: Sketch of the visual representation of the representative node with bundled (hidden) nodes inside of it.



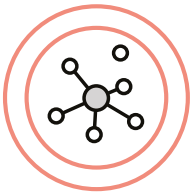
↑ Figure 4.21: Illustration of bundling technique.

Bundling aggregation takes a regular node or a set of regular nodes to be bundled and a regular node to become their representative node. As a result, the bundled nodes are removed from the visualization and the bubble appears next to the representative node.

Links that were originally connected to the currently bundled nodes are transferred to the representative node. We call these links **transferred links** and they are represented by a dotted line, as illustrated in Figure 4.21.

General group

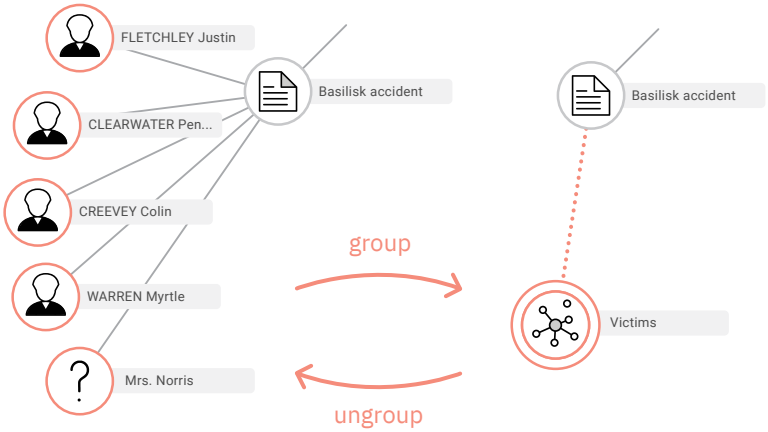
The last type of aggregation represents the most general one. It is not designed for any special case, it is just plain grouping of any nodes that do not even need to be connected by a link. Examples of such groups could be a group of Drug smugglers, or all nodes related to a specific event like a Drug smuggling to Ljubljana.



↑ Figure 4.22: Sketch of the visual representation of a grouped node.

To create a group, the analyst selects a set of nodes, both regular and group nodes, chooses a label, and optionally a type of a group (e.g., event, or group of people). The analyst can also specify the group node's attributes. The new group node differs visually from a regular node only by having double-lined outline of a circle in the node icon. The distinct visual representation, along with an illustrative example from the network visualization, are represented in Figures 4.22 and 4.23.

All links that existed between the currently grouped nodes are removed from the visualization, while links that existed between the grouped nodes and the remaining nodes in the visualization are again transferred to the new group node. As usually, transferred links are drawn by a dotted line.



↑ Figure 4.23: Illustrative example of grouping operation

4.2.2.6 Annotations

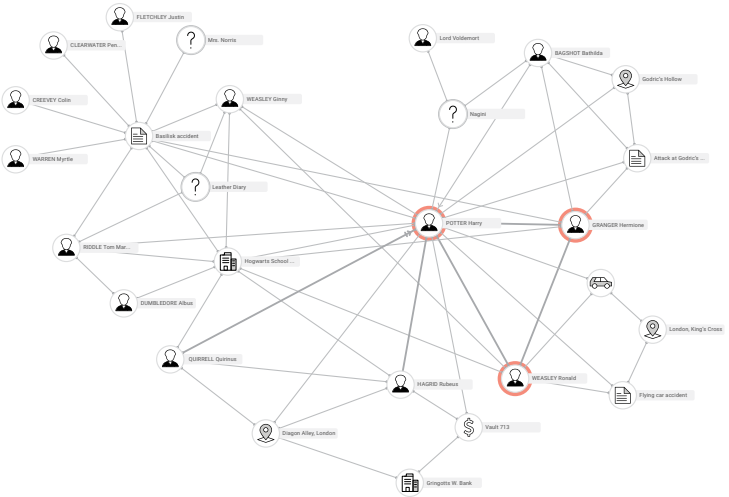
Annotation techniques offer ways for analysts to include their knowledge, observations, questions, or just regular working notes into the network visualization. Annotations enhance the visual analysis in the form of textual notes, tags and colors, or highlighting.

We will present three types of annotation techniques: focus, tags, and notes.

Focus

← Figure 4.24: Sketch of the visual representation of a focused node.

Throughout visual analysis in criminal investigation, analysts usually try to identify key entities that played some role in the investigated crime. To do so, they perform exploratory analysis of the data. It is quite natural that some entities and links are more interesting or relevant to the case analysis than others, however, once they find an especially interesting or suspicious entity, it is useful to mark it appropriately. And this is exactly what the focus technique



↑ Figure 4.25: Illustration of visibility of focused nodes in the network visualization.

provides. It helps to bring visual emphasis to those objects that are considered the most relevant for the case analysis. Thanks to strong visual emphasis, the focused objects are always easily detectable in the visualization. At the same time, they serve as a kind of visual anchor for the data exploration, which facilitates the orientation in the visualization. The visual representation of the focused node and its visibility in the network visualization are shown in Figures 4.24 and 4.25.

Tagging

← Figure 4.26: Sketch of the visual representation of a tagged node.

Another option for visual highlighting is provided by tagging. While the focusing technique targets at the key objects, tagging provides more general means to mark objects with interesting properties that are relevant to the current analysis. Tagging allows the analyst to create a tag and assign it to any nodes and links. These are then visually highlighted by colored dots appearing next to the node in the color of the tag , as depicted in Figure 4.26.

Tags can be of any nature, for example, it can be based on information from data, such as visually marking all victims or convicted in past, or it can be used just as working notes of an analyst, for example to mark all objects that are missing some information or that are suspicious.

Information which is encoded by the tag is quickly accessible during visual analysis. Since it encodes the information that is in some way relevant to the current analysis, it facilitates the visual analysis and decision making.

However, a tag's goal is not to visually attract analysts' sight and allow them to find the tagged node or link, no matter which part of visualization they are exploring. Rather, it just visually represents certain information, so that when analysts decide to explore the tagged node or link, they are made aware of the tagged information.

Each tag is defined by a title and a color. Nodes and links with assigned tag are indicated by little dots appearing next to the node or on the link in the tag's color. Since there can be several tags, in the network visualization window, there is an interactive legend presenting all used tags with their title and corresponding color, as illustrated in Figure 4.27.

Currently, the expected number of tags usually used in a case analysis is up to five (based the interviews with analysts).

