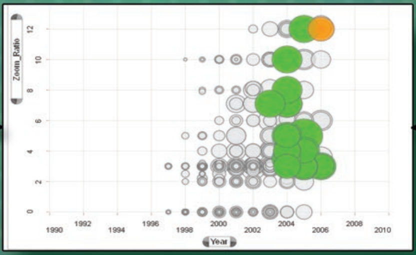
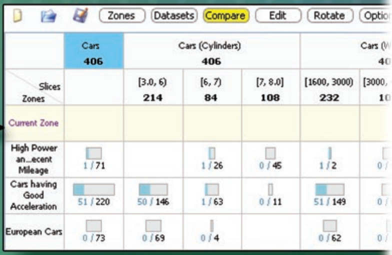
Supporting the sensemaking process in visual analytics

DOI:
10.6100/IR673142

Supporting the Sensemaking Process in Visual Analytics





	Cars 406	Cars (Cylinders) 406			Cars (W 40	
Slices Zones		[3.0, 6) 214	[6, 7) 84	[7, 8.0] 108	[1600, 3000) 232	[3000, 10
Current Zone						
High Power an...ecent Mileage	1 / 71		1 / 26	0 / 45	1 / 2	0 /
Cars having Good Acceleration	51 / 220	50 / 146	1 / 63	0 / 11	51 / 149	0 /
European Cars	0 / 73	0 / 69	0 / 4		0 / 62	0 /

Yedendra Babu Shrinivasan

Supporting the Sensemaking Process in Visual Analytics

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
maandag 21 juni 2010 om 16.00 uur

door

Yedendra Babu Shrinivasan

geboren te Tiruvallur, India

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. J.J. van Wijk

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-2229-3

To my Mother

Promotor:
prof.dr.ir. J.J. van Wijk (Technische Universiteit Eindhoven)

Kerncommissie:
prof.dr. Helwig Hauser (University of Bergen, Norway)
prof.dr. Menno-Jan Kraak (University of Twente, The Netherlands)
prof.dr.ir. J.B.O.S. (Jean-Bernard) Martens (Technische Univeriteit Eindhoven)
prof.dr.ir. Robert van Liere (Centrum voor Wiskunde en Informatica, The Netherlands)

NWO
Netherlands Organisation for Scientific Research

Advanced School for Computing and Imaging

This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 643.100.502.

The work in this thesis has been carried out under the auspices of the research school ASCI (Advanced School for Computing and Imaging). ASCI dissertation series number: 203.

Published by Technische Univeriteit Eindhoven

Typeset in LaTeX.
Printed by Gildeprint Drukkerijen, Enschede

Contents

Preface

In 2005, I worked with analysts at Disaster Management Center, National Remote Sensing Center (NRSC) in Hyderabad. This was an exciting opportunity and a big motivation to pursue this research work. Analysts had to handle large volumes of remote sensing and attribute data for assessing and managing disasters such as floods, draught, cyclones and earth quakes. One of the major problems they faced is the management of the analyses results and provenance. To achieve a solution for the above mentioned, I wanted to bank on Geo-informatics to design and develop few geovisualization tools to support their analysis. During this collaboration, analysts expressed interest to capture visualization views along with notes that can ease their report writing process. I developed a report organization tool called 'Vritrahan' to support this reporting process. This tool, however, captured only screenshots of the visualization views (similar to Microsoft OneNote) and did not capture the provenance information. During this collaboration, it occurred to me how most of the analysis tools only support the process of converting data to useful information, and stop right after there. Analysts faced a hectic task of managing the inputs and results of different iterations of an analysis.

In 2006, I came across the NWO 'Expression of Interest' project proposal through the Academic Transfer website. The proposal had a section on supporting user navigation in interactive visualizations. It aimed at managing user interest on data items during an exploration process by intuitively capturing and presenting user interest, on data items. I felt that there was a match between the problem recorded earlier while interacting with analysts and the problem described in the proposal. So, I was stimulated to apply for this PhD position.

After a the telephonic interview, in a few days Prof. Jarke van Wijk invited me for a personal interview. Due to my job commitments I was unable to travel abroad. Alternatively, he spoke to my masters supervisor Prof. Menno-Jan Kraak, and decided to provide me with the fortunate opportunity to further explore my potentials under his guidance. I am very thankful to Prof. Jarke van Wijk for being flexible with me in this regard and taking the risk of hiring me without an initial meeting also; and Prof. Menno-Jan Kraak for recommending me for this position, even when he was also arranging a PhD position for me in the meantime. I also thank the Netherlands Organisation for Scientific Research (NWO) for funding my PhD Project (Project no. 643.100.502).

When I started to work at TU/e in May 2006, I knew little about Prof. Jarke van Wijk, who was known as Jack in the visualization group. We met weekly and discussed about my work. Soon, I learnt he is an easy to approach, extremely bright and smart person.

I had a relatively simple idea to solve the above problem that only looked new due to the combination of existing techniques. Also, the complete implementation of the idea took over a year. Aruvi was my first C++ GUI program as well as the largest application I developed. Because of Jack's sharp guidance and patience with me, the idea saw the light, and was well-received in the visual analytics community. In this process, he taught me how to pursue research, and also, he identified one of my strengths — networking which were never realized until then. One afternoon, when we had a walking meeting, I asked him a question, "what is the purpose of doing a PhD?" expecting from him an answer that gives some career guidance. But he gave an enlightening reply: "for me, PhD is the process of making of a person. You test your strengths, identify your weaknesses and learn how to handle them." This reply has a great impact on the personal account and also helped me to remain positive during the undulating course of the PhD research. The way he pursued his hobby project was really amazing and inspiring. His PhD students never realized that he was on sabbatical to work on his hobby project, because he was always available for discussions during this period. Jack, you have led us by example. I have quite a number of situations that are retained in my memory and will keep me motivated. Thank you so much for being such a great advisor.

I thank Prof.dr. Helwig Hauser (University of Bergen, Norway), Prof.dr. Menno-Jan Kraak (University of Twente, The Netherlands), Prof.dr.ir. J.B.O.S. (Jean-Bernard) Martens (Technische Univeriteit Eindhoven) and Prof.dr.ir. Robert van Liere (Centrum voor Wiskunde en Informatica, The Netherlands) for taking part in the core doctoral committee. Your comments were useful in strengthening this dissertation. I also thank David Gotz (IBM Research, NY, USA) and Prof. dr. M.G.J. (Mark) van den Brand (Technische Univeriteit Eindhoven) for participating in the extended committee. I am thankful to Dr. Tamara Munzner (University of British Columbia, Canada) and Prof.dr. John T. Stasko (Georgia Institute of Technology, USA) for productive discussions during their visit to Eindhoven.

Throughout the four years of my PhD I have enjoyed the company of my colleagues at the visualization group, with whom I had fruitful and sometimes fun filled discussions. I thank my fellow doctoral students and post-docs Hannes Pretorius, Lucian Voinea, Jing Li, Dennie Reniers, Danny Holten, Koray Duhbaci, Romain Bourqui, Mickeal Verschoor and Niels Willems. I also thank senior researchers at our group, Huub van de Wetering, Alex Telea, Kees Huizing, Michel Westenberg and Andrei Jalba. I also thank Frank van Ham (IBM/ILOG, France) for his motivation and guidance. I also thank Ajay, Christian Lange, Serguei Roubtsov, Reinier Post, and Joost Gabriels for their feedback on Aruvi. I am grateful to Tineke van den Bosch, Elisabeth Melby, and the personnel affairs staff members for their support in the complex administrative procedures. I also thank Cicek Guven and Elena for their support while I carried out my tasks as a chairman at the PromoVE board.

I thank David Gotz for providing me an opportunity to do an internship at IBM Research NY, USA. It was a good experience to work in a world class industry lab. The collaboration work was successfully turned into a paper and two IPs. I also thank Jennifer Lai, Jie Lu, Shimei Pan, Zhen Wen, Peter Kissa and Michelle Zhou.

This PhD study would not have been possible without the help of the people who helped me to shape my foundation in both studies and personal development. I am grateful to prof.dr.Menno-Jan Kraak who inspired me to pursue visualization research through his interesting lectures and his guidance during master's thesis. I thank prof.dr. Sanjeevi for motivating me to pursue research, during my undergraduate studies. I also thank my school teachers Natesan (Physics), Viswanathan (Social Science) and Babu (English) who helped me groom my analytical and leadership skills at school and remain as a big inspiration for me till date. I also thank my senior colleagues at NRSC, Dr. Y.V.S Moorthy, G. Srinivasa Rao, Dr. P.S. Roy and Dr. K. Radhakrishnan for their encouragement in pursuing a research career. I also thank my college seniors Ashok Subramanian (Shinota NLP consulting) for exposing me personality management and Narayanan Ramanathan (Satyam Computer Services Ltd) for supporting my NUFFIC application that helped me to visit the Netherlands.

On a personal level, support from my friends was really appreciated. I thank Suhasini Natarajan, Thulasiraman, Sudhira, Ajay, JP, Archana, Kavitha. J, Murali Krishnan, Karade, Bhole, Abhinav, PP, Kaushik, Subbu, Ravi, Sunil, KMJ, and Ujwal. Kuru, you are such a simple and considerate person. I enjoy every second being with you. Thank you so much.

This dissertation is dedicated to my Mother. She has been the source of my inspiration. She always hid all the pains, and made us see only the beautiful side of the life. She is always there to support me. She and my Father (Nina) were supportive and encouraged me to pursue all my dreams. Nina, you have been my strength. I thank my Sister Janaki, my Brother-in-law Sathish kumar, and my nephews Akshay and Aryia for their unlimited support and motivation.

Yedendra B. Shrinivasan
Eindhoven
June 2010

Chapter 1

Introduction

The power of the unaided mind is highly overrated. Without external aids, memory, thought and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities. How have we increased memory, thought and reasoning? By the invention of external aids: It is things that make us smart. — Donald A. Norman, Things That Make Us Smart: Defending Human Attributes In The Age Of The Machine, 1993.

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces [122]. It involves representing information visually and allowing the human to directly interact with it, to gain insight, to draw conclusions, and to ultimately make better decisions [78]. It aims to support the sensemaking process in which information is collected, organized and analyzed to form new knowledge and inform further action [30]. A recent report [122] identifies developing tools and techniques for supporting the sensemaking process as a grand challenge in the visual analytics research agenda.This dissertation focuses on developing external aids to support the sensemaking process in visual analytics during interactive data exploration.

Pirolli and Card [103] identified two major loops in the sensemaking process — the information foraging loop and the sensemaking loop. They also found that analysts opportunistically mix these two loops during that process. During the information foraging loop, analysts transform data into meaningful information and get insight into the problem. In the sensemaking loop, they review and organize insights to build a case and present it to others. Often they tend to refer back to the analysis process and the findings during the sensemaking loop. However, until recently, researchers, designers and developers of analytical systems have given most emphasis on just developing tools and techniques for supporting the information foraging loop.

1.1 Making Sense of Data

Today, data is abundant. We collect data about our daily activities and about objects that we interact with during those activities. We need to make sense of such abundant data for making effective decisions. The management of large and complex data was a challenging task until the development of various database technologies. Now using databases, we can organize large volumes of structured and unstructured data at home, at enterprises and on the Internet. An important aim for collecting and organizing data is to facilitate data analysis for effective decision making. In this context,

> *The major obstacle to solving modern problems isn't the lack of information, solved by acquiring it, but the lack of understanding, solved by analytics.*
>
> - Malcolm Gladwell, journalist and writer, SAS Institutes Innovators Summit, 2009.

During data analysis, analysts engage in confirming or deriving hypotheses by interactively exploring data using various techniques such as information visualization, statistical analysis, spreadsheets, and data mining, to name a few. They often perform analytical activities such as summarizing data, making predictions and identifying trends, patterns and outliers to derive new knowledge [96]. However, deriving new knowledge is not the end of the sensemaking process. The new knowledge creates more questions and hypotheses that require further analysis of the data. Hence, analysis is an iterative process. Each iteration produces new insight which analysts have to manage for effective reasoning during a long exploration process.

Visual analytics has a wide range of application areas including business, biology, health care, engineering, cyber security, public safety and security, governance, environmental protection, and personal information management. Visual analytics research focuses on handling complex and large data. Stock market analysis, portfolio analysis and risks management in the financial business need to handle large amounts of historic and real-time data. Analysts carry out complex analysis processes to make business decisions such as market and customer analysis and business process optimization. Also, in the case of public safety and security, data from heterogeneous sources such as text data from news articles, intelligence report, and blogs; network data from telephone calls and social network have to be integrated and analyzed for making effective security decisions.

On the other hand, Christian Chabot, CEO of Tableau Software, during his keynote speech at VAST 2008, emphasized on a general misconception that 'people adopt visual analytics primarily to help them see and understand only massive and complex data.' Most people handle massive simple data; often stored in Excel spreadsheets and Access databases. Also, he argued that people often don't only look for hidden insights. They use visual analytics tools in more mundane tasks that help them to get out of the way, and think about the data; rather than distracted by the mechanics of using the software. For instance, some data encountered at home such as income expenditure, energy consumption, and health care, though small, can become large as these accumulate over a long time period. Thus, we encounter much data that are either complex or simple, both at work as well as at home; and have to make sense of this. We do not keep track of all the findings and key aspects of those analysis processes; hence, we cannot review or reuse them for making effective decisions in a timely manner.

Often sensemaking of data is a social process [67, 130]. Many analysts collaboratively investigate the data with different analysis goals within an organization. They need to review and share their findings as well as their analysis process. They also have to be aware of their collaborators findings to avoid redundant rediscovery and lose time by inadvertently repeating an analysis process. Thus, an approach to support the sensemaking process in visual analytics should consider both the analysts and their collaboration environment.

1.2 Research Problem and Approach

The central theme of this dissertation is

> *How to support users in their sensemaking process during interactive exploration of data?*

One approach to support the sensemaking process in visual analytics is to enable analysts to capture aspects of interest while interactively exploring data; and to support analytical tasks such as reviewing, reusing and sharing these. The key aspects of interest while interactively exploring the data concern the analysis process and the findings. In addition to developing tools and techniques to interactively explore data and get insight, we argue that for an effective sensemaking process users must be enabled to

- capture the key aspects of interest along with the rationale by which a finding is derived;
- reuse the key aspects of interest during the exploration process to simplify and derive insights in a rapid manner;
- review and share the analysis process and the findings; and
- identify connections between findings.

Our approach is shown in Figure 1.1. When analysts explore the data using interactive visualization, we enable them to capture and archive the key aspects of interest concerning the analysis process and the findings. Later, they can retrieve those key aspects of interest from past analyses to reuse these in the current analysis. They can also organize their findings and engage in discussion by sharing or presenting these to their collaborators. During discussion several questions can be raised or hypotheses can be formed. Next, analysts can retrieve and review their previous analyses or seek out an alternate line of inquiry to verify them. The new findings are again captured. Thus, analysts can revisit, reuse, review and share their analysis process and findings.