Textual Notes

The last, but not least option analysts can use to input their thoughts, observations or just working remarks into the visualization is writing textual notes. There are three levels on which a note can be created.

1. A case analysis
2. A visualization document
3. Nodes and links in the network visualization

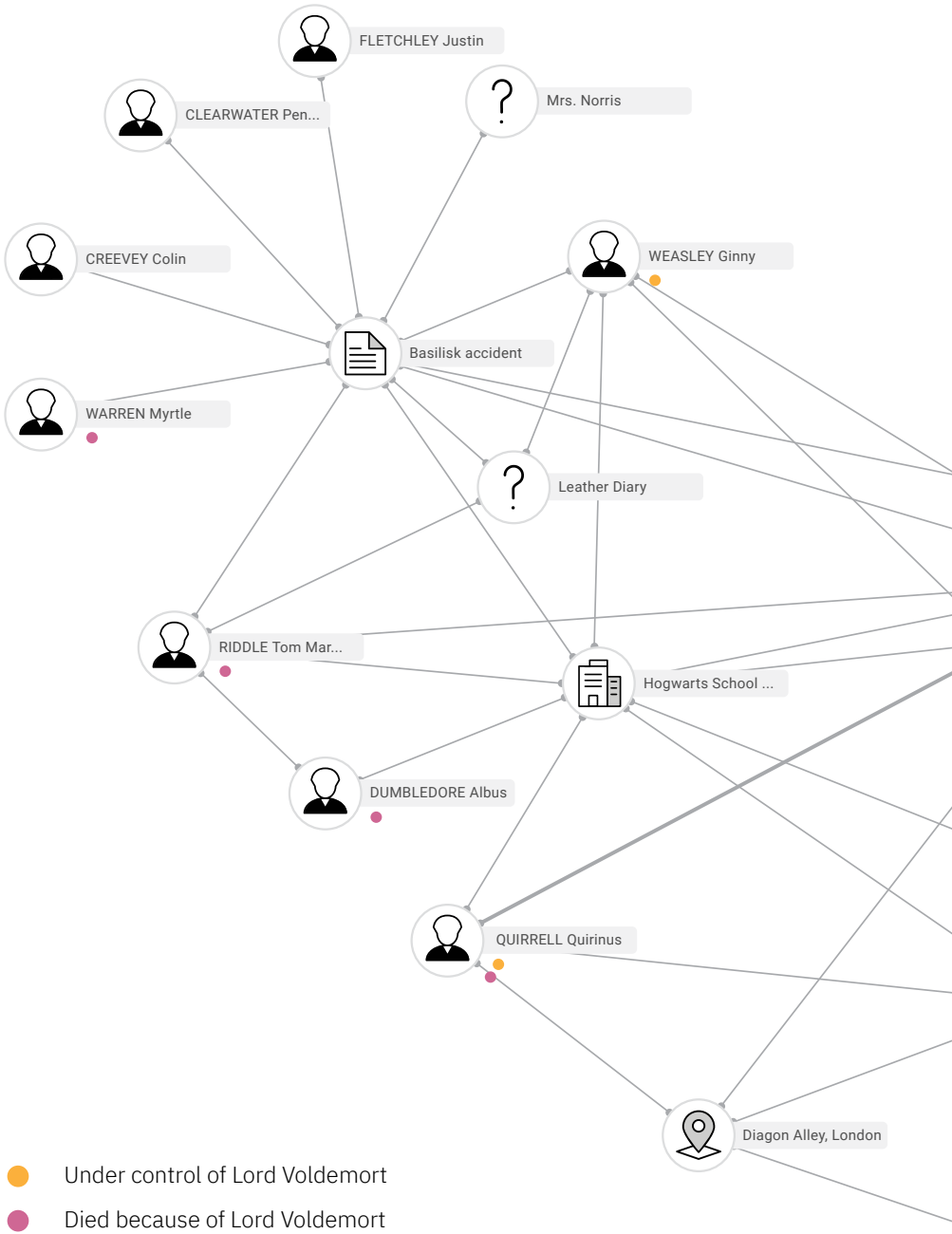
When these notes are created, an identification of the analyst who wrote them and a timestamp of creation are stored. These notes are also accessible to other collaborators.

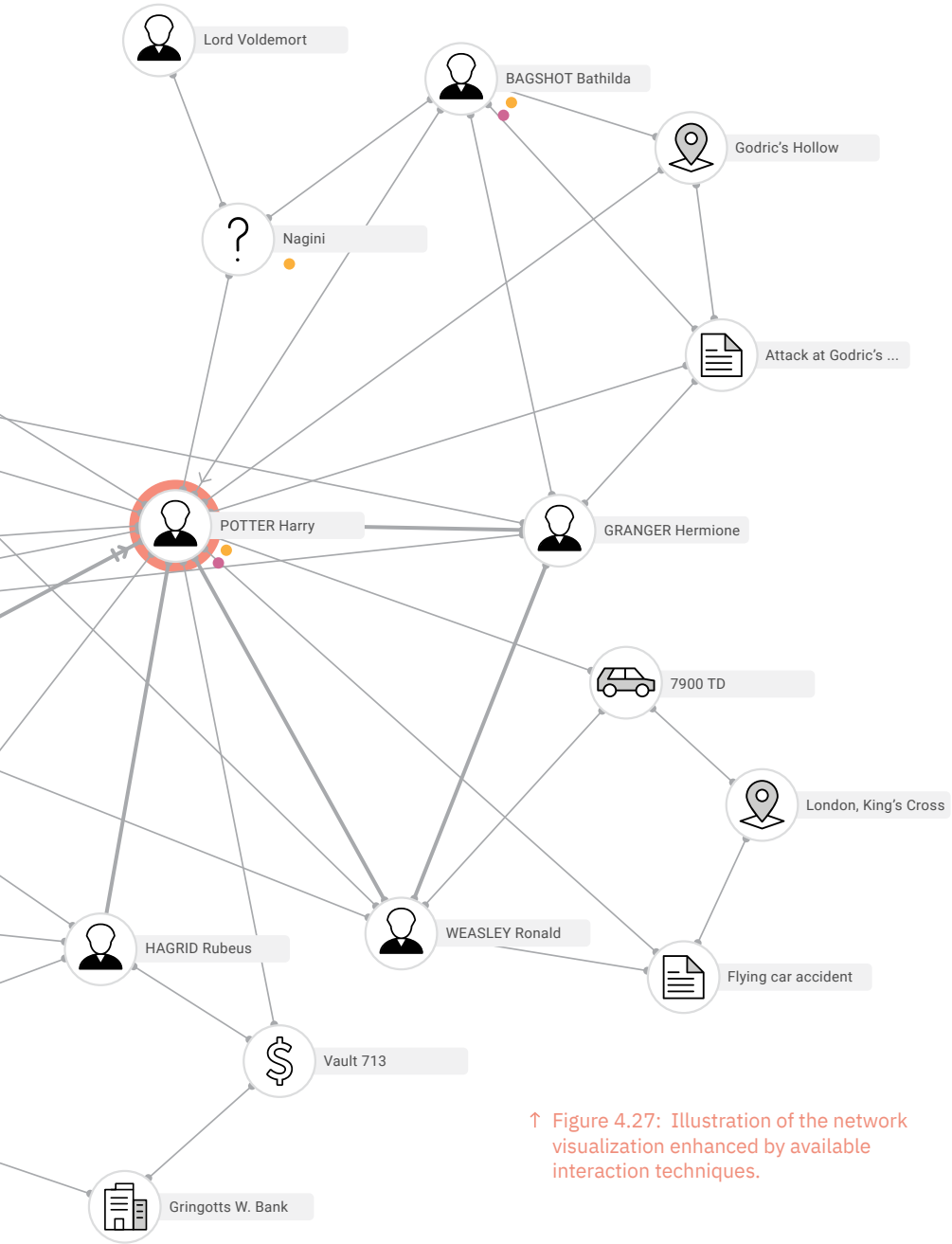
All notes can be accessed through special Notes panel in the visualization component. It contains an interactive list of all notes, divided by their level.