Therefore, to support the sensemaking process in visual analytics, we mainly focus on

> How to support users to *capture, reuse, review, share and present* the key aspects of interest concerning *the analysis process and the findings* during interactive exploration of data?

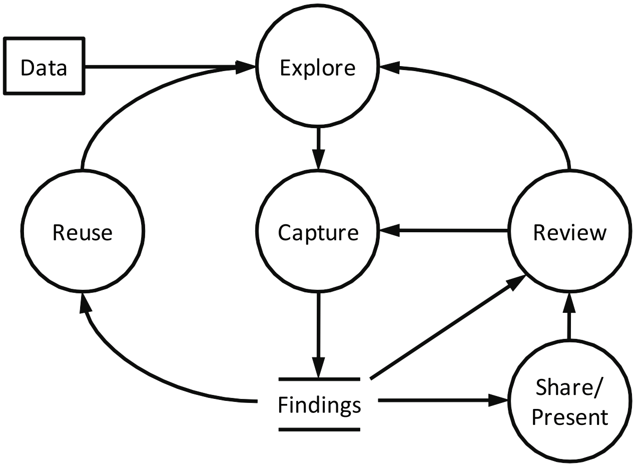


Figure 1.1: An approach to support the sensemaking process in visual analytics.

1.3 Contribution

The key contributions of this dissertation are as follows:

1. A new information visualization framework that contains three linked views: a data view, a navigation view and a knowledge view for supporting the sensemaking process in visual analytics. The data view offers interactive data visualization tools. The navigation view automatically captures the interaction history using a semantically rich action model and provides an overview of the analysis structure. The knowledge view is a basic graphics editor that helps users to record findings with provenance and to organize findings into claims using diagramming techniques. Thus, users can exploit the automatically captured interaction history as well as manually recorded findings to review and revise their visual analysis. Finally, the analysis process can be archived and shared with others for collaborative visual analysis.

2. Semantic Zones: areas in data space with a clear semantic meaning. Users are enabled to define zones using data selection techniques such as dynamic queries and direct manipulation while interactively exploring the data. A *Select & Slice* table is used to project slices of data on different zones. Semantic zones and data slices are arranged along the horizontal and vertical headers of a table, each cell contains a set of items of interest obtained by projecting a semantic zone on a data slice. These sets can be visualized in various ways, ranging from just a count, an aggregation of a measure to a separate visualization, such that the table gives an overview of the relation between zones and slices. Furthermore, users can reuse zones, combine zones, and compare and trace items of interest across different semantic zones and data slices.

3. Support for exploration awareness via an overview of what has been done and found during an analysis process. Users are enabled to develop exploration awareness through a key aspects overview. A users' information interest model is developed to extract key aspects of a visual analysis and an overview of these is presented. The key aspects of the exploration process are the visualization specification, the data specification, viewed objects and selected objects. By interactively exploring the analysis structure and the key aspects overviews, users can identify analysis strategies used in a visual analysis. Such overviews help to review and continue a past visual analysis.

4. Searching techniques to retrieve visualizations and notes from the past analyses for supporting a review process, based on keywords, content similarity and context. Also, related notes and visualizations are recommended to users during a visual analysis using a context based retrieval algorithm. Thus, they can identify connections between findings discovered at various point of time that would normally go unnoticed during a visual analysis.

5. Aruvi is a research prototype developed to study the implications of these models on a user's sensemaking process. Currently, data analysts from different domains such as software quality analysis and urban planning use Aruvi to carry out some of their data analysis tasks. They participated in short-term and long-term case studies conducted to investigate the impact of the Aruvi system on their sensemaking process. The observations of the case studies are used to evaluate the models.

1.4 Outline

The remainder of this dissertation is organized as follows:

Chapter 2 discusses background work related to visual analytics and the sensemaking process.

Chapter 3 introduces an information visualization framework to support the analytical reasoning process. It consists of three views: a data view, a navigation view, and a knowledge view. We present Aruvi, an information visualization prototype that supports the analytical reasoning process in information visualization using the new framework. It helps analysts to capture the analysis process and findings and to link findings to visualization states. We also present a user study that evaluates the support offered by the framework.

Chapter 4 introduces semantic zones and presents techniques to capture them during a visual data analysis. We present a Select & Slice table to project zones on different data slices. Finally, we discuss the implications of the Select & Slice table during the exploration process using case studies.

Chapter 5 introduces the concept of exploration awareness and the user's information interest model. We present our method to provide the analysis structure and the key aspects overview. Next, we describe two search and retrieval mechanisms - keyword based and content similarity based — to retrieve visualizations from past analysis. Finally, we present three case studies to evaluate the support for exploration awareness during the exploration process.

Chapter 6 presents an analysis context based retrieval algorithm that supports connection discovery during exploration process. For a given visualization state, it retrieves related notes and related concepts from past analyses. A recommendation feature is implemented in HARVEST, a web based visual analytics system, based on the context based retrieval algorithm. This work was done by the author during his internship at IBM Hawthorne in 2008.

Chapter 7 presents the lessons learned from analysts using Aruvi.

Chapter 8 concludes this dissertation and presents future work.

Parts of this dissertation have been published before, specifically

- Shrinivasan, Y.B. and Van Wijk, J.J. 2008. Supporting the analytical reasoning process in information visualization. In Proc. ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '08, 1237–1246. (Chapter 3);

- Shrinivasan, Y.B. and Van Wijk, J.J., Supporting exploratory data analysis using the Select & Slice table, Computer Graphics Forum: Eurographics/IEEE Symposium on Visualization (EuroVis '10), To appear, 2010.(Chapter 4)

- Shrinivasan, Y.B. and Van Wijk, J.J. 2009. Supporting exploration awareness in information visualization. IEEE Computer Graphics & Applications. 29, 5 (Sept. 2009), 34–43. (Chapter 5);

- Shrinivasan, Y.B. Gotz, D. and Jie Lu. 2009. Connecting the dots in visual analytics, Proc. of IEEE VAST, 123–130. (Chapter 6);

Chapter 2

Background

> *From the smallest necessity to the highest religious abstraction, from the wheel to the skyscraper, everything we are and everything we have comes from one attribute of man - the function of his reasoning mind.* — Ayn Rand.

In Chapter 1, we discussed our aim to support the sensemaking process in visual analytics during interactive data analysis. In this chapter, we discuss background work related to visual analytics and the analysis process. Visual analytics primarily has evolved out of the field of visualization. First, we discuss visualization, and then introduce visual analytics research and its scope. Next, we review models that support the analysis process in visual analytics. Based on this discussion, we derive requirements for supporting the sensemaking process during visual data analysis. Finally, we present an overview of the state-of-the-art in visual analytics, and position our dissertation in this work.

2.1 Visualization

Visual analytics has evolved out of the fields of information visualization, scientific visualization and geovisualization. The idea behind these visualization fields is to represent data or concepts using graphical representations, and enable users to interactively explore these. These fields engage human visual information processing capabilities to reason about data following the saying *'A picture is worth thousand words.'* These data graphics acts as an external aid to enhance human cognition on data. Card et al. define visualization as *'the use of computer-supported, interactive, visual representations of data to amplify cognition'* [30]. With the advent of computing technology, large datasets can be quickly transformed into meaningful visualizations. Therefore, users can quickly see and explore representations of large data under investigation on a computer screen.

Scientific visualization handles large sets of scientific data to enhance scientists' ability to see phenomena in the data [93]. It concerns interactive investigation of physical data — the human body, the earth, molecules or other [30]. Information visualization handles non-physical information such as finance data, business information, documents and abstract conceptions. This information does not have an obvious spatial mapping. Hence, the fundamental challenge in information visualization is about choosing or developing representations to visualize these abstract data. Geovisualization handles geographic data and helps to gain insight into geographic processes such as transportation, urbanization, demographics, and natural or man-made hazards, to name a few. It is a form of information visualization in which principles from cartography, geographical information systems, exploratory data analysis and information visualization are integrated to facilitate exploration, analysis, synthesis and presentation of geo-referenced information [47]. An interactive geographic map is the key visual representation on top of which layers of geographic information are visualized. General guidelines for design and development of visualizations are detailed elsewhere [22, 125, 124, 126, 133, 55, 56].

In the following subsections, we present a basic visualization reference model that focuses on transforming data into visualizations. Next, we discuss design principles that help to build interactive visualization systems. Finally, we present models that describe the application of these visualization systems.

2.1.1 Visualization Reference Model

In scientific visualization, data-flow networks are used to represent the process of constructing visualizations [127, 62, 4, 110]. In information visualization, Lee and Grinstein [85] presented a conceptual model for visual database exploration, which describes the analysis process as a series of value-to-value, value-to-view, view-to-value, and view-to-view transformations. Card and Mackinlay [29], and Chi and Riedl [34] provide information visualization frameworks to facilitate the design of interactive visualization systems.

Card et al. [30] provide a basic reference model for visualization (Figure 2.1). Visualization is described as the mapping of data to visual form that supports human interaction in a workspace for visual sensemaking of data. There are three processes to support the sensemaking tasks — data transformations, visual mapping and view transformations. Data transformation maps raw data into data tables with relational descriptions of the data along with metadata. Visual Mappings transform data tables into visual structures that combine spatial substrates, marks, and graphical properties. View transformations create views of the visual structures by specifying parameters such as position, scaling, and clipping. Users can interactively change these transformations to perform their visual sensemaking tasks.

2.1.2 Visualization Design Models

To explore large volumes of data using interactive visualization, Shneiderman's visual information seeking mantra [114] —

overview first, zoom and filter, then details-on-demand

— is widely adopted in the design of interactive visualization systems. First, users are provided with an overview of data to identify global patterns, relations and outliers. Next,

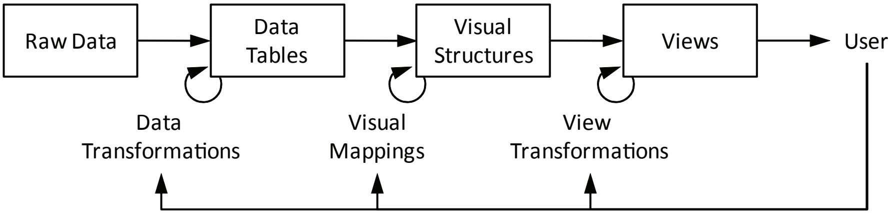


Figure 2.1: Visualization Pipeline of Card et al. [30]

they can drill down to particular areas or objects of interest, and access details of the data. During an exploration process, users may iterate these steps. It is important that visualization systems support smooth transitions between these steps. Over the years, many interaction techniques have been developed for this, including dynamic filtering [113], zoom-in and zoom-out, animation, overview + detail, focus + context (fish-eye [57], distortions [90] and table lens [104]).

Other tasks emphasized by Shneiderman [114] for effective visualization design are *relate*, *history* and *extract*. Relate allows users to view relationships between items using techniques such as linking and brushing [20]. History allows users to keep track of actions for supporting undo, replay, and progressive refinement. Extract allows users to capture data subsets or query parameters, and reuse these later in the analysis or in other computing systems. Craft and Cairns [39] provide an overview of how the visual information seeking mantra is used in visualization systems by reviewing 52 visualization papers. They found that the mantra was merely used as a guideline, and often interpreted as a prescriptive framework. Most of the current visualization systems offer limited support for history and extract tasks.

Amar and Stasko [15] provided a knowledge task-based framework for the design and evaluation of visualization systems. They argue that successful decision-making and analysis are more a matter of serendipity and user experience than of support offered by visual information seeking tasks. They identified analytical gaps for facilitating higher-level analytic tasks such as decision-making and learning in visualization. To bridge these gaps, they propose a design and evaluation framework for information visualization. Visualization systems should allow users to *determine domain parameters* (by providing facilities for creating, acquiring, and transferring knowledge or metadata about important domain parameters within a data set); to *expose multivariate explanation* (by providing support for discovery of useful correlative models and constraints); to*facilitate hypothesis testing*; to *expose uncertainty*; to *concretize relationships* (by clearly presenting what comprises the representation of a relationship and presenting concrete outcomes where appropriate); and to *expose cause and effect* (by clarifying possible sources of causation). Although this framework provides extensive details for designing a visualization system, it is not explicitly used in the implementation of current visualization systems.

Keim et al. [78] recommend that visualization can be used as a means to efficiently communicate and explore the information space when automatic methods fail. On a similar note, Van Wijk [128] calls for effective visualization design through *"visualization is not 'good' by definition; developers of new methods have to make clear why the infor-*

mation sought cannot be extracted automatically.” He presented an abstract model for visualization in which gaining knowledge through visualization is the main goal of the interactive visualization. This model of visualization is shown in Figure 2.2. Data (D) is transformed into an image (I) based on the user’s specification (S). The specification includes data, visualization and view transformations. After perceiving (P) the image, the user gains knowledge (dK/dt) and provides a new specification (dS/dt) to the visualization. Thus, the user continues to explore (E) the data by iteratively changing the specification to the visualization system. He also argues that a good visualization design has to enable users to gain positive knowledge and rapidly achieve their goals.

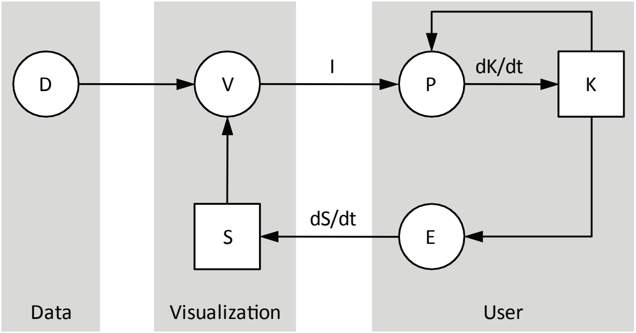


Figure 2.2: Van Wijk’s model of Visualization [128].

Recently, Munzner [95] presented a nested process model for the design and validation of visualization systems. It contains four nested layers — characterize the task and data in the vocabulary of the problem domain, abstract into operations and data types, design visual encoding and interaction techniques, and create algorithms to execute techniques efficiently. It is a prescriptive framework that helps authors of visualization papers to analyze the threats, and validate approaches possible at each level for their new visualization design.

Most of these design models for building interactive visualization systems are prescriptive in nature. Hence, these models are not extensively used to review visualization systems for supporting data analysis.

2.1.3 Application Models

Keim et al. [78] describe visualization techniques based on the goal of the visualization — presentation, confirmatory data analysis and exploratory data analysis. For presentation purposes, the facts to be presented are well known in advance. The main user task is to choose appropriate presentation techniques to effectively communicate the results of an analysis. For confirmatory data analysis, analysts have one or more hypotheses about the data as a starting point. It is a goal-oriented approach where visualization can help analysts to accept or reject these hypotheses. In exploratory data analysis, analysts search and analyze databases to find implicit but potentially useful information. They have no hypothesis about the data to start with. However, domain expertise and understanding of the data attributes are obviously very helpful.

Similarly, MacEachren [88] summaries the application of geovisualization tools for data exploration and presentation using a map-use cube (Figure 2.3). The dimensions of the interaction space are defined by three continua: from map use that is private (individual) to public (designed for a wide audience); map use that is directed towards revealing unknown (exploration) versus presenting known (presentation) information; and map use that has high interaction versus low interaction. The aim of the map-use cube is to clearly distinguish exploratory geographic visualization, which is located in the private, exploratory and high interaction corner; and map communication, which is located in the opposite corner. Nowadays, interactive visualization also plays a major role in the communication of the results of an analysis. Instead of static reports, interactive visualization based discussion blogs, for instance Many Eyes [131], and interactive dashboards in visualization systems such as Tableau [9] and Tibco Spotfire [10] have become a medium of communication, and also support collaboration processes.

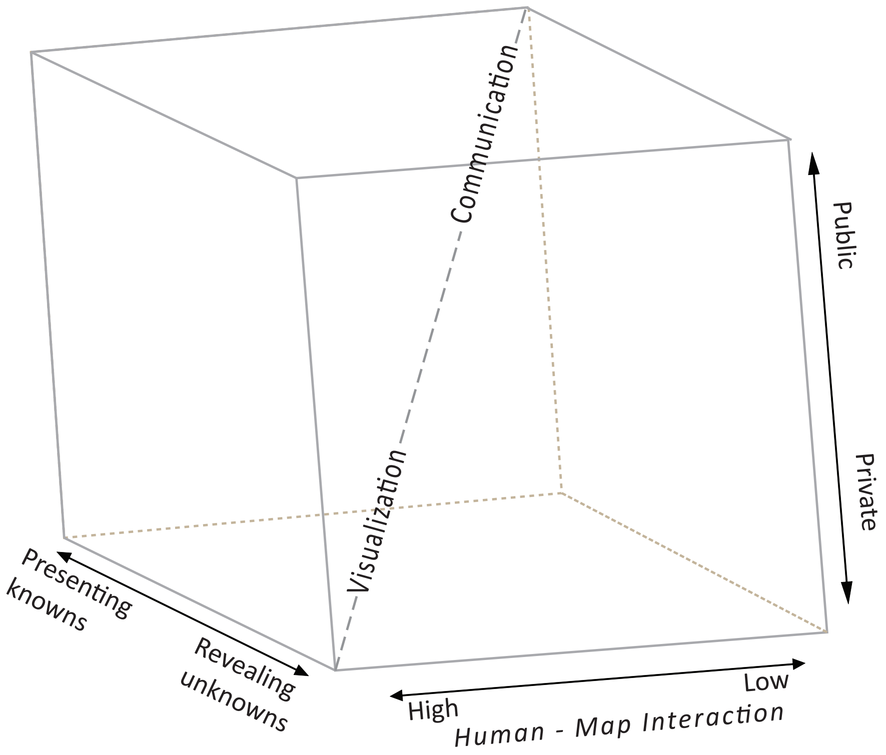


Figure 2.3: The map-use cube [88].

During a complex analysis process, large amounts of data have to be investigated in a timely manner. Though interaction techniques can come in handy to explore large datasets, it can be effective to automatically identify interesting pieces of information from large datasets and visualize these. Often visualization techniques do not scale to handle large datasets due to limitations on the amount of information that can be shown on a digital display. Automated analysis techniques such as knowledge discovery in databases, statistics and mathematics are used to analyze and extract information of interest. Although for many users automated analysis techniques remain a black box, these are a well proven approach to handle large datasets. In the next section, we introduce the field of visual analytics that combines interactive visualization and automated analysis techniques to support sensemaking of large datasets in a timely manner.

2.2 Visual Analytics

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces [122]. It is a multi-disciplinary field of research that combines techniques from information visualization, statistics, machine learning, cognitive psychology, and human factors for analyzing data. Analysts use various computing technologies to analyze data and solve problems in domains such as defense, health, governance, business and cyberspace, to name a few. During a complex analysis process, analysts need to integrate solutions obtained by investigating data using various technologies.

The definition of visual analytics claims a multi-disciplinary approach to support reasoning process. Previously, data visualization, statistics and automated data analysis were considered different approaches to solve a problem. These approaches provide different perspectives on the problem, and help users to make informed decisions. Visual analytics was developed due to the need for integrating these approaches to solve problems in a holistic manner, especially after the 9/11 terrorists attack in the USA. Following that, Jim Thomas set the research agenda for visual analytics in ‘Illuminating the Path’ [122], strongly focusing on Homeland Security in the USA. The goal of visual analytics is to facilitate the analytical reasoning process through the creation of software that maximizes human capacity to perceive, understand and reason about complex and dynamic data and situations [122]. Recently, application areas of visual analytics have been extended to fields such as health, governance, astronomy, cyber security, business and finance, to name a few. We now discuss the scope of visual analytics research and how it combines the strengths of automated data analysis and interactive visualization techniques to handle analytical problems.

2.2.1 Scope of Visual Analytics

An analysis process involves management of human background knowledge, intuition and bias in addition to data exploration. Hence, visual analytics extends beyond the combination of the fields of visualization. It can be seen as an integration of visualization, automated data analysis and human factors [78]. Figure 2.4 illustrates the scope of visual analytics. Visualization concerns the integration of methodologies from *information* *visualization, geospatial visualization,* and *scientific visualization*. With respect to automated data analysis, visual analytics furthermore profits from methodologies developed in the fields of *data management & knowledge representation*, *knowledge discovery*, and *statistical analytics*. Human factors play a key role in the analytical discourse — communication between human and computer — as well as in collaborative decision-making processes.

Finally, production, presentation and dissemination of the analysis results are important and often the most time consuming part of analysis [122]. Production is defined as the creation of materials that summarize the results of an analytical effort. Presentation involves the packaging of those materials in a way that helps the audience understand the analytical results in context using terms that are meaningful to them. Dissemination concerns the process of sharing that information with the intended audience.

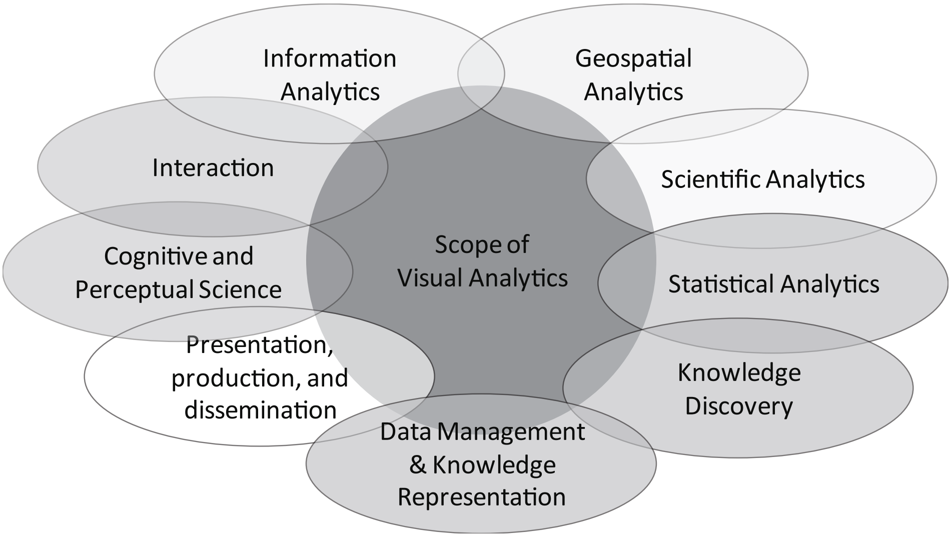

Information Analytics
Geospatial Analytics
Interaction
Scientific Analytics
Scope of Visual Analytics
Cognitive and Perceptual Science
Statistical Analytics
Presentation, production, and dissemination
Knowledge Discovery
Data Management & Knowledge Representation

Figure 2.4: The scope of Visual Analytics [78].

Depending on the problem at hand, visual analytics applications will exploit different tools and techniques from the fields of visualization, automated data analysis and human factors, to support analytical reasoning, collaboration, production, presentation and dissemination during an analysis. Initially, visual analytics was introduced for solving challenging problems that were unsolvable using automatic or visual analysis. Automatic Analysis methods can be used to solve analytical problems, in particular, when we have means for measuring and comparing the quality of candidate solutions to the problem at hand. These methods may fail when algorithms are trapped in local optima, which are unrelated to the globally best solution [79]. Visualization methods use human background knowledge, creativity and intuition to solve the problems at hand. Keim et al. [79] argue that these approaches often give good results for small datasets, however, they fail when the available data for solving the problem is too large to be captured by a human analysts. Visual analytics combines the strengths of these two methodologies to solve analytical problems. On the one hand visual analytics takes advantage of intelligent algorithms and vast computational power of modern computers and on the other hand it integrates human background knowledge and intuition to find a good solution. This potential of visual analytics is shown in Figure 2.5.

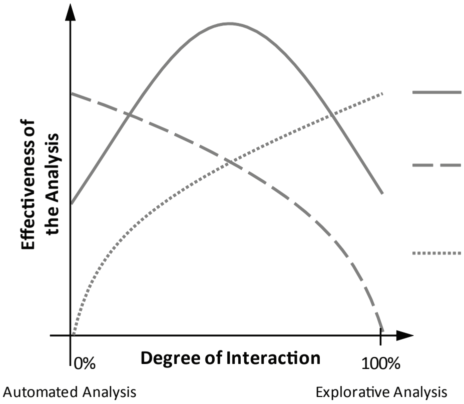


Figure 2.5: The potential of visual analytics [79].

Keim et al. [79] describe the potential of visual analytics using two problem classes: analytical problems and general application areas of IT, and three methodologies to solve these problems: Automatic analysis, Visualization and Visual Analytics. Figure 2.6 shows this scope of visual analytics in general application areas of IT. They demonstrate that visual analytics can be used to solve simpler problems that are also solvable by automatic or visual analysis means. For example, a visual tool that supports users to archive their e-mails into several folders based on content similarity, and a visual interface that displays ranking of the most relevant folders solve a task, which can be solved using traditional approaches. In these cases, visual analytics focuses on improving the effectiveness and efficiency of the reasoning process of the user, as well as the quality of the solution to a problem.

Visual Analytics gives high priority to data analytics from the start and through all iterations of the sensemaking process compared to data visualization [77]. Most research efforts in data visualization have focused on the process of producing views and creating valuable interaction techniques for a given class of data (social network, multi-dimensional data, etc.). However, there is less emphasis on how user interactions on the data can be turned into intelligence to support the sensemaking process. For instance, a system might observe that most of the user's attention concern only a subpart of an ontology (through queries or by repeated direct manipulations of the same graphical elements, for instance). Keim et al. [77] argue that this knowledge about the user's interest can be used to update various parameters by the system (trying to systematically place elements or components of interest in center view, even taking this fact into account when driving a clustering algorithm with a modularity quality criteria, for instance).

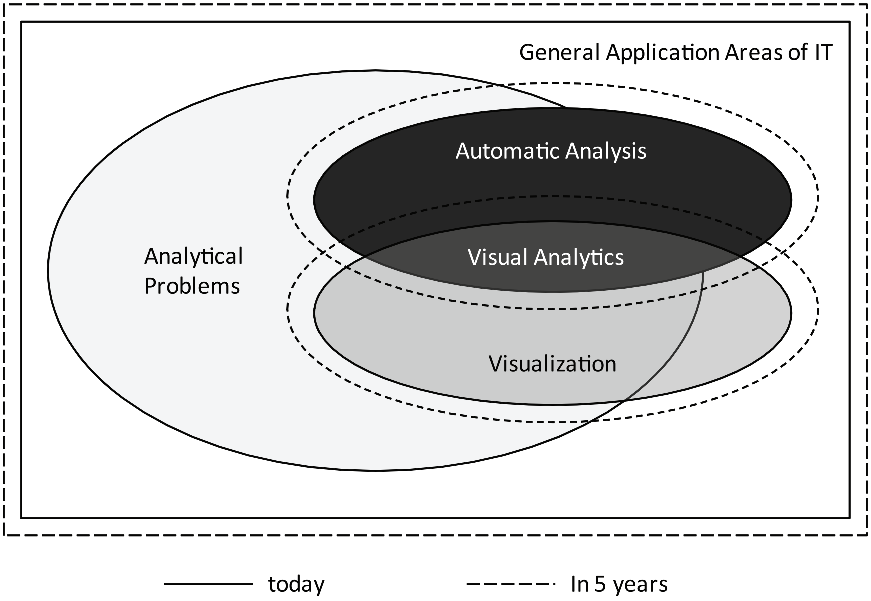


Figure 2.6: The scope of visual analytics in general application areas of Information Technology (IT) [79].

2.2.2 Visual Analytics Process

Keim et al. [78] present an insight-centric model for visual analytics. They explicitly distinguish the support offered by automated analysis methods and interactive visualization during data analysis. This model is shown in Figure 2.7. The input for the datasets used in the visual analytical process is organized from heterogeneous data sources (S) such as the Internet, newspapers, books, scientific experiments and expert systems. Insight (I) into these data is either directly obtained from the set of visualizations (V) or through confirmation of hypotheses (H) as the results of automated analysis methods such as data mining and statistics. The visual analytical process is a transformation $F : S \rightarrow I$, where F is a concatenation of functions f such as data pre-processing (D_w), hypotheses generation processes (H_V and H_S), visualization (V_H and V_S) and interactions with visualizations (U_V and U_{CV}) and hypotheses (U_H and U_{CH}).

Unlike interactive visualization, the visual analytics process often combines automatic analysis methods before and after interactive visual representations are used. This is primarily due to the fact that data sets are complex on the one hand, and too large to be visualized straightforwardly on the other hand. Therefore, a general approach recommended by Keim et al. [78] for designing visual analytics systems to support exploration of large datasets in the visual analytics is

Analyze first; Show the important; Zoom, Filter and Analyze Further; and Details-on-demand.

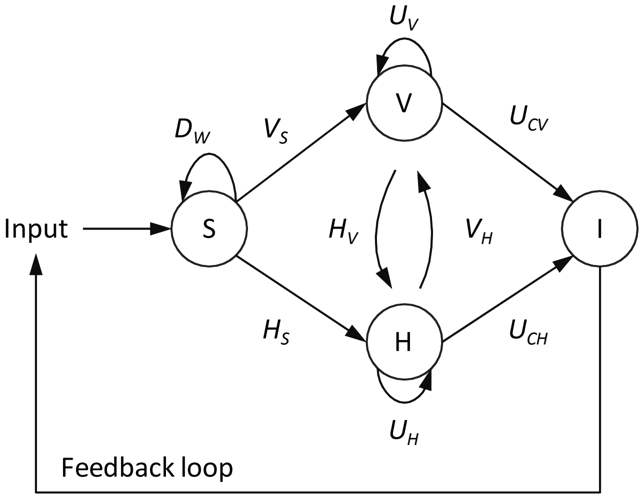


Figure 2.7: Visual Analytics Process of Keim et al. [78].