Notes that are connected to nodes or links in the network visualization are also accessible through their corresponding overview card.

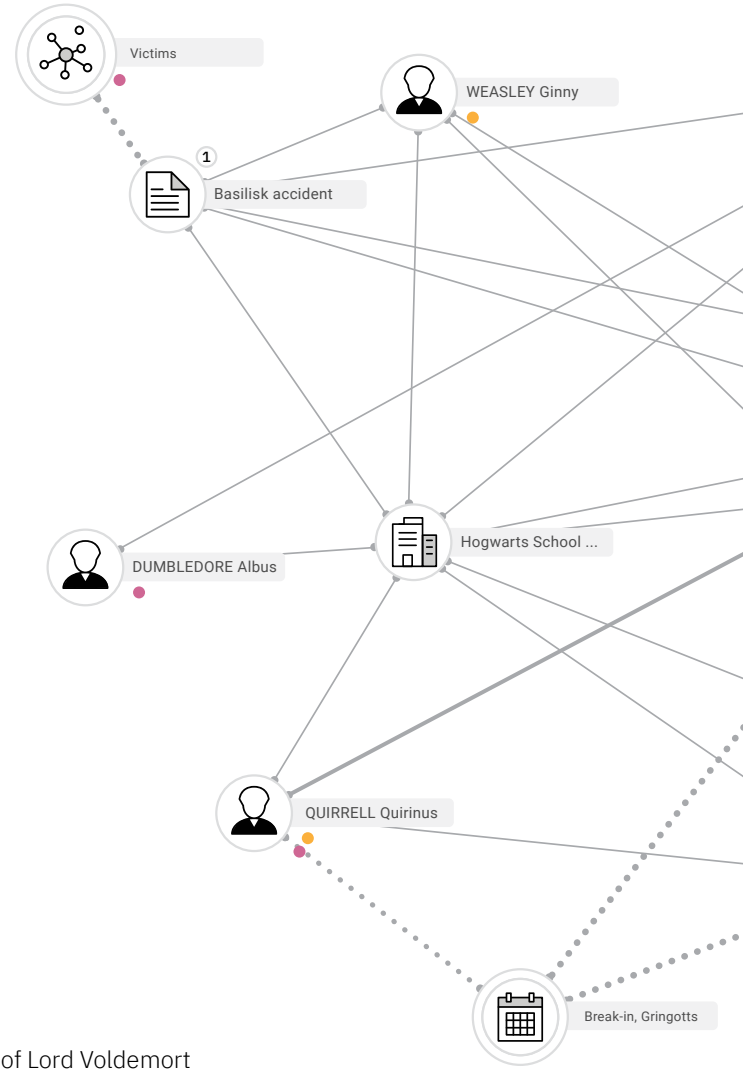
5.2.2.7 Bookmarking

Analyst's interactions and manipulations of data that change the view of visualization or change of general visualization settings, along with saved filters and selections, are always stored in the corresponding visualization document. These changes can be saved and exported as a visualization bookmark at any particular point of time and reloaded later, when necessary.