This visual analytics process model and mantra focus on designing and developing visual analytics systems for supporting the exploration process; and do not directly support the management of insights gained during data analysis. Therefore, systems based on just this process model and mantra do not enable users to review and validate their findings or analysis process, in order to support an effective reasoning process.

2.3 The Sensemaking Process

Analytical reasoning is the central part of an analysis process. Analytical reasoning involves applying human judgment to reach a conclusion from a combination of evidence and assumptions [122]. Human judgment will help to assess and understand situations, to forecast future scenarios, and to develop options [96]. Analysts pursue smaller questions related to the overall large question to be answered, and engage in the iterative refinement of procedures or parameters during the analysis. They may also refer to similar situations in past analyses to compare results: to take alternative views, or to reuse procedures. Finally, they have to identify solutions for problems in a timely manner with a decent accuracy, or limited and conflicting information.

Making judgments is the first step in the reasoning process. Subsequently, these judgments have to be revised and verified before valid conclusions are reached [73]. Often, analysts have to defend their judgment when they present it to others. They need to build knowledge structures using estimations and inferential techniques to form a chain of reasoning that articulate and defend their judgments [31]. Defending a judgment means that the reasoning, evidence, level of certainty, key gaps, and alternatives are made clear [122].

Analysis is often a collaborative process [122]. It involves analysts collaborating at the same place and time, at different places at the same time, as well as at different places and times. During an analysis, analysts use different strategies to uncover findings and make judgments. They need to effectively communicate their analysis process to defend their judgment. Analysts must be aware of what has been done and found by others for this. Unless they externalize their strategies, automatically uncovering these is a complex process. Therefore, a common ground for sharing an analysis and its results among them that promotes shared understanding has to be established. We refer to the process of creating a common ground for sharing an analysis as grounding analysis.

Clark and Barren [36] discuss eight criteria for creating effective common grounds for sharing information among people across different media. They are *copresence* (can see the same things), *visibility* (can see each other), *audibility* (can hear each other), *cotemporality* (messages received at the same time as sent), *simultaneity* (can both parties send messages at the same time or do they have to take turns), *sequentiality* (can the turns get out of sequence), *reviewability* (can they review messages, after they have been first received), and *reviseability* (can the producer edit the message privately before sending). Now with the advent of collaboration support tools such as video conferencing, workspace sharing and discussion forums, to name a few, most of these criteria are well supported. The criteria most relevant for collaborative analysis process are the reviewability and reviseability of the analysis process, analysts' strategies and their findings for grounding their analysis, and defending their judgments. Therefore, analysts must be enabled to perform three activities while making sense of data during an analysis — to make judgments, to ground their analysis, and to defend these for collaborative analysis. These activities are summarized in Figure 2.8.

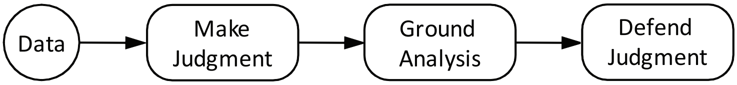


Figure 2.8: A model for analyst's sensemaking activities during an analysis process.

To understand the requirements for supporting the sensemaking process in visual analytics during an analysis, we first take a close look at the sensemaking model of Pirolli and Card [103] for intelligence analysis, which was derived from a cognitive task analysis. They present a data flow where raw data is transformed into reportable results (Figure 2.9). External data sources contain the raw evidence, largely text data. The shoebox is the much smaller subset of that external data that is relevant for processing. The evidence file contains snippets extracted from items in the shoebox. Schemas are derived by re-representing or organizing information from evidence files, and help to draw conclusions. Hypotheses are the tentative representations of those conclusions with supporting arguments. Finally, the conclusions and hypotheses are presented.

In this analysis process, there are two major activities: the information foraging loop and the sensemaking loop. In the information foraging loop, analysts seek information, search and filter it, and read and extract information possibly into some schema [102]. In the sensemaking loop, they iteratively develop a mental model (a conceptualization) from the schema to support a claim [107]. In these activities, Pirolli and Card identify two

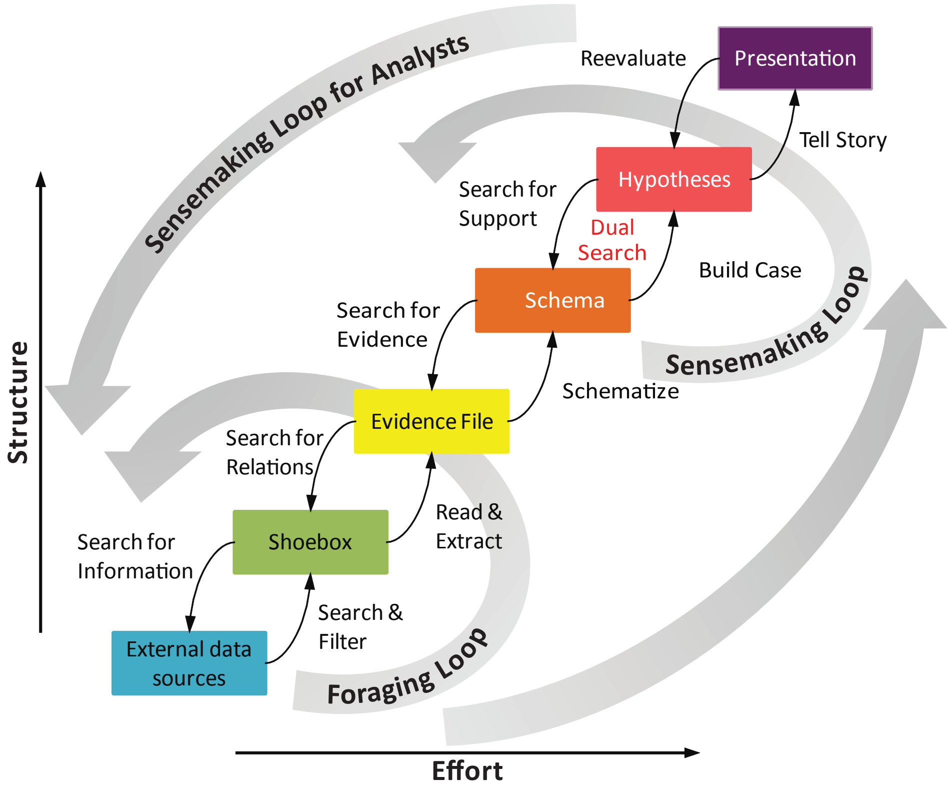


Figure 2.9: The Sensemaking Model for Intelligence Analysis, Pirolli and Card, [103].

processes: a bottom-up process (from data to theory) and a top-down process (from theory to data). They found that analysts opportunistically mix the two processes. The bottom-up process involves search and filter raw data; read and extract information; organize information into schemas; build a case; and tell a story to some audience. The top-down process involves re-evaluation of feedback from the audience; search for support from schema; search for evidence and relations in evidence files; and again search for information from the raw data.

In interactive visual data analysis, many tools and techniques have been developed that focus on the foraging activity. The visualization pipeline model and visual analytics process model focus on exploring and gaining insight into data. However, little support is offered by visual analytics systems to capture findings (into evidence files), organize these findings (into schemas), construct arguments to validate hypotheses, and present these. Hence, we argue that for supporting the sensemaking process in visual analytics during an analysis, the user must be enabled to

- perform both foraging and sensemaking activities; and

- carry out bottom-up and top-down processes during these activities.

2.4 Supporting the Sensemaking Process

In the following, we describe our approach to support the sensemaking process in visual analytics (Figure 2.10).

Data (D) is *transformed* into information (I) based on the users' specification (S). I includes automated data analysis results, text summaries and visualizations. They gain *knowledge* (K) by *reasoning* with I, and continue to *explore* until the analysis goal is reached. During a long analysis session, they may not keep track of all the interesting knowledge. Therefore, the system automatically *captures* S and I, and *archives* these as an *action trail*. An action trail contains a sequence of S, specified by the users during interactive data exploration. Also, they can manually *externalize* and archive *findings* (F), such as notes, schemas, entity-relationships and images, during the exploration process.

Later, users can *review* S, I and F of past analyses. For this, the system automatically provides interactive *overviews* of the past analyses. Also, they can *search and retrieve* specific S, I and F from the archive. Next, they can *reuse* S from a past analysis in the current analysis. During the review process, they can also obtain new findings, or edit the previous findings. Finally, they can *share or present* their analysis process and findings to others. The archive can be synchronously or asynchronously accessed to support collaborative analysis. In summary, we argue that a visual analytics system should meet the following requirements. Users must be enabled:

- to automatically capture and manually externalize the interesting aspects of the analysis;
- to review the analysis process and the findings using overviews of the analysis; and to search & retrieve specifications, processed information and findings; and
- to reuse, share, and present the interesting aspects of the analysis.

2.5 State of the Art

In this section, we review a number of visualization and visual analytics systems, based on the requirements for supporting the sensemaking process in visual analytics: capture; review; reuse, share and present interesting aspects of the analysis process. Table 2.1 provides an overview of widely used visual analytics systems and their support for the sensemaking process.

Vistrails is a popular scientific workflow management system [19]. It supports the creation of data flow diagrams by composing various scientific visualization operators. It captures changes to a workflow using a history tree representation. Users can query for workflows from history, and review and reuse them [109]. They can reuse workflows for different sets of parameters, reuse visualizations across different data and compare the

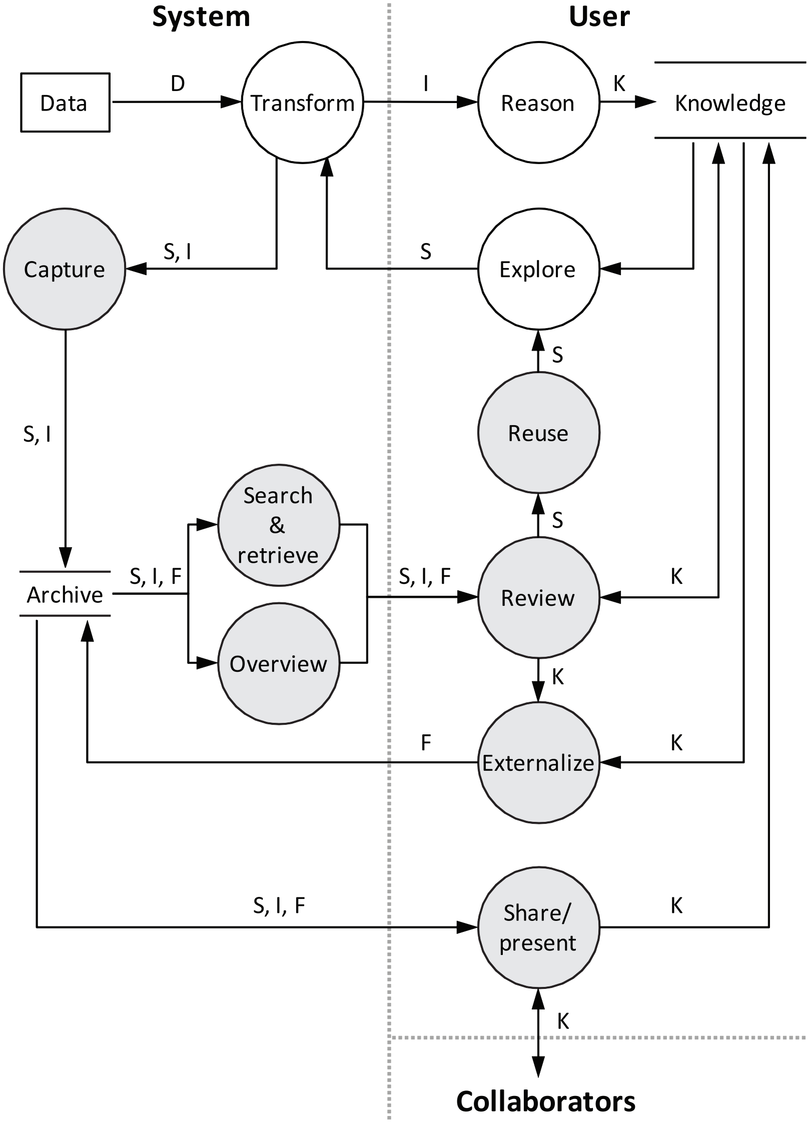


Figure 2.10: A model for supporting the sensemaking process in visual analytics. The items in orange highlight requirements for supporting the sensemaking process in visual analytics. D: Data; I: Information; S: Specification; K: Knowledge; F: Findings.

different visualizations by arranging them side-by-side. It supports real-time collaborative design of workflows [50].

Generally, most of the information visualization tools such as Improvise [134] and Jigsaw [119], to name a few, focus on interactive data exploration; and offer limited support to capture interesting aspects of the analysis process, for instance by taking screenshots. Visual Analytics Inc.'s VisualLinks and DataClarity [11], and Magnaview [5] support bookmarking visualizations and sharing these visualizations with collaborators through the Internet. General Dynamics's CoMotion Discovery, CoAction and Command Post of the Future [1] enables users to annotate and record notes over a visualization workspace and synchronously share them. Tibco Spotfire [10] supports capturing visualizations with annotations and sharing these on the Internet. Sense.us [67] is a web-based asynchronous collaborative visualization system that supports users to annotate and share visualizations. It also enables users to review notes and have discussion on visualizations, similar to IBM's ManyEyes [131]. Tableau [9] enables users to share visualizations with annotation through a web based interactive dashboard. During an analysis, users can also capture sets of objects as computed sets, and reuse these.

Often visual analytics systems have to handle unstructured data such as documents and email corpus, news stream and blogs, to name a few. Analysts are interested in extracting entities, events and their relationship from these data. Visual analytics systems such as Oculus Info [6], Xerox Parc's Entity Workspace [24] and i2 Analyst's Notebook [3] support analyzing large collections of unstructured data. Entities and their relationships are automatically extracted. Users can edit them and reuse them to find similar entities and documents from the archive.

Oculus Info (nSpace and Geotime) helps users to manage entities and create stories based on visualizations, entities and notes for sharing the analysis results. Xerox Parc's Entity workspace supports evidence marshalling using the entity graph. During collaborative analysis, it helps analysts to identify entities of mutual interest. In addition to entity-relationship, the Analyst's Notebook supports analysts to capture, review, reuse and share events and domain-specific knowledge. X-media project [40, 41], a knowledge management system, captures a domain-specific ontology in a distributed analysis environment. Users can interactively explore the ontology using knowledge lenses and graphs during an analysis. They can review, reuse and share the ontology during an analysis. Most of these systems help analysts to capture findings for sharing and presentation purposes; they do not capture the analysis process. So, they do not enable their users to revisit and review analysis process. Hence these systems do not directly support the sensemaking process during data exploration.

Very few visual analytics systems capture both the analysis process and the findings. HARVEST [59], a web based visual analytics system, captures the analysis process as action trails. While interactively visualizing data, users can record notes, which are captured as a part of the action trail. An action trail is archived only when users bookmark a visualization state. It does not maintain an integrated action trail of the entire analysis process. A list of bookmarks is shown to the users. They can revisit and reuse action trails via the bookmarks list. Palantir's Government and Finance [7] captures action trails, entity relationships and events during an analysis; and users can share annotated action trails for collaborative analysis. Analysts can do keyword search to retrieve action trails; also they can edit and combine different action trails. However, they cannot get an overview of what has been and found during the analysis process. PNL's Scalable reasoning system [101] aims to support teams of collaborating analysts to capture, share, and reuse analysis processes and their reasoning strategies through a combination of desktop and mobile environments. This is currently a work-in-progress. Though these visual analytics systems capture both the analysis process and findings, they do not offer enough support for the users to get an overview of the archived analysis processes and findings for an effective sensemaking process.

2.6 Research Scope

The workflow model, described in Section 2.4, to support the sensemaking process in visual analytics is developed based on Pirolli and Card's sensemaking model for intelligence analysis. The workflow model contains four key processes: capture, reuse, review and share of interesting aspects of a data exploration to support the sensemaking process. These processes may require different sets of tools and techniques for handling different interesting aspects concerning the analysis processes and findings.

In this dissertation, we describe generic models and tools to support the sensemaking process in visual analytics during an analysis. We begin by looking at a simpler problem and try to show that the quality of results and the effectiveness of the reasoning process can be improved by supporting the four sensemaking tasks: capture, reuse, review, and share. For this, we consider a simple interactive visualization tool consisting of visualizations such as scatterplots and barcharts attached with dynamic query interface. We apply the generic models and tools which we developed to support the sensemaking process on this visualization tool. We have implemented these models and tools using Aruvi, a research prototype. Some of these models are implemented in HARVEST during a collaborative research work.

We enable users to capture interesting aspects such as action trails, objects of interest, selections and notes during interactive data exploration; and provide users tools to gain overview of the analysis process and findings, and effectively review and reuse these during the analysis process. In the future, other interesting aspects of the exploration process can emerge that are useful for supporting the sensemaking process. We believe that the models and tools described in this dissertation can be used as a starting point for effectively capturing, reviewing, reusing and sharing such new interesting aspects.

2.7 Evaluation

Evaluation in visual analytics is challenging and notoriously hard. The visual analytics research agenda [122] identifies three levels that can be considered for evaluation: component, system, and work environment. At the component level, the evaluation focuses on analytical algorithms, visual representations, interaction techniques, and interface design. At the system level, visual analytics combines multiple components to support an analytical reasoning process. An evaluation at the system level can be done by comparing with the technology currently used by the target user. At the work environment level, the evaluation focuses on technology adaptation and productivity.

Plaisant [105] identifies three main methods for user centered evaluation in information visualization: controlled experiments, usability evaluation and case studies. In controlled experiments a novel visualization system is compared with the state of the art to determine if it performs better. Since the work presented in this dissertation is empirical and significantly different from techniques discussed in Section 2.5, direct comparison to these existing techniques is not possible. Usability evaluation provides feedback on the problems encountered while users interact with a system. The system is evaluated based on the accuracy or efficiency of the users completing certain tasks [112]. Usability evaluation was difficult to apply, as it is difficult to create generalized sensemaking tasks and analysis goals to enable comparison of users' feedback. Case studies involve studying the feasibility of tools in a real-use context, that is, real users performing real data analysis in their work environment. The advantage of case studies is that they report on users in their natural environment doing real tasks, demonstrating feasibility and in-context usefulness. The disadvantage is that they are time consuming to conduct, and results may not be replicable and generalizable [105].

We primarily used case studies approach to study the implications of new tools for supporting the sensemaking process. In particular, we used our prototype as a *technology probe* that exposes users to new ideas and then use this as the means to obtain qualitative feedback. A technology probe involves installing a technology into a real use context, watching how it is used over a period of time, and then reflecting on this use to gather information about the users and inspire ideas for new technologies [69]. It is not just a prototype, but a tool to help to determine which kinds of technologies would be interesting to design in future. Users can adapt to the new technology in creative new ways for their analysis process [89].

In chapter 3, we present a sensemaking framework based on an empirical approach starting by closely looking at models presented in Figures 2.1, 2.2, 2.9, and 2.10. We evaluated the framework by deploying Aruvi as a technology probe in the real use context and gathering analysts' feedback. Then we analyzed the usage pattern and analysts' feedback to check if the sensemaking framework is useful during an analysis. Also, we encountered some new issues related to supporting the sensemaking process in visual analytics. Subsequent chapters address three of the many issues identified.

Table 2.1: An overview of visual analytics systems for supporting sensemaking process.

		Capture		Review		Reuse		Share and Present	
No.	**Products**	**Analysis**	**Findings**	**Analysis**	**Findings**	**Analysis**	**Findings**	**Analysis**	**Findings**
1	Jigsaw [119], Improvise [134]								
2	MagnaView [5]		✓ *(V)*						✓ *(V)*
3	General Dynamics (CoMotion Discovery, Command Post of the Future, CoAction) [1]		✓ *(N and V)*		✓ *(N)*				✓ *(N and V)*
4	Sense.us [67]		✓ *(A, N and V)*		✓ *(A, N and V)*				✓ *(A, N and V)*
5	Tableau [9]		✓ *(A, V, and S)*		✓ *(A, V, and S)*		✓ *(A, V, and S)*		✓ *(A and V)*
6	Visual Analytics Inc. (VisualLinks, DataClarity) [11]		✓ *(V)*						✓ *(V)*
7	Tibco SpotFire [10]		✓ *(A and V)*						✓ *(A and V)*
8	OculusInfo (nSpace, GeoTime) [6]		✓ *(N and ER)*						✓ *(N)*

***V** - Visualizations; **N** - Notes; **A** - Annotations; **S** - Set of Objects; and **ER** - Entity Relationship.*

Table 2.1: An overview of visual analytics systems for supporting sensemaking process.

		Capture		Review		Reuse		Share and Present	
No.	**Products**	**Analysis**	**Findings**	**Analysis**	**Findings**	**Analysis**	**Findings**	**Analysis**	**Findings**
9	Entity Workspace [24]		✓ *(N and ER)*		✓ *(N and ER)*		✓ *(ER)*		✓ *(N and ER)*
10	I2 Analyst's Notebook [3]		✓ *(N, ER, E, and V)*		✓ *(N, ER and E)*		✓ *(ER)*		✓ *(N, ER, E, and V)*
11	X-Media Project [40, 41]		✓ *(O and N)*		✓ *(O and N)*		✓ *(O)*		✓ *(O and N)*
12	Vistrails (VisTrails, VisIt, Paraview) [19, 50, 109]	✓ *(W)*		✓ *(W)*		✓ *(W)*		✓ *(W)*	
13	Palantir (Government and Finance) [7]	✓ *(T)*	✓ *(A, ER, E, and S)*			✓ *(T)*	✓ *(A, ER, E, and S)*	✓ *(T)*	✓ *(A)*
14	HARVEST [59, 115]	✓ *(T)*	✓ *(N)*	✓ *(T)*		✓ *(T)*	✓ *(N)*	✓ *(T)*	✓ *(N)*
15	Aruvi [117, 116]	✓ *(T, Z and S)*	✓ *(N)*	✓ *(T, Z and S)*	✓ *(N)*	✓ *(T, Z and S)*	✓ *(N)*	✓ *(T, Z and S)*	✓ *(N)*

V *- Visualizations;* ***N*** *- Notes;* ***A*** *- Annotations;* ***ER*** *- Entity Relationship;* ***E*** *- Events;* O *- Ontologies;*
S *- Set of objects;* ***Z*** *- Selection;* ***W*** *- Workflows; and* ***T*** *- Actiontrails.*

The interesting aspects of the exploration process investigated in this dissertation are highlighted in yellow.

Chapter 3

A Sensemaking Framework for Visual Analytics

> *The goal of mankind is knowledge. Now this knowledge is inherent in man. No knowledge comes from outside; it is all inside. What we say a man 'knows', should, in strict psychological language, be what he 'discovers' or 'unveils'; what man 'learns' is really what he discovers by taking the cover off his own soul, which is a mine of infinite knowledge.* — Swami Vivekananda.

For effective analytical reasoning during data analysis, analysts need an integrated analysis framework which enables them to capture interesting aspects of the exploration process; and review, reuse and share these. In this chapter, a new visual analytics framework is presented by considering analytical reasoning in general, as well as in combination with visualization. Using this framework, analysts can capture and review the analysis, validate the findings and revise them. They can also organize the findings to build a case. The analysis process can be saved and presented to others along with the findings. We have developed a prototype, Aruvi, based on this framework. A user study is presented to evaluate the perceived usefulness of the framework, using Aruvi. Following that, two case studies are presented.

3.1 Introduction

As discussed in the previous chapters, the grand challenge in the visual analytics research agenda [122] calls for developing interactive visual interfaces to perform data analysis as well as structured reasoning. This includes the construction of arguments, convergent-divergent investigation and evaluation of alternative hypotheses. The fields of visualization, automated data analysis and human factors are used to build the interactive visual interface. Information visualization tools and techniques act as a frontend to automated data analytics: to provide input and to analyze its output; in addition to supporting interactive data exploration using abstract visual representations [30]. During interactive visualization, users can encounter many discoveries in terms of relations, patterns, outliers and so on.

Sensemaking involves seeking information, organizing and analyzing it, and possibly forming new knowledge and informing further action [30]. Pirolli and Card [103] organize the sensemaking process of analysts into two major loops: the information foraging loop and the sensemaking loop as discussed in section 2.3; where analysts opportunistically mix the two loops for effective analytical reasoning. In practice, the support of visualization tools for the sensemaking process is limited to the information foraging loop, and the sensemaking loop has to be done in the analyst's mind. It is difficult for the human working memory to keep track of all findings. Hence, synthesis of many different findings and relations between those findings increases the cognitive overload [103]; and thereby hinders the reasoning process. In the following section, we derive requirements for developing such a framework by considering analytical reasoning in general, and then in combination with visualization.

3.2 Analytical reasoning - a close look

The analytical reasoning process is often not a systematic process. Information foraging in information visualization can be described as navigation through an information space facilitated by various interactions such as dynamic query [113], overview + detail [106], direct manipulation [35], focus + context [57] and so on. These interactions enable the analyst to view the data in different ways during the exploration process. The exploration evolves based on the analyst's prior knowledge, and clues or findings in each visualization state. It is similar to berry picking [17] in which the evolution of the navigation is opportunistic, and information is gathered in bits and pieces. In this context, the knowledge creation process is unsystematic, continuously evolving and emergent [23]. Hence, analysts must be aware of what has been done and found during the exploration process to perform effective reasoning.

During data analysis, analysts looks for evidence from the data to construct, confirm or contradict a claim. Based on the relations that the evidence has with the data in context of the analysis' purpose, their mind constructs mental models of the information structure [73]. In the context of interactive information visualization, the evidence can be found in terms of patterns or outliers by changing visualization and data specifications. If the argumentation process is complex, it is important to externalize the evidence and causal links between them for effective reasoning [99].

To further understand the requirements for the analytical reasoning process during visual data analysis, we looked at traditional well-founded reasoning theories. Johnson-Laird and Byrne [74] observed that there are three basic stages in different reasoning theories such as spatial reasoning, propositional inferences, syllogisms, and so forth. They are *model construction*, *revision* and *falsification*. In the first stage, the argument premises are understood and mental models are constructed based on the premises' content. In the second stage, the model is revised to formulate a putative conclusion. In the third stage, alternate models are searched for to contradict the putative conclusion. If there are no alternative models, the conclusion is accepted; otherwise, the analyst returns to the second stage to assert the validity of the other conclusions against the alternate models. Therefore, it is clear that externalization of the mental models is not enough to support the entire reasoning process. The two other analytical reasoning phases — revision and falsification — also have to be supported.

Further, it is important to communicate what has been found during the exploration process to others for a collaborative decision making process. Viégas and Wattenberg strongly argument for communicating insights along with visualization to others through their communication-minded visualization framework [130]. Further design considerations for sharing insights in collaborative visual analytics are discussed by Heer and Agarwal [64].

From the above discussion, we set out the following requirements for a visual analytics framework to support the analytical reasoning process. An analyst has to be enabled to:

1. externalize the analysis artifacts such as evidence, hypotheses, assertions and causal links between them;
2. organize the analysis artifacts and the causal links between them to support or contradict a claim;
3. review and revise the exploration process;
4. link externalized analysis artifacts and visualizations to support these;
5. present his findings along with his analysis process to others.

In summary, for an effective reasoning process, the user must have an overview of what has been done and found. Therefore, to keep track of the exploration process and insights, a history tracking mechanism and a knowledge externalization mechanism respectively are essential. Hence, to support the analytical reasoning process in visual analytics, a framework with both a history tracking mechanism and a knowledge externalization mechanism is required.

3.3 Related work

We now present previous work in history tracking and in knowledge externalization.

3.3.1 History Tracking

A common approach to automatically record the exploration process is to capture low-level user actions such as mouse events, keyboard events and to provide a linear history. The user can revisit the linear history using an undo-redo mechanism. It is used for recovery and reversal operations [12]. On performance of a new action by the user after backtracking, the recent forward actions are deleted. Hence, the complete navigation is not captured.

Another approach is to use a tree structure to capture the exploration process. In GRASPARC [25], a problem solving framework which integrates the computation and visualization process, a history tree is used to model the search for an optimal solution to numerical simulations. The nodes of the history tree hold snapshots of the parameters, raw data and image representations at various stages of the analysis, and edges represent the user navigation. It allows the user to select a snapshot as a new branch point, or to select a sequence of snapshots for visualization.

In image-graphs [87], a graph representation is used to capture the parameter settings during visual data exploration. The edges hold parameters, nodes display the resulting images. The user can perform operations on the edges and nodes to produce new visualizations. Since the image graph is a parameter-based interface, the rate of growth of parameter settings makes it difficult to display and compare resulting images. A branching time model is used in Visage [44] to capture direct manipulation tasks during visual data exploration. A time-travel interface is used to visualize the branching time model that allows the user to revisit the analysis and reuse a sequence of direct manipulations on a new branch timeline.

In scientific visualization, there is a growing interest in the management of scientific data and the visualization process. Often, the scientific data changes during analysis and the specifications of the visualization pipeline have to be tweaked for accurate results. VisTrails [19], a scientific visualization workflow system, allows the creation and maintenance of visualization pipelines, and optimizes their execution.

The models described above provide solutions for backtracking visualization states using history or workflow mechanisms. However, they do not enable analysts to capture their reasoning while viewing the data.