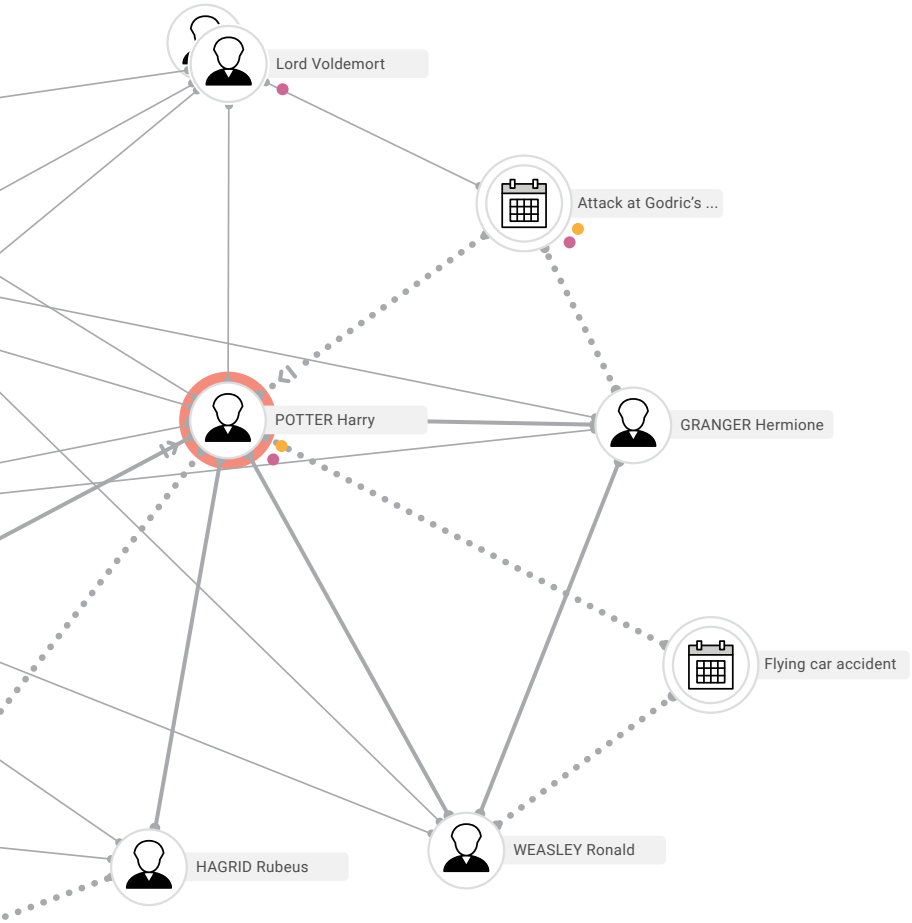

FLETCHLEY Justin
CLEARWATER Pen...
Mrs. Norris
CREEVEY Colin
WEASLEY Ginny
Basilisk accident
WARREN Myrtle
Leather Diary
RIDDLE Tom Mar...
Hogwarts School ...
DUMBLEDORE Albus
QUIRRELL Quirinus
Diagon Alley, London
Under control of Lord Voldemort
Died because of Lord Voldemort



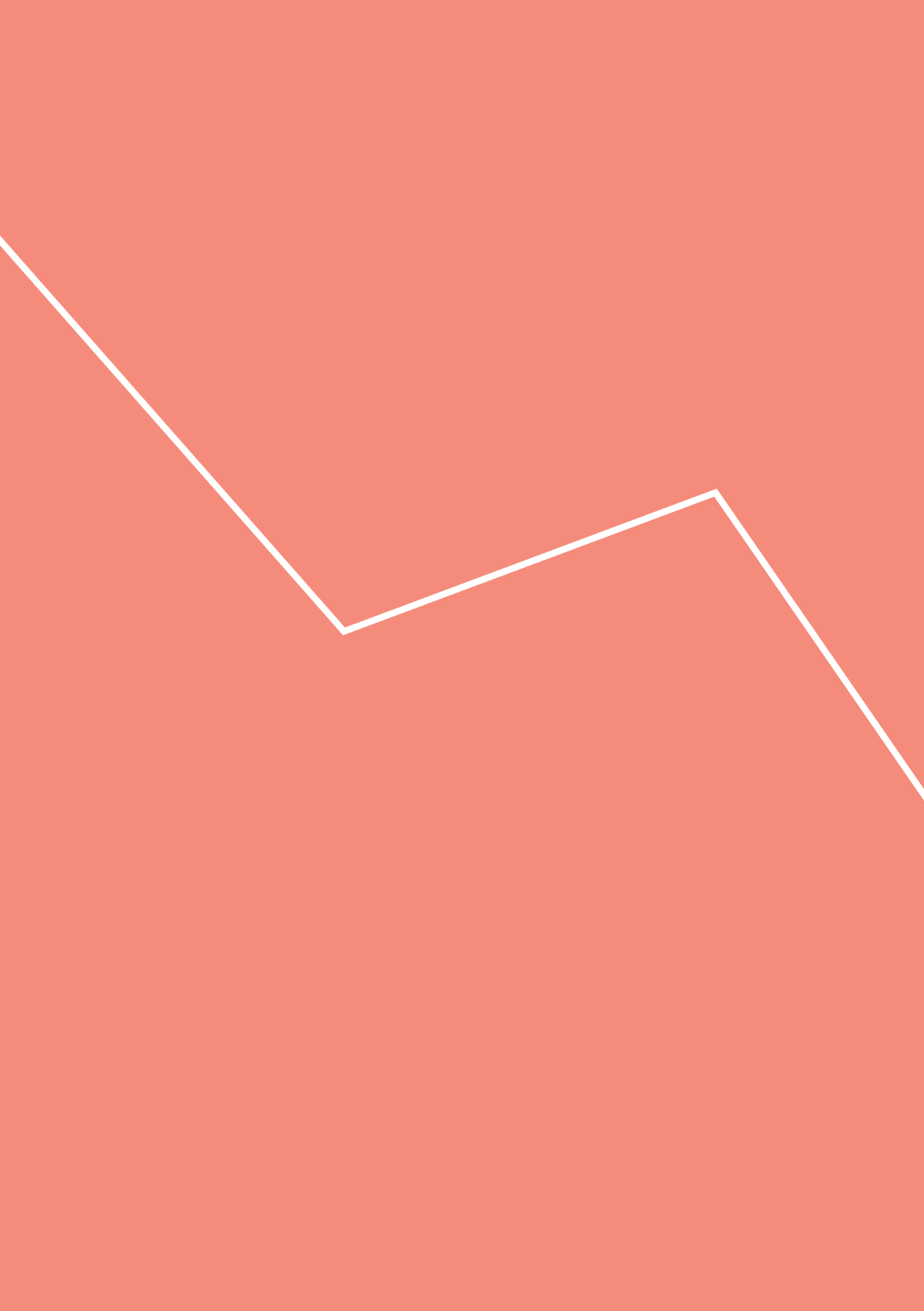
↑ Figure 4.27: Illustration of the network visualization enhanced by available interaction techniques.



- Under control of Lord Voldemort
- Died because of Lord Voldemort



↑ Figure 4.28: Illustration of the network visualization enhanced by available interaction techniques.



5 Implementation

Along with creating a design for the visualization component with detailed design of the network visualization, one of the goals of the thesis is to create a web-based prototype of the network visualization.

In this chapter we first present the technologies we chose for the web application and explain what role they play in the development of the prototype. Then, we provide a description of the web application code to outline what are its main parts and how it works.

5.1 Technologies

Web applications are nowadays the easiest available applications for users, so the focus on development of web-based technologies has been quite significant. Therefore, there is a wide range of options we can choose from in order to build the online application. However, the problem that usually comes with choosing from many is that we need to precisely know what we are looking for and have an overview of what is offered.

5.1.1 Powerful Trio

Independently of the type of web application built, the essence for modern web development is the HTML, CSS and JavaScript, together allowing to specify the content, presentation, and behavior of web pages.

HTML

The basic structure of web page is ensured by HTML (HyperText Markup Language) in an HTML document. HTML specifies which elements, such as titles, paragraphs, or images, should be included in the web page in order to display content to a user. An HTML document is parsed by a web browser into Document Object Model (DOM), hierarchical structure of objects that are displayed on the page and can be further programmatically manipulated.