3.3.2 Knowledge Externalization

The scope of information visualization tools is often limited to interactive visualization to explore the data, and little support for information synthesis for analytical reasoning is offered. Sometimes, users can annotate interesting patterns or objects in the visualization. Annotations can be attached to hand-drawn marks that are used to highlight interesting patterns or objects, for instance, encircling a region in the visualization. Denisovich [42] uses hand-drawn marks on top of a map to select objects, similar to lasso selection, and attaches annotations to them. The user can access the findings from the annotations list. Ellis and Groth [49] use annotations to share discoveries in their collaborative data visualization environment. The annotations are stored in a separate layer on top of the data and enable expression of free thoughts. Often, annotations are used as attention pointers to the synthesized information. If the number of annotations on top of the visualization is large, it is difficult to express relations among the annotations.

In Harvest [61], knowledge is represented as concept instances. A concept is described using a data ontology based on type, parent type and user-defined attributes. Users can create a new concept or collect evidence to an existing concept. The concepts can be modified, merged, or removed. The links between the concepts and evidence are maintained by the synthesis manager based on the data ontology. The synthesized knowledge is visualized using a graph-like structure in the synthesis space. Sandbox [139] allows analysts to jot down hypotheses and evidence using a white board metaphor within the TRIST framework [75]. They can save references to any relevant information, including documents, snippets, images, tables, etc.. Concept maps are automatically generated by

Sandbox based on the text-to-concept map algorithm. Harvest and Sandbox offer support for evidence marshalling. However, they do not associate synthesized concepts with any visualization. It is not possible to review the synthesized concept using the corresponding visualization that leads to the finding.

In sense.us [67] a discussion forum is used to express opinions on visualizations. Users can share their findings or free thoughts by starting a new thread or adding to an existing thread along with a link to the visualization. The threads within the discussion forum are independent and do not provide an overview of the causal links between the findings shared in the discussion forum. While performing complex analysis, it becomes difficult for the human working memory to maintain causal links between various propositions in the discussion forum [21]. Jigsaw [118], a visual analytics system, helps to visualize connections and relationships between entities extracted from document collections. It has a shoebox that enables users to capture entities and documents, to record hypotheses and to organize these into groups. Visualizations in Jigsaw can be bookmarked, and linked to the items in the shoebox.

In summary, current visualization systems help to capture annotations on top of visualizations; and interesting findings such as notes, hypotheses, entities and their relationship in the shoebox. Only Sense.us and Jigsaw maintain links between findings and visualizations. However, these systems do not capture the analysis process along with bookmarked visualizations. Hence, it can be difficult to review items in the shoebox within the context of the exploration process. Hence, they do not meet the requirements for supporting the analytical reasoning process.

3.4 Approach

To satisfy the requirements for the analytical reasoning process through information visualization, we argue that the user has to be provided with three different types of visual representations:

- Data view: visual representation(s) of the data;
- Navigation view: visual representation(s) of the exploration process;
- Knowledge view: visual representation(s) of the analysis artifacts and their causal links.

The data view consists of interactive information visualization tools. The navigation view provides an overview of the exploration process by capturing the visualization states automatically. The knowledge view enables users to record their analysis artifacts and the causal links between them. They can also organize the analysis artifacts in the knowledge view to build a case to support or contradict an argument. They can establish a link between an analysis artifact in the knowledge view and a visualization state in the navigation view. Hence, they can revisit a visualization state from both navigation and knowledge views to review their analysis and validate their findings. After revisiting the visualization state, they can reuse it to look for alternative views. Thus, the three phases of the analytical reasoning process (model construction, revision and falsification) are supported.

This information visualization framework for supporting the analytical reasoning process is shown in Figure 3.1.

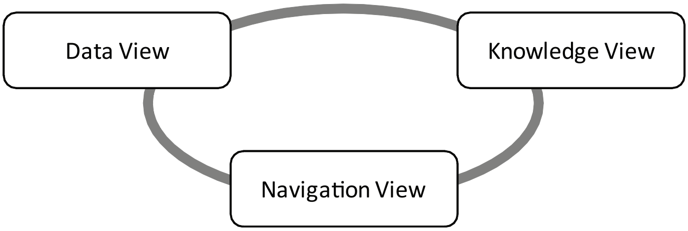


Figure 3.1: A visual analytics framework for supporting the analytical reasoning process.

A key feature in this framework is that it allows the user to establish a link between the externalized knowledge artifact in the knowledge view and a particular visualization state *asynchronously*. A visualization state can be associated with more than one analysis artifact in the knowledge view. In the following sections, we describe the components of the framework in detail.

3.4.1 Data View

The data view is a container for interactive information visualization tools to explore the data. It has two components: visual representations and interactions. Visual representations can vary from a single visualization to multiple visualizations depending on the nature of the data and the analysis. Often, the data is large and complex such that static visualizations fall short. Hence, interactions are needed to explore the data by modifying the data transformation, visual mappings and view transformation [30].

An *interaction interface* is defined as an interface that translates user actions such as mouse events, key events and other input events into visualization specifications. These interaction interfaces enable the user to apply changes to various stages in the visualization pipeline [30] such as the data organization, data filtering, data mapping onto visual representations and displaying these to the user. A dynamic query interface is used for specifying data filters. A visual mapping interface is used to specify transformations from data to visual representations, for instance to change shape encodings and color maps, and to reconfigure axes. Direct manipulation is used to select objects for tracking or emphasis. A view settings interface is used to change camera parameters, overview and details, panning and zooming. These interfaces are some examples of how interaction interfaces help to specify visualizations interactively and to explore complex datasets rapidly.

3.4.2 Navigation View

The navigation view provides an overview of the exploration process by capturing the visualization states automatically. We now describe a history tracking mechanism to capture the visualization states automatically.

In interactive visualization, the dataset D is transformed into an image I based on a specification S [128]. S includes visualization methods, attribute filters, graphical filters applied through direct manipulation, color mappings, clustering and so forth. The user provides the specification S_t to the system based on the current knowledge K_{t-1} to generate the image I_t. K_t is the total knowledge gained by the user. The user repeats the process of generating a new image I_{t+1} by providing a new specification S_{t+1} based on K_t, until the desired results are achieved. Thus, the user navigates through the data by changing S. Figure 3.2 shows the visualization state in the user navigation at time t.

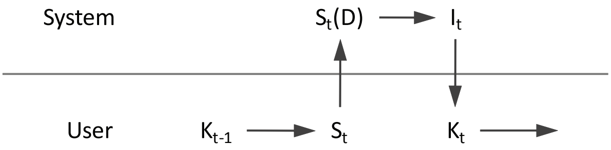
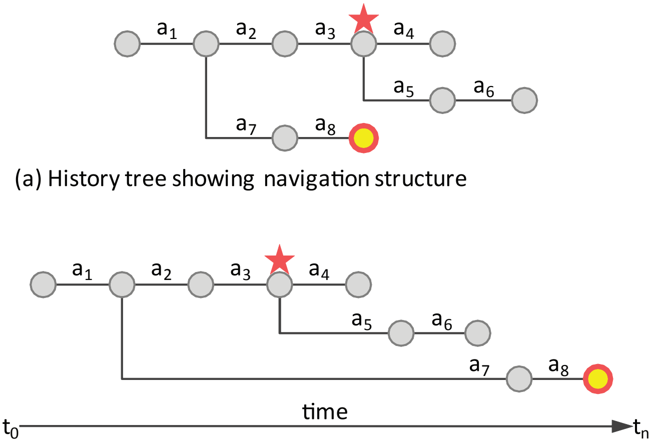


Figure 3.2: The visualization state in the user navigation at time t.

A new visualization state is recorded automatically when the visualization specification is changed via an interaction interface. This allows users to roll back to previous visualization states. When a visualization state is revisited, the image I is regenerated based on S and D of that state. Then, they can reuse the revisited visualization state by changing S and D. This creates a new branch, resulting in a tree structured navigation path that is similar to a history tree representation [25].

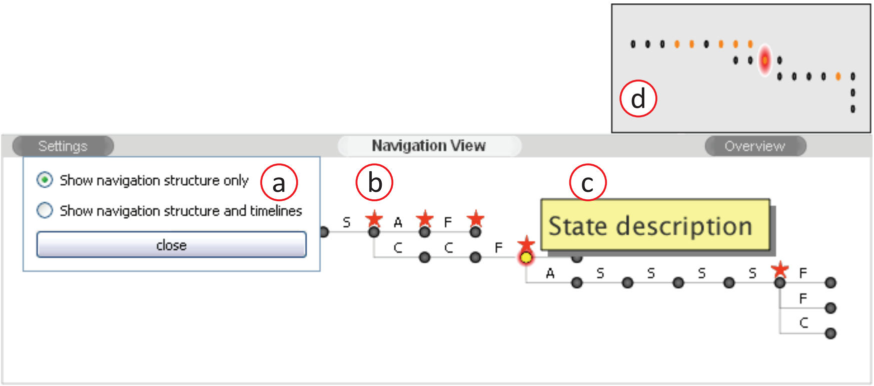


(a) History tree showing navigation structure

(b) History tree showing navigation structure ordered by time

Figure 3.3: The navigation view.

Initially, we used a history tree representation to show the structure of the exploration process where nodes represent visualization states, and edges between adjacent nodes are labeled with the user action (see Figure 3.3(a)). The history tree is drawn using a



Settings

◉ Show navigation structure only

○ Show navigation structure and timelines

close

Figure 3.4: An implementation of the navigation view using the history tree representation. (a) The settings interface. (b) A node marked with a star that represents a visualization state with links to objects in the knowledge view. (c) The visualization state description. (d) An overview of the exploration process captured by the navigation view.

horizontal-vertical tree layout. A new node is appended to the tree at the right in the horizontal direction. A new branch is created below existing ones in the vertical direction. To avoid cluttering between edges, a right heavy horizontal-vertical layout algorithm is used [123]. Figure 3.3(a) shows the structure of the navigation. A branch represents a revisit and reuse of an already existing visualization state.

To understand the temporal context, it is important to see the sequence of visualization states along with the structure of the navigation. Figure 3.3(b) shows the structure of the navigation ordered by time in the horizontal direction. Users can toggle between the two representations during the analysis via the *settings* interface (see Figure 3.4(a)). They can revisit the visualization states sequentially in the order of creation using back and forward arrow keys. This action is similar to the undo-redo mechanism. Also, they can hover over a node to get information about the visualization state (see Figure 3.4(c)) and jump to any visualization state in the navigation view. An overview window is used for panning over the history tree (see Figure 3.4(d)).

When a visualization state is linked to objects in the knowledge view, it is marked with a star in the navigation view (see Figure 3.3 and Figure 3.4(b)). The current visualization state in the navigation is highlighted in yellow.

3.4.3 Knowledge View

According to Larkin and Simon [84], a diagrammatic representation is better than a sentential representation for searching and recognizing concepts and their relations. The use of an appropriate diagram helps analysts to make all the possibilities explicit and reason more rapidly and accurately [18]. Based on these premises, numerous diagramming techniques such as mind maps, concept maps, cognitive maps, affinity diagrams, causal maps, and so forth have been developed to facilitate the reasoning process [26]. However, in

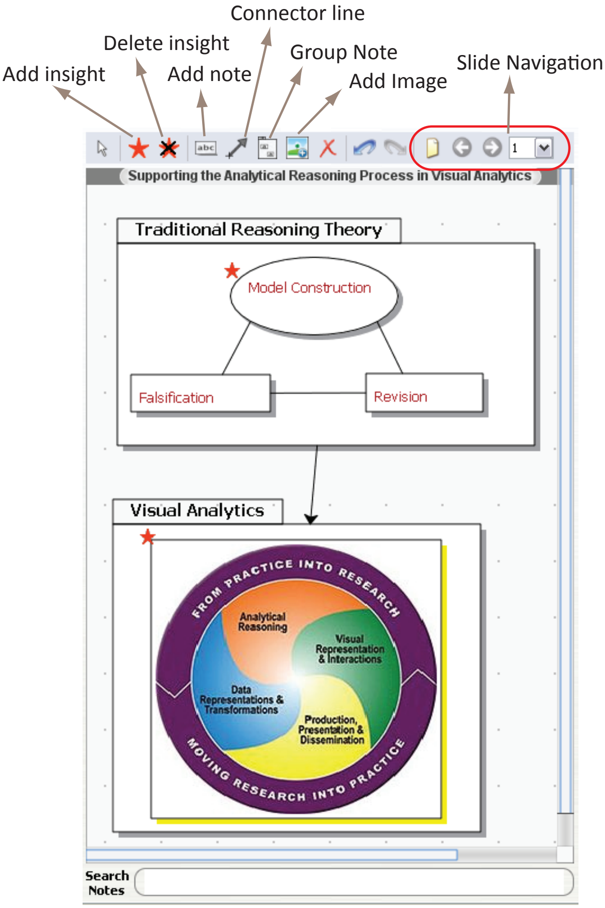

Connector line
Delete insight
Add insight
Add note
Group Note
Add Image
Slide Navigation
Supporting the Analytical Reasoning Process in Visual Analytics
Traditional Reasoning Theory
Model Construction
Falsification
Revision
Visual Analytics
FROM PRACTICE INTO RESEARCH
Analytical Reasoning
Visual Representation & Interactions
Data Representations & Transformations
Production, Presentation & Dissemination
MOVING RESEARCH INTO PRACTICE
Search Notes

Figure 3.5: The knowledge view.

some cases just placing the concepts next to each other in some meaningful order will already be sufficient. Hence, a knowledge view should be a flexible environment for analysts to structure the analysis artifacts according to their thought process. We therefore have chosen to design the knowledge view as a basic graphics editor. The knowledge view is shown in Figure 3.5. It helps the users to construct diagrams to externalize their mental models and structure arguments.

A note is the basic entity to record the findings. A note is shown as a rectangle or an ellipse with centered text. A bitmap image from an external source, or a visualization snapshot from the data view can be included in a note. Notes can be organized into a group: a rectangle with a title. The tool supports multiple group levels. A connector line can be drawn between notes, groups, and a note and a group. The connector line can be drawn with or without directed arrows to represent causal relations between findings. These entities enable analysts to record the analysis artifacts such as findings, assumptions, hypotheses and causal relations; organize them into some schema; and build a case to support or contradict an argument using a diagramming technique. Thus, the output of the knowledge view can vary from simple placement of notes next to each other to a highly structured and systematic argumentation based on a diagramming technique. The knowledge view canvas can be panned in all directions if more space is needed. The knowledge view uses a flip-chart metaphor, such that the analyst can create any number of sheets to record the findings. In Figure 3.5, our aim to support the analytical reasoning process based on the processes in traditional reasoning theories is shown using the diagramming techniques available in the knowledge view.

When an entity in the knowledge view is linked to a visualization state, it is marked with a star as shown in Figure 3.5. Analysts can revisit visualization states by clicking on the starred entities in the knowledge view. The knowledge view supports an undo and redo mechanism for creating entities, rearranging and linking with a visualization state. The linking is synchronized with the history tracking mechanism, described in the previous section.