In our web application, we are using the latest version HTML5 which has brought along new features, such as elements, attributes, event handlers, and APIs, thus facilitating web application development and supporting more sophisticated form handling (Robbins, 2013).

CSS

As we have mentioned, HTML provides a raw structure of contents. However in order to give it a certain, hopefully, nice look, we use CSS (Cascading Style

Sheet) which defines what each element should look like by setting its style, e.g. colors, fonts, layout, etc.

JavaScript

Even though the two technologies, HTML and CSS, are the building blocks of any website, they only allow us to build static content. Therefore, the majority of modern web applications uses also JavaScript to include dynamic content to their webpages. Since all modern web browsers include JavaScript interpreters, they have helped make the JavaScript the most ubiquitous programming language in history (Flanagan 2011).

JavaScript is a high-level, dynamic and untyped interpreted programming language that has been standardized as ECMAScript. In our web application prototype, we are using version ECMAScript 2015 (also known as ES6) which introduced an extensive set of new features for easier web development, such as arrow functions, sets, maps, classes, and much more ("ECMAScript 6: New Features: Overview and Comparison", 2018).

Since JavaScript is so widely used, there is a big range of tools built on top of the core JavaScript language, providing thus a growing amount of functionality and allowing for fast web development. Among these are for example frameworks supporting module-driven websites, creation of visualizations, or offering a multitude of user interface components.

5.1.2 Powerful Frameworks

The development of our network visualization prototype is built primarily on two frameworks, D3.js[1], that allows us to create interactive visualizations built on standard DOM, and Vue.js[2], providing mechanisms for building reusable components with support for reactivity.

1 https://d3js.org/
2 https://vuejs.org/

Another framework that helped facilitate the development is Element, a library offering UI components designed in a Vue-esque pattern.

5.1.2.1 Data-Driven Documents – D3.js

D3 is a JavaScript library created by Mike Bostock and released under a BSD license, which aims at providing a tool which facilitates creation of interactive and animated data-driven visualizations with focus on compatibility, debugging, and performance. D3's force is that instead of creating proprietary representation, "D3 enables direct inspection and manipulation of a native representation: the standard DOM" (Bostock et al., 2011). It selectively binds data to DOM elements and allows for dynamically created and updated web content.

Moreover, D3 enables the user to display large datasets and their dynamic behavior in an interactive manner. (Bostock, 2018)

In the prototype development we worked with the latest release, D3 v4 (version 4). Its biggest change compared to its previous versions is that the original D3 single library has been broken up into many smaller libraries, that are called micro-libraries and are designed to work together (Rininsland and Teller, 2017). The modularization approach also makes it easier to understand, develop, and test the application ("d3/d3", 2018).

The development of the network visualization is based mainly on micro-libraries d3-force[3] used for creation and maintenance of a node-link diagram and d3-selections[4] enabling an effective way for creating element selections and subsequent data-driven transformations of the DOM, such as setting attributes, styles, properties, or content.

D3-force

D3-force micro-library provides a special type of D3's layouts, which are generally used to determine positions of elements in a visualization or graph.

3 https://github.com/d3/d3-force
4 https://github.com/d3/d3-selection

A force layout does so by using a physics-based model, where positions of nodes are computed, based on a combination of attractive and repulsive forces (Foxall, 2018).

Therefore, in order to create our network visualization in D3, all we need is:

- a list of nodes,
- a list of links,
- a simulation,
- and forces that act on the simulation.

Lists of nodes and links our supplied by our data storage, a simulation by D3-force library, the forces are also predefined by the library and we specify their values and weights.

The simulation is the engine of the network visualization which controls the movement of nodes, their position and velocity, by monitoring the forces that are applied to individual nodes. One of variables the simulation also keeps track of is “alpha” referring to the current “energy of the graph”. At the beginning, the alpha is set high and then it decreases with the run of simulation until it reaches certain value and stops. When the simulation stops, the graph stops moving as well. When the simulation runs, the graph can move as well, depending whether the nodes have found their ideal position with respect to the forces. The simulation runs in a sequence of discrete steps that are called ticks. During each tick, simulation’s variables are updated. (Puzzlr, 2018)

As we have already mentioned, forces determine the evolution of the graph. They are applied to simulation which then instructs the graph elements how they should move. Some examples of the forces applicable to our network visualization are:

- A centering force which pushes all nodes to a specified position, thus making it a center of the visualization.
- A collision force making sure nodes do not collide.
- A link force allowing setting approximate distance of connected nodes.

Usually each force has an adjustable strength which determines what are the weights of respective forces in the overall model (Puzzlr, 2018).

5.1.2.2 Vue.js

The second of the two main frameworks used in the development of the network visualization is Vue.js. It is a performant progressive JavaScript framework for building user interfaces. It is designed to be incrementally adoptable, and thus easy to integrate into web applications. It helps to build modular application, utilizes a virtual DOM, provides reactive and composable view components and maintains focus on the core library, while concerns such as routing and global state management are handled by companion libraries. ("Introduction — Vue.js", 2018)

Vue's reactivity is one of its strongest points. Once a Vue instance is created, it stores all properties found in its data object to Vue's reactivity system, and when the data changes, it takes care of all the places which use the data and re-renders the corresponding views to match the updated values ("Introduction — Vue.js", 2018). Each component includes its data object in which all the data properties that has been initialized upfront allow then for reactive changes in the view.

Vue allows creating single file components, which means that in one file we can define component's content (HTML), behavior (JavaScript) and style (CSS).

5.1.3 Make It Work Together

Yarn, JS package manager

Modern web development uses a wide range of frameworks and libraries to make the development more efficient. However, the more of them we use, the more difficult it is to keep the project organized and its management becomes very quickly tedious and susceptible to mistakes. Fortunately, there are tools that help automate the process of downloading and upgrading libraries and frameworks, and keep them all organized. Such tools are called JavaScript package managers. For our web application, we chose Yarn[5] package manager which states to offer fast, reliable and secure dependency management.

Babel, JS transpiler

Another very powerful tool used in our project development is a JavaScript transpiler Babel[6]. As we have mentioned earlier, all modern web browsers include JavaScript interpreters. However, they do not often respond that fast to newer versions of JavaScript standard and, therefore, are not able to interpret it. In real life, that would mean that unless all web browsers support the newer version, developers could not use the new standard, what would be really limiting. Therefore, transpilers, such as Babel, are able to convert edge JavaScript into the supported version of standard that can be run in any browser, and which is currently the ES5 (Johnson, 2018). Therefore, using transpilers makes both sides happy, developers can use the latest standard with all the syntactical sugar and web browsers are still able to interpret the code converted by transpilers.

Webpack

Another build tool helping prepare our web application is a module bundler Webpack[7]. It puts all project's assets, such as JavaScript files, images, fonts, or CSS, into a dependency graph and creates a single file (or a group of files). That means that a developer no longer needs to manually manage JavaScript

5 https://yarnpkg.com/en/
6 https://babeljs.io/
7 https://webpack.js.org/

dependencies by including external files in another file where they are used and make sure they included them in the right order.

5.2 Implementation Details

In this section, we discuss the implementation details and workflow of the prototype network visualization. Thanks to the Vue framework, the implementation of the prototype is divided into components, which create clear structure of the code. However, before we look at individual components in detail, we first present the data format for the network visualization.

5.2.1 Data Format

In order to create the network visualization using D3-force micro library, we have two options to provide it the data, either in the tree-like format of nodes hierarchy, or as two lists, one for the nodes and one for links.

As in criminal investigation it is often not possible to clearly define the hierarchy relationship between two data entities, in other words, to define which one is the parent and which is the child in their relationship, we provide the data using the second type of organization. Therefore, our data is divided into two lists, `nodes` and `links`.

Both nodes and links include many attributes, some of which have information value for the case analysis, and other present only visualization-specific data.

In the Section 3.1.1, we already described the input data that are analyzed using the visualization component. The possible attributes with the information value for analysts are generally derived from the data entry type given by the `type` attribute, however none of these is obligatory. Therefore, it is impossible to present a unified set of present attributes. The only exception is the link's

attributes **source** and **target**, defining the nodes (by their id) which are connected by the link, and therefore are indispensable for the link definition. Examples of node and link data entries are depicted in Figures 5.1 and 5.2.

However, we will describe the remaining visualization-specific data attributes which are crucial for the implementation of the network visualization. Nodes contain the following visualization-specific attributes:

- **id:** numerical identification of nodes in the network visualization
- **node_type:** indicates whether a node is a regular node (**'regular'**) or a group node (**'group'**), which are described in Section 4.2.2.5.
- **user_defined:** boolean variable stating whether the data entry was loaded from the central data storage (**false**), or whether it was created by the analyst in the case analysis (**true**)

```
{
   id:0,
   name: 'Harry',
   surname: 'Potter',
   type: 'person',
   node_type: 'regular',
   label: 'POTTER Harry',
   props: {
      user_defined: false
   }
}
```

↑ Figure 5.1: Example of node data entry.

While links also contain the **id** and **user_defined** attributes with the same meaning as in node's case, they also contain following visualization-specific attributes:

- **direction:** boolean attribute indicating if the link is directional (**true**) or non-directional (**false**). In case it is defined as non-directional, the attributes **source** and **target** are interchangeable

- `transferred:` boolean attribute presenting whether the link is transferred (`true`), that means it is linked to at least one of aggregated nodes, or if it is an original link from the data storage (`false`)
- `multilink:` is set to `true`, if the link represents an aggregation of links between the given two nodes, otherwise to `false`
- `links`: attribute which is present only in case the link is a multilink and stores a list of links that are aggregated in the current multilink.

```
{
   id:11,
   source: 4,
   target: 2,
   direction: false,
   description: 'friends',
   props: {
      user_defined: false,
      multilink: false,
      transferred: false
   }
}
```

↑ Figure 5.2: Example of link data entry.

For the purposes of this prototype, the data is stored in the JSON format in `data.js` file included in the project.

5.2.2 Components

The list of the Vue components used in the implementation of the prototype is depicted in Figure 5.3 which presents the components in the file system structure of the project, and also in Figure 5.4 which depicts the hierarchy of components applied in the implementation.

Even though we created the names for the components so that they would describe their functionality as clearly as possible, we provide their brief description:

App: The main component of the whole prototype.

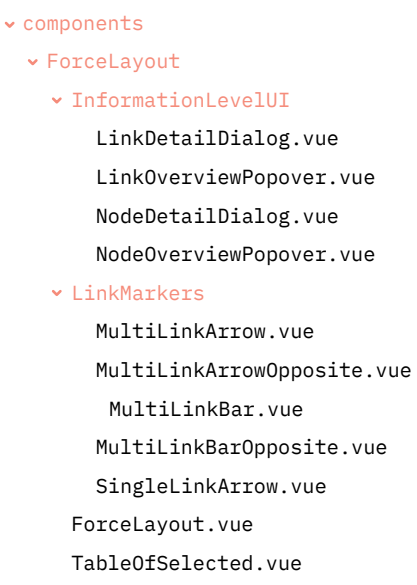
Header: Visual header of the page.

Force Layout: Network visualization with its control panel.

Table of Selected Nodes: Table displaying the list of currently selected nodes along with their basic information.

Node/Link Overview Card: A popover containing the definition for presentation of node's or link's overview information, as described in Section 4.2.2.1.

Node/Link Detail Card: A component containing dialog window presenting the node's or link's detailed information.

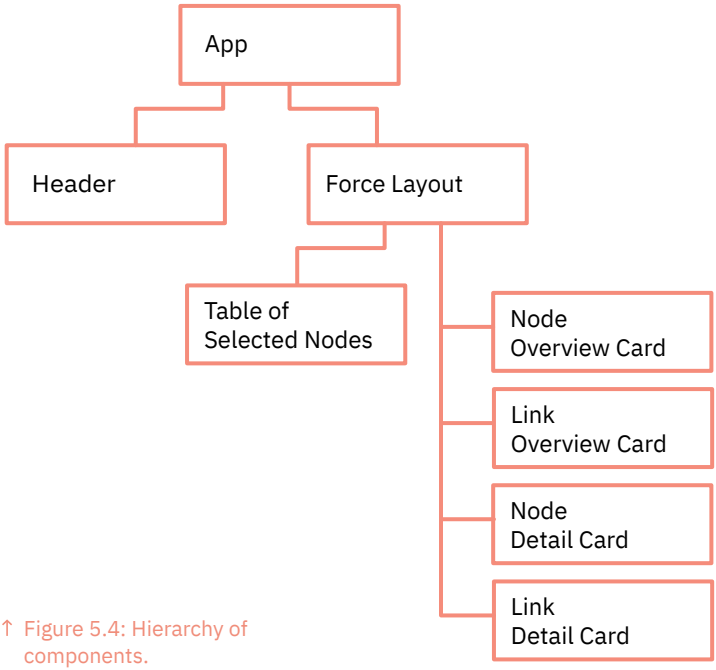


↑ Figure 5.3: Project structure showing individual components.