3.5 Prototype

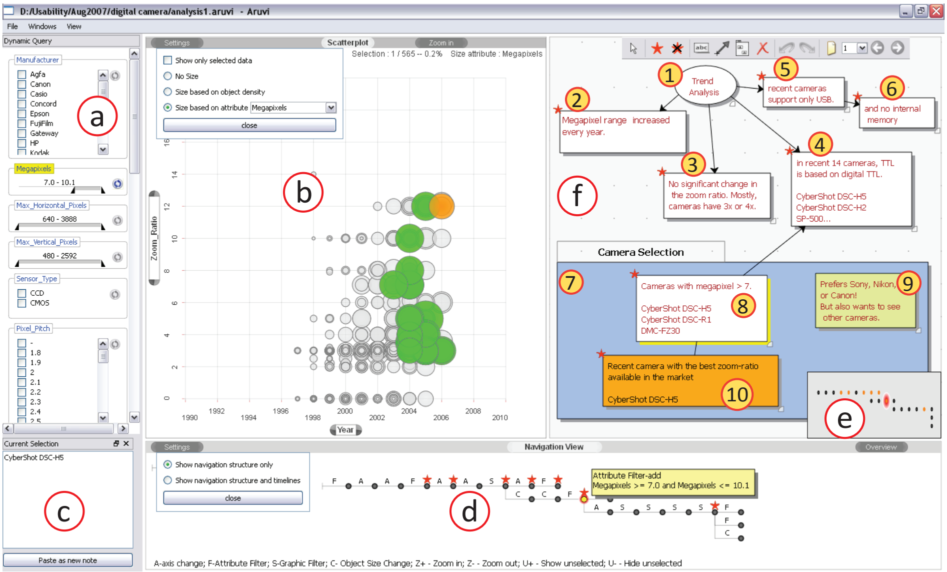
For understanding the support offered by the visual analytics framework for analytic reasoning process, we implemented a prototype of the framework: ***Aruvi***[1]. We developed a data view consisting of a dynamic query interface, a scatterplot and a current selection list, see Figure 3.6(a), (b) and (c) respectively. Scatterplots are extensively used in multivariate data analysis to identify correlations between attributes. A classic example of a scatterplot combined with a dynamic query interface is the Dynamic Home Finder application [136]. This approach is also found in many modern tools such as Spotfire$^{\text{TM}}$and GapMinder$^{\text{TM}}$, to name a few.

The scatterplot in the data view can plot ordinal and nominal attributes on the *x* and *y*-axes. In case of nominal attributes, the unique values of the attribute are sorted alphabeti-

[1] Aruvi means *waterfall* in Tamil, an Indian language. The flow of an exploratory data analysis is similar to the flow of water. An analysis path is opportunisitic and non-linear similar to how a natural watercourse flows towards a lake, a sea or an ocean from an elevation such as mountains. Also, there is a lot of variation within the analysis path based on an analyst's background knowledge, and data and tools availability similar to the different types of waterfall. There are around 10 types of waterfall [76]: *Block* (water descends from a relatively wide stream or river); *Cascade* (water descends a series of rock steps); *Cataract* (a large, powerful waterfall); *Fan* (water spreads horizontally as it descends while remaining in contact with bedrock); *Horsetail* (descending water maintains some contact with bedrock); *Plunge* (water descends vertically, losing contact with the bedrock surface); *Punchbowl* (water descends in a constricted form and then spreads out in a wider pool); *Segmented* (distinctly separate flows of water form as it descends); *Tiered* (water drops in a series of distinct steps or falls); and *Multi-step* (a series of waterfalls one after another of roughly the same size each with its own sunken plunge pool).

The flow of water is also similar to the flow of mind. Edward de Bono, a famous physician and an author, is the originator of the term 'lateral thinking.' He proposed a *water logic* against the traditional thinking. He contends that traditional logic is static, based on the solid foundations of 'is' and identity. In contrast to this traditional 'rock logic', the 'water logic' is based on 'to' and the flow of the mind: "What does this lead to?" as opposed to "What is...?" According to him, this new logic results in a visual 'flowscape', which allows you to lay out and then look at your thinking.

Inspired by the water flow and its analogy to our thinking, we named our prototype Aruvi. Moreover, *'vi'* in Aruvi is a prelude to *vi*sual, *vi*sualization or *vi*sual analytics.



D:/Usability/Aug2007/digital camera/analysis1.aruvi - Aruvi

Selection : 1 / 565 -- 0.2% Size attribute : Megapixels

□ Show only selected data
○ No Size
○ Size based on object density
● Size based on attribute Megapixels

Megapixels 7.0 - 10.1
Max_Horizontal_Pixels 640 - 3888
Max_Vertical_Pixels 480 - 2592

Current Selection
CyberShot DSC-H5

● Show navigation structure only
○ Show navigation structure and timelines

Attribute Filter-add
Megapixels >= 7.0 and Megapixels <= 10.1

A-axis change; F-Attribute Filter; S-Graphic Filter; C- Object Size Change; Z+ - Zoom in; Z- - Zoom out; U+ - Show unselected; U- - Hide unselected

1 Trend Analysis
2 Megapixel range increased every year.
3 No significant change in the zoom ratio. Mostly, cameras have 3x or 4x.
4 in recent 14 cameras, TTL is based on digital TTL.
CyberShot DSC-H5
CyberShot DSC-H2
SP-500...
5 recent cameras support only USB.
6 and no internal memory

Camera Selection
7
8 Cameras with megapixel > 7.
CyberShot DSC-H5
CyberShot DSC-R1
DMC-FZ30
9 Prefers Sony, Nikon, or Canon! But also wants to see other cameras.
10 Recent camera with the best zoom-ratio available in the market
CyberShot DSC-H5

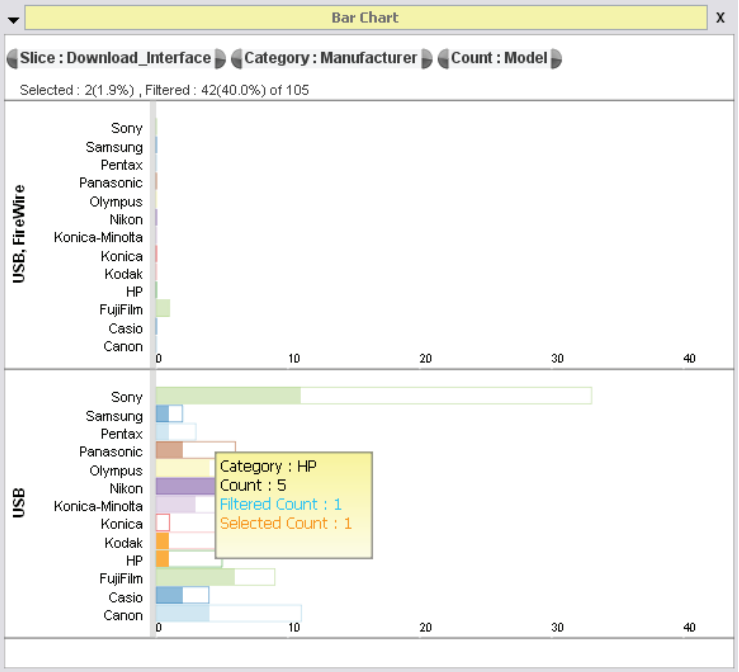
Figure 3.6: The Aruvi prototype. The data view consists of (a) a Dynamic query interface, (b) a scatterplot and (c) a current selection list. (d, e) The navigation view. (f) The knowledge view.

cally and mapped onto an axis. When the attribute mapping of an axis of the scatterplot is changed via a drop-down menu, the transition to a new mapping is animated. When one of the axes is kept constant and the other axis is changed continuously, it aids to recognize the change in the correlation between the new attribute and the previous attribute.

Three different mappings of the data are available in the scatterplot based on size encoding. First, objects can be plotted on the scatterplot without size encoding. This view helps to understand the correlation between the two attributes. Second, the objects can be grouped according to the x- and y- axes values and the density of the objects at each data point on the scatterplot can be plotted using size. Third, the object size can be set based on an attribute value. This mapping enables comparison of three attributes at the same time. These different mappings can be chosen via the *size* interface.

In a later version of Aruvi, a barchart visualization was added. The data shown in the barchart can be sliced based on the values for a nominal attribute or intervals of an ordinal attribute. The bar represents the count of objects within a category or a measure of a category. The barchart is shown in Figure 3.7.

The visualizations in Aruvi are attached to a dynamic query interface. The dynamic query interface automatically generates query widgets for the data attributes according to the data type. For text and boolean data types, check box lists with unique values are created. For numeric data types, sliders are created to specify range selection. Any change in the attribute filters is reflected on the scatterplot dynamically. An attribute filter is reset using the reset button.



Selected : 2(1.9%) , Filtered : 42(40.0%) of 105

Category : HP
Count : 5
Filtered Count : 1
Selected Count : 1

Figure 3.7: A barchart visualization in Aruvi.

The visualizations in Aruvi implement a Degree of Interest (DOI) model based on attribute filtering through a dynamic query interface, and selection through a direct manipulation technique. The objects on the visualizations are selected or unselected by picking, and rectangles or lasso drawn on top of the visualizations. There are three levels of DOI: *low* (objects that do not satisfy the attribute filters), *medium* (objects that satisfy the attribute filters), and *high* (objects that satisfy the attributes filters and are selected through direct manipulation). The color encodings for the three levels of DOI are gray, green and orange respectively. Only the objects with *medium* DOI can be selected through direct manipulation. The DOI of the objects does not change when the data mapping is changed by changing the axes or size. Hence, it is possible to track or emphasize the interesting objects during the entire exploration process. Analysts can also choose to show or hide the *low* DOI objects. The three levels of DOI facilitate convergent analysis. However, analysts can revert back to a previous DOI of the objects using the history tracking mechanism and continue the analysis with different DOIs for the objects. Hence, divergent analysis is also supported.

The *current selection list* interface displays the list of objects with *high* DOI. When there is no selection, it displays the list of objects with *medium* DOI. The object list in the *current selection* interface can be added as a note in the knowledge view using the *paste as new note* interface. The scatterplot allows zoom-in to a particular region of the scatterplot via the *Zoom in* interface. The *settings* interface toggles the display of the *size* and *show* *only filtered data* interfaces. An *information bar* interface is used to display details about the selection and size encoding. Finally, analysts can save, reopen or recover the last analysis using a file menu.

3.5.1 Implementation Notes

Aruvi is implemented in C++ using Qt[2], a cross-platform application and UI framework. The Aruvi user interface has three views: the data view, the navigation view and the knowledge view. These three views are loosely coupled using a backend system.

The backend system has three components: a data manager, a visualization factory and a mediator. Qt's signal and slot mechanism is used to coordinate interaction among these components. Figure 3.8 shows the implementation architecture of Aruvi. It represents the user interface and backend system components, and the key signals that loosely bind the three views.

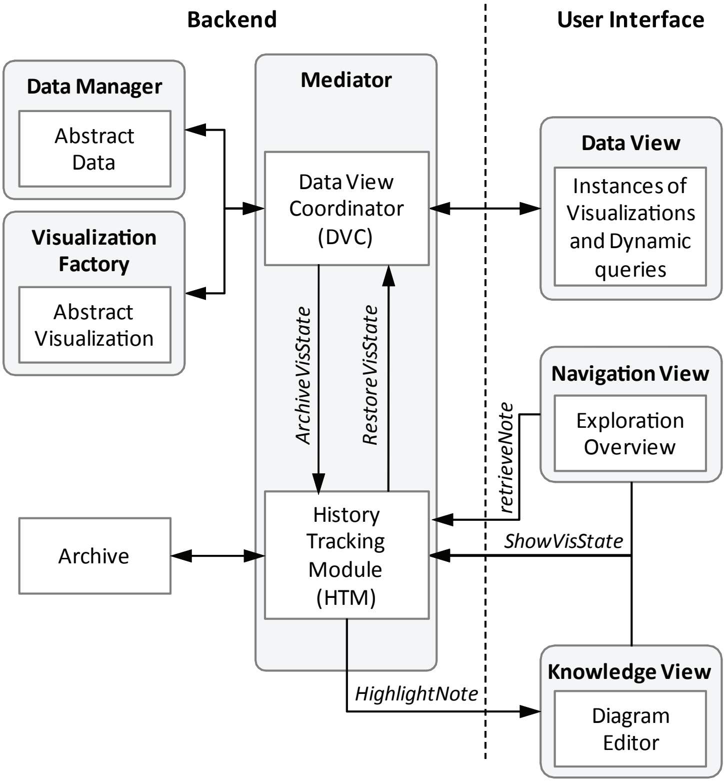


Figure 3.8: Aruvi implementation architecture. It represents the user interface and backend system components, and the key signals that loosely binds the three views.

[2] http://qt.nokia.com/products

The data manager loads data, processes queries and maintains objects' Degree of Interest (DOI). We have designed an abstract data interface to support different data models such as relational data, XML data, and so on. Currently, we have implemented support for relational data, and the SQL query model is used to process queries. The primary key of a dataset is used as object identifier. A DOI map is used to maintain objects' DOI. The data manager updates the hash table after processing queries, and emits a *dataStateChanged(String changeType)* signal. The data change type includes data filtering via dynamic queries and direct manipulation of objects in visualizations.

The visualization factory adopts the factory method design pattern to create visualizations using a type name. An abstract visualization interface defines a basic set of methods such as *create(String TypeName)*, *setProperties(Map Properties)*, and *getProperties()* methods; and a *visualizationStateChanged(String changeType)* signal. A new visualization class defined in Aruvi inherits the *AbstractVisualization* interface, and has a unique type name. Also, it has to emit the visualizationStateChanged signal, when a user changes its specification. The visualization change type includes visual mapping (such as axis-mapping), view setting (such as zooming and panning), and formatting (such as resizing and rearranging visualization windows).

The mediator contains two components: a data view coordinator (DVC) and a history tracking module (HTM). The DVC coordinates the interaction between the data manager and the data view. The data view can request the data manager to process queries and retrieve objects information via the DVC. The DVC also contains the description of the current visualization state that includes properties of visualizations in the data view, data query specifications and the DOI map.

The HTM processes requests for archiving a visualization state, restoring a visualization state, and retrieving notes attached to a visualization state. The DVC sends an *ArchiveVisState* request to the HTM, when the data manager emits the dataStateChanged signal or a visualization emits the visualizationStateChanged signal. Based on the change type, the HTM archives the current visualization state as a new visualization state or merges it with the recent visualization state. The granularity of the history tracking can be chosen in various ways. For instance, all changes to the visualization specification can be captured. However, some heuristics can be applied to avoid too much low level detail. For instance, in Aruvi when a user continuously changes the data filter in the dynamic query interface, the changes are reflected in the visualizations, but are not captured by the history tracking module. We found it to be convenient just to capture the visualization state when the mouse pointer leaves the dynamic query interface and if at least one of the filters has been changed. Other heuristics, like detection of (not necessarily continuous) change patterns could be used and will be studied in the future. The base model itself does allow for a variety of choices here.