Single-Link Arrow: Marker definition of arrow which is displayed with a single directional link.

Multi-Link Arrow + Multi-Link Arrow Opposite: Marker definition of arrows for directional multilinks. Each component contains the definition of arrows in one direction, source to target, or target to source.

Multi-Link Arrow + Multi-Link Arrow Opposite: Marker definition for multilinks containing both directional and non-directional links. Each component contains a definition of direction markers for one direction.



↑ Figure 5.4: Hierarchy of components.

5.2.3 The Network Visualization Generation

In this section, we describe the process of generation of the network visualization which takes place in the Force Layout component.

Generation of the network visualization takes place in the Force Layout component. It creates the visualization from the original input data, sets the special data properties for the visualization purposes and ensures the presentation of the visualization to users using DOM objects. The generation process can be divided into the following four steps:

1. **Loading data:** As the first step, before creation of the actual visualization, it is necessary to load data that will be displayed in the visualization. The data is stored as a part of the project in JSON format, and it is loaded into the Force Layout component's `data`'s `graph` property. The data contains a list of nodes and a list of links which are from now on accessible through `this.graph.nodes` and `this.graph.links`.

2. **Initializing force layout:** In order to create the force layout, we first need to create the D3's simulation and assign it nodes, links and forces, as shown in Figure 55. The simulation is initialized by calling the `d3.forceSimulation()` method and is stored in `this.simulation` data property of the component. In the next step, we link data to the simulation. The layout's nodes are specified by calling `nodes(this.graph.nodes)` function on the simulation, while links are assigned to the simulation along with a definition of a link force that takes care of the distance between the connected nodes.


```
this.simulation = d3.forceSimulation()
  .nodes(this.graph.nodes)
  .force("link",
    d3.forceLink(this.graph.links).distance(150).strength(0.05))
  .force("charge",
    d3.forceManyBody().strength(-300).distanceMax(300))
  .force("center",
    d3.forceCenter(
      this.settings.svgWidth / 2,
      this.settings.svgHeight / 2
    ))
  .force("colide",
    d3.forceCollide().radius(50).strength(1).iterations(1))
```

↑ Figure 5.5: Initialization of force layout.

The **link force** is the first force defined for the network visualization. It determines the default length of links, in our case to 150px, along with its strength which specifies how strongly the default length needs to be hold in the force layout. In our case, as we do not require of the links to have any exact or same length, we set its strength to a low value as we set this force's distance attribute only in order to provide an approximation of the default length.

We apply the **many-body force**, which simulates repulsion between all nodes. We also specify the maximum distance property which states how far nodes can be placed from each other. This property needs to be included, so that the nodes do not depart too far, or move away from the visualization window.

The **centering force** allows the nodes to be positioned around selected point in space, in our case, the center of the SVG element containing the force layout.

The last, but not least, force is the **collision force** which prevents nodes from overlapping by specifying node's radius in pixels where no other nodes may appear. The strength set for this force takes the maximum value, indicating that the constraints of this force on the force layout need to be always respected.

Figure 5.5 also presents exact values of individual properties that we assigned to layout forces.

3. **Setting computed properties:** Anytime there is a change of data that is displayed in the network visualization, for example when some nodes are grouped, or a new link is created, we need to update the visualization view. For this purpose, we use the computed properties for the definition of link, node, and sticky paws visual representation.

Computed property is a Vue's feature, which acts as a getter function for the specified property, and it provides a way to define more complex logic for the property definition. Moreover, Vue is aware of all dependencies of the computed property on other data, and anytime this is changed, it updates the computed property and any bindings that depend on the computed property, which is in our case the force layout. ("Introduction — Vue.js", 2018)

Therefore, computed properties for nodes, links, and sticky paws allow for a reactive display of the network visualization anytime the visualization data change.

4. **Setting updated function:** The last step of the network visualization generation is creating a function that updates position of node, link and sticky paw SVG elements based on computations of d3's simulation. This function is called as a part of updated function that is Vue's lifecycle hook called after a data change.

5.2.4 List of Features Presented in the Prototype

The list of following features is included as a part of the network visualization prototype:

- Basic visual representation of nodes
- Basic visual representation of links and multilinks

Data representation

- Local exploration of nodes, including instant, overview and detailed information
- Local exploration of links, including overview and detailed information
- Focus and context techniques for nodes and links
- Manual selection of individual nodes and links
- Creation of new nodes and links
- Basic variant of alias aggregation, bundling and grouping
- Setting focus to selected nodes

Interaction techniques

Conclusion

The aim of this thesis was to create the design of a system for visual analysis of multidimensional data analyzed in criminal investigation, with focus on the design of the network visualization whose basic web-based prototype forms one of the outputs of the thesis.

First, we conducted numerous interviews with the analysts from the criminal investigation domain and performed a thorough study of the existing solutions. Then we formulated the list of requirements on both the whole system for visual analysis, as well as on the network visualization.

Based on the list of requirements, we created the design of the visualization component which allows the analysts to explore the data at three basic levels of detail, one of which is ensured by the network visualization. The network visualization design includes also a detailed description of the data representation and interaction techniques allowing for its visual analysis. Then we created a prototype of the network visualization which presents the basic functionality of the network visualization.

This thesis presents the first, and probably the most important step of the visualization part of the Analýza project. It helped to understand the criminal investigation domain, the typical workflow of the analysts, and gather their requirements on the visual analysis tool. Creating the design of this tool and one part of it, the network visualization, form only the beginning of a long path towards the tool with all desired functions which will aid the analysts in their daily investigation tasks.

Our next steps on early future will focus on the design of a set of specialized visualizations, such as temporal, geospatial, or financial ones, which will help for more complex visual analysis of the criminal investigation data and which will be interconnected with the network visualization.

Bibliography

Angelini, M., Prigent, N., & Santucci, G. (2015). PERCIVAL: proactive and reactive attack and response assessment for cyber incidents using visual analytics. *2015 IEEE Symposium On Visualization For Cyber Security (Vizsec)*. doi: 10.1109/vizsec.2015.7312764

Ball, R., Fink, G., & North, C. (2004). Home-centric visualization of network traffic for security administration. *Proceedings Of The 2004 ACM Workshop On Visualization And Data Mining For Computer Security - Vizsec/DMSEC '04*. doi: 10.1145/1029208.1029217

Bernhard, M. (2018). [Image]. Retrieved May 7, 2018, from http://www.mathiasbernhard.ch/diary/wp-content/uploads/2015/02/wiki_full.png

Bertin, J., & Berg, W. (2011). *Semiology of graphics*. Redlands, Calif.: ESRI Press.

Bostock, M. (2018). D3.js - Data-Driven Documents. Retrieved March 24, 2018, from https://d3js.org/

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 Data-Driven Documents. *IEEE Transactions On Visualization And Computer Graphics, 17*(12), 2301-2309. doi: 10.1109/tvcg.2011.185

Card, S., Mackinlay, J., & Shneiderman, B. (1999). *Readings in information visualization*. San Francisco, Calif.: Morgan Kaufmann.

d3/d3. (2018). Retrieved March 24, 2018, from https://github.com/d3/d3/blob/master/CHANGES.md#changes-in-d3-40

Duncan, J., & Humphreys, G. (1989). Visual search and stimulus similarity. *Psychological Review, 96*(3), 433-458. doi: 10.1037/0033-295x.96.3.433

ECMAScript 6: New Features: Overview and Comparison. (2018). Retrieved March 24, 2018, from http://es6-features.org

Few, S. (2009). *Now you see it*. Oakland, Calif.: Analytics Press.

Few, S. (2013). Data Visualization for Human Perception. In M. Soegaard & R. Dam, *The Encyclopedia of Human-Computer Interaction* (2nd ed.). The Interaction Design Foundation. Retrieved from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed

Flanagan, D. (2011). *JavaScript: The Definitive Guide*. O'Reilly Media.

Foxall, B. (2018). What's new in D3 v4?. Retrieved April 21, 2018, from https://blog.pusher.com/whats-new-in-d3-v4/

Gutfraind, A., & Genkin, M. (2017). A graph database framework for covert network analysis: An application to the Islamic State network in Europe. *Social Networks*, *51*, 178-188. doi: 10.1016/j.socnet.2016.10.004

Harrison, L., Hu, X., Ying, X., Lu, A., Wang, W., & Wu, X. (2010). Interactive detection of network anomalies via coordinated multiple views. *Proceedings Of The Seventh International Symposium On Visualization For Cyber Security - Vizsec '10*. doi: 10.1145/1850795.1850806

Healey, C. (2018). Perception in Visualization. Retrieved April 22, 2018, from https://www.csc2.ncsu.edu/faculty/healey/PP/index.html

Hughes, C., Bright, D., & Chalmers, J. (2017). Social network analysis of Australian poly-drug trafficking networks: How do drug traffickers manage multiple illicit drugs?. *Social Networks*, *51*, 135-147. doi: 10.1016/j.socnet.2016.11.004

Introduction — Vue.js. (2018). Retrieved March 24, 2018, from https://vuejs.org/v2/guide/

Johnson, N. (2018). What is Babel, and how will it help you write JavaScript?. Retrieved April 21, 2018, from http://nicholasjohnson.com/blog/what-is-babel/

Kosara, R. (2018). Encoding vs. Decoding. Retrieved March 10, 2018, from https://eagereyes.org/basics/encoding-vs-decoding

Kosara, R., Miksch, S., & Hauser, H. (2002). Focus+context taken literally. *IEEE Computer Graphics And Applications*, *22*(1), 22-29. doi: 10.1109/38.974515

Liao, Q., Striegel, A., & Chawla, N. (2010). Visualizing graph dynamics and similarity for enterprise network security and management. *Proceedings Of The Seventh International Symposium On Visualization For Cyber Security - Vizsec '10*. doi: 10.1145/1850795.1850799

Masys, A. (2014). *Networks and Network Analysis for Defence and Security*. Cham: Springer International Publishing.

Ozgur, S. (2018). Behance. Retrieved November 11, 2017, from https://www.behance.net/gallery/30156317/100-Free-Web-and-App-UI-icons

Preattentive processing. (2018). Retrieved April 22, 2018, from http://www.infovis-wiki.net/index.php?title=Preattentive_processing

Puzzlr. (2018). Basics of d3 force directed graphs. Retrieved April 4, 2018, from http://www.puzzlr.org/basics-of-d3-force-directed-graphs/

Quinlan, P., & Humphreys, G. (1987). Visual search for targets defined by combinations of color, shape, and size: An examination of the task constraints on feature and conjunction searches. *Perception & Psychophysics*, *41*(5), 455-472. doi: 10.3758/bf03203039

Rininsland, Æ., & Teller, S. (2017). *D3.js 4.x Data Visualization - Third Edition*. Birmingham: Packt Publishing.

Robbins, J. (2013). *HTML5 Pocket Reference, 5th Edition*. O'Reilly Media, Inc.

Strang, S. (2014). Network Analysis in Criminal Intelligence. In A. Anthony J. Masys, *Networks and Network Analysis for Defence and Security* (pp. 1-26). Cham: Springer International Publishing.

Thomas, J., & Cook, K. (2005). *Illuminating the path*. Los Alamitos, Calif.: IEEE Computer Society.

Thomas, J., & Cook, K. (2006). A visual analytics agenda. *IEEE Computer Graphics And Applications*, *26*(1), 10-13. doi: 10.1109/mcg.2006.5

Treisman, A. (1986). Preattentive Processing in Vision. *Human And Machine Vision II*, 313-334. doi: 10.1016/b978-0-12-597345-8.50017-0

Tsigkas, O., Thonnard, O., & Tzovaras, D. (2012). Visual spam campaigns analysis using abstract graphs representation. *Proceedings Of The Ninth International Symposium On Visualization For Cyber Security - Vizsec '12*. doi: 10.1145/2379690.2379699

Tukey, J., & Wilk, M. (1966). Data analysis and statistics. *Proceedings Of The November 7-10, 1966, Fall Joint Computer Conference On XX - AFIPS '66 (Fall)*, 695-709. doi: 10.1145/1464291.1464366

van der Hulst, R. (2008). Introduction to Social Network Analysis (SNA) as an investigative tool. T*rends In Organized Crime*, 12(2), 101-121. doi: 10.1007/s12117-008-9057-6

Ware, C. (2013). *Information visualization*. Amsterdam: Elsevier/MK.


Conclusion
quite a lot
too ashamed to admit
5
4
3
2
1
Chapter
Expected time
Actual time
Introduction
a little
quite some


15
and much more
142
People consulted
Coffee drank
Last chapter page
0
a lot
0