The HTM maintains links between notes and visualization states. These links are represented by stars both in the history tree representation and the knowledge view. The HTM processes a user's request to *showVisState* via either the history tree representation or the knowledge view. It retrieves the description of the visualization state from the archive, and requests the DVC to restore the visualization state in the data view. Also, it processes a user's request to retrieve a note via the history representation, and highlights the matching note in the knowledge view.

3.6 Use case

We now present a simple use case where a user explores a digital camera dataset (565 cameras with 15 attributes) using Aruvi. This use case is a constructed example to demonstrate the support offered by Aruvi for the sensemaking process. There are several tasks that the user might perform with the data, such as detecting trends and finding cameras that meet his requirements. In the use case, we specifically emphasize on the sensemaking tasks: capture, review, reuse, share and present (discussed in Section 2.4 and in Figure 2.10). The use case video is available at *http://www.win.tue.nl/%7Eyedendra/imgs/chi1145-shrinivasan.mov*.

To perform trend analysis, the user compares the digital camera attributes for different years. For this comparison, he uses an interactive scatterplot in the data view. He *records* the findings in the knowledge view using a mind map. The mind map is a diagram used to represent ideas linked to and arranged radially around a central idea [27]. He records the central idea — trend analysis — in note 1 (see Figure 3.6(1)). Firstly, he plots the number of megapixels over the years. He records the finding in note 2 and links the note to the current visualization state in the navigation view. Subsequently, he compares the zoom-ratio, eyepiece and download interface attributes against year by changing the scatterplot y-axis. Notes 3, 4 and 5 are his findings; each of these notes is linked to the corresponding visualization state in the navigation view. He then checks whether the selected cameras have internal memory and records the findings in note 6. He completes the mind map by connecting notes 2, 3 and 4 with note 1, and note 6 with note 5 using the connector line with arrow.

Based on the trend analysis, the user defines requirements for selecting a camera. He revisits the visualization states by clicking on the notes recorded in the knowledge view to gain an overview of his analysis. He records those requirements in note 7. In this case, he is looking for a recent camera from manufacturers such as Canon, Nikon, and Sony with 7 megapixels and with a digital TTL (through-the-lens) eyepiece. He *revisits* the visualization state where the cameras with digital TTL were selected by clicking on note 4. Then he changes the size encoding to the megapixels attribute. This creates a new branch in the navigation view (see Figure 3.6(d)).

Using the dynamic query interface, the user selects those manufacturers recorded in note 7 and sets the megapixels attribute range to above 7 megapixels. Three cameras match the requirements. He records this finding in note 9. Then, he plots the zoom-ratio attribute against year, and picks the most recent camera with high zoom-ratio satisfying his requirements. The scatterplot in Figure 3.6(b) highlights this camera. He records this state with note 10. The user *connects* notes 4 and 9, and 9 and 10 to indicate the selection process. He then *groups* notes 7, 9 and 10 used for the camera selection. Finally, he *archives* the analysis along with his findings and decision. This archived analysis can be shared and presented to others using Aruvi.

In this use case, the user combined both the information foraging loop and sensemaking loop opportunistically to reason about the camera dataset and to select cameras. The sensemaking tasks highlighted by this use case are to capture key aspects of the analysis (by recording notes and visualization states, and connecting and grouping notes); and to review the analysis (by revisiting visualization states via notes, and revising visualization states). The share and present tasks were not much emphasized here. Currently, in Aruvi, we support only asynchronous sharing of analysis and findings. Thus, using the three views in Aruvi, the user can capture, review, reuse and share the exploration process and findings which are requirements for supporting the sensemaking process in visual analytics (as discussed in Section 2.4).

3.7 User Study

We conducted a user study to understand the support offered by Aruvi to the analytical reasoning process. The user experiment focused on the quality of results achieved using Aruvi. Further, the user requirements were captured for adding new features and for enhancing the existing features of Aruvi to improve the analysis process.

We invited analysts from different domains to participate in the user study. Four analysts participated in the final user study. The analysts came up with their own datasets. Analyst 1, a usability researcher, was interested in understanding the qualitative output of a usability analysis. The analyst was using a scatterplot to generate hypotheses on the data and perform an initial assessment to decide on the choice of statistical analysis method to draw clear conclusions. Analyst 2, a software quality consultant, was interested in the correlation between the software metrics of a software project to assess the software maintainability and design test cases. The analyst so far was using a pivot table [72] to arrange and sort the columns for comparison. Analyst 3, a software quality modeling researcher, was interested in identifying the outliers and build a case for software analysis based on the software metrics data of a software project. Analyst 4, an urban planning researcher, was interested in chronological building characterization for a city in India to understand how the buildings were developed and their attributes were shaped.

The study had four steps: a training session, an exploration session, an exit questionnaire and an interview with the analyst. Following the training session, the analysts were asked to perform an analysis of their own dataset using Aruvi without a time limit. Usage characteristics were captured while the analysts performed the analysis. After the exploration session, each analyst was interviewed to reflect on the following:

- Which features of Aruvi made a difference in their analysis process?
- Why were those features important for them?
- Express opinions on the prototype in general, especially, its positive and negative aspects.

Further, if there were any interesting usage patterns observed during the exploration sessions, the analyst was asked to explain the intention of such usage behavior.

3.7.1 Data View Usage

The analysts were satisfied with the fairly straightforward visualization offered. They commented that the interactive scatterplot visualization improved their analysis process. The analysts expressed that the interactive scatterplot was quite handy since they need not look back into the data to modify the data selection and visualize it. Figure 3.9 summarizes the usage pattern of the scatterplot with dynamic query in the data view. The usage pattern varied based on the different analysis processes of each analyst. Analyst 1 used mostly dynamic query and *toggle show selected* interfaces; while other analysts predominantly changed axes. The analysts commented that size encoding is useful. Analyst 4 used size coding based on object density for the entire analysis. Analyst 2 and 3 used selection to track software modules' behavior while comparing various different attributes. Analyst 2 suggested adding trend lines to the scatterplot, and displaying statistical information such as x and y axes averages, and correlation coefficients of the current selection against the entire dataset. They also expressed a need for more visualizations. Analyst 1 asked for a scatterplot matrix for simultaneously plotting different datasets. Analysts 2 and 3 said linking scatterplot visualizations to an UML diagram would help them to have an overview of the architecture of the software. Analyst 4 asked for a map visualization to get a spatial context of a pattern seen in the scatterplot.

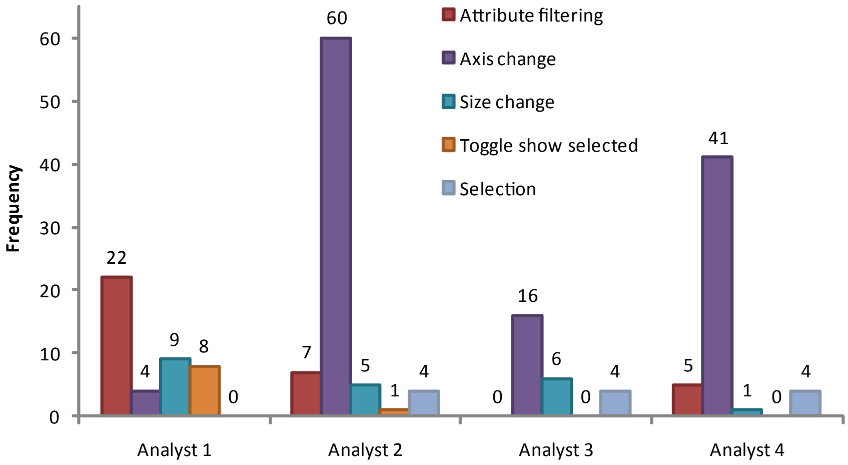


Figure 3.9: The data view usage pattern.

3.7.2 Sensemaking Process Summary

To support sensemaking process during an analysis process, we designed the knowledge view that enables analysts to capture and organize findings along with links to visualization states. Analysts can review analysis via findings in the knowledge view, and via exploration overview in the navigation view. We summarize the knowledge view and the navigation view usage pattern of analysts during their sensemaking process.

Capture and Organize Findings

Analysts recorded findings using notes in the knowledge view. To build a case, analysts grouped notes, and created causal relationship between notes using arrows. Figure 3.10 shows artifacts used by the analysts during their sensemaking process. We found analysts either used sheets or groups to organize findings into topics. Analyst 2 used only sheets for this purpose. Analysts 1 and 4 organized major topics using sheets, and grouped notes into sub-topics within each major topic. Most of the notes recorded were findings, and few were assumptions, hypotheses and reminders.

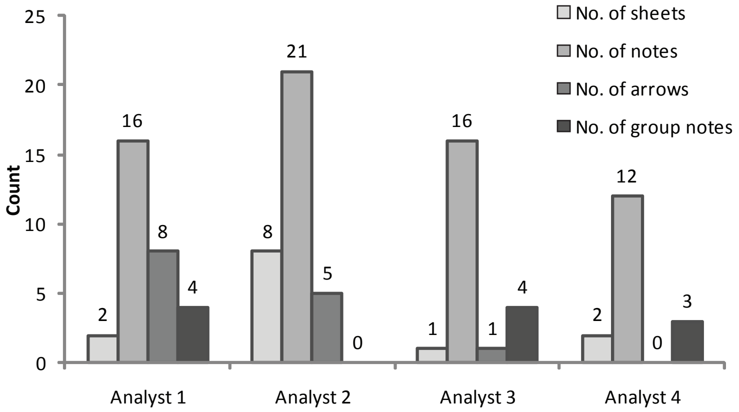


Figure 3.10: The knowledge view usage pattern.

After recording notes, analysts spatially rearranged notes and moved notes to different groups. Analyst 1 had a high artifacts rearrange rate. She rearranged the analysis artifacts quite often in the middle of the analysis. Later, during the interview, she explained that rearranging notes helped her to restructure the analysis process and look for clear conclusions. We discuss her analysis process in Section 3.8.2. Analyst 3 recorded findings at random locations in the sheets during his analysis. After completing his analysis, he rearranged and moved notes to different groups. His analysis process is discussed in Section 3.8.1. Figure 3.11 summarizes the artifacts rearrange rate for the analysts.

Analysts found linking visualization states and artifacts in the knowledge view helpful. Figure 3.12 summarizes the linking pattern of analysts. More than 60% of the artifacts in the knowledge view were linked to visualization states. By connecting notes using arrows, analysts created a semantically richer analysis structure than the automatic analysis structure captured by the history tree representation. On average, 35% of the visualization states were externalized. Analysts found externalizing visualization states with notes in the knowledge view intuitive. It helped them to capture important aspects of the exploration process, and to easily keep track of them during their entire analysis process.

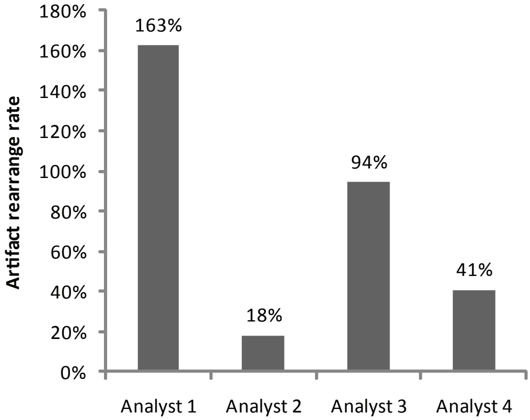


Figure 3.11: Artifacts rearrange rate expressed as the total number of artifacts rearranged as a percentage of the number of artifacts.

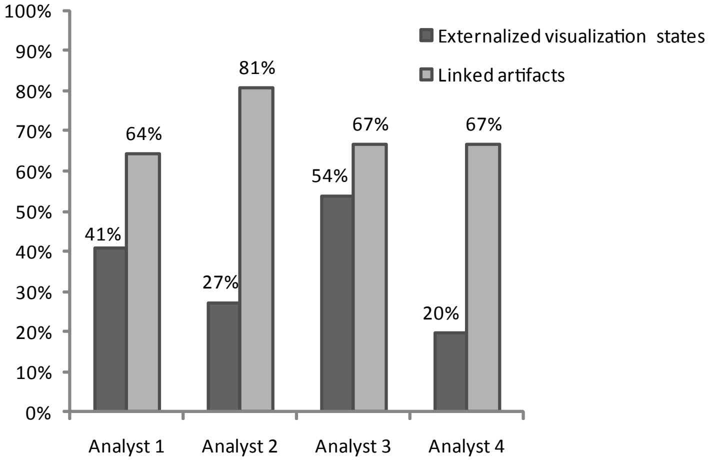


Figure 3.12: Knowledge externalization expressed by percentage of visualization states externalized and artifacts linked with visualization states.

Review and Revise Analysis

Analysts reviewed past visualization states during their sensemaking process. They revisited visualization states either using linked artifacts in the knowledge view or the history tree representation. Mostly, analysts reviewed visualization states via notes. They used the navigation view either to refer back to recent steps and to compare results, or to undo actions. Figure 3.13 presents an overview of visualization state revisits by analysts.

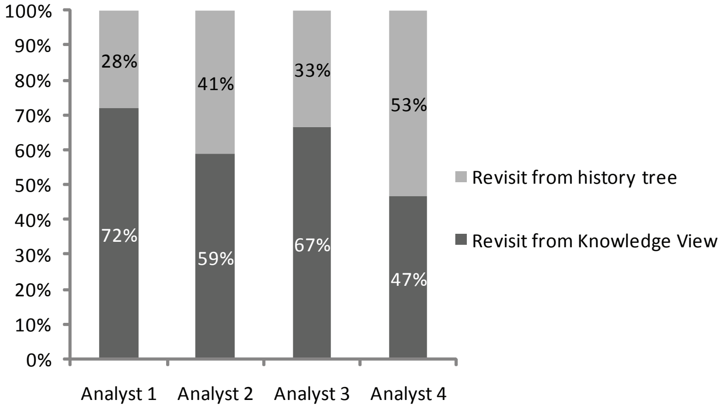


Figure 3.13: Visualization states revisit pattern.

After revisiting visualization states, most often analysts reused visualization states. They mainly reused visualization states to look for alternative solutions. Figure 3.14 summarizes the branching in the analysis process created by reuse of visualization states. Analyst 1 had two main hypotheses that she tried to verify using different analysis paths. Analyst 4 investigated characterized buildings using three analysis paths, which is clearly reflected by his branching pattern. He affirmed this reasoning behind the branching pattern during the interview. Analysts 2 and 3 revisited and reused last visualization states to undo actions during their analysis. Also, Analysts 1 and 4 edited links between visualization states and notes when they found a better visualization state supporting their findings. Figure 3.15 presents an overview of link edit percentages for analysts.

3.7.3 Questionnaire Results

After the analysis session, the analysts were asked to fill out an exit questionnaire about their experiences. The questionnaire is based on the Unified Theory of Acceptance and Use of Technology (UTAUT) model. The model provides guidance to assess the likelihood of success for new technology introductions and helps them understand the drivers of acceptance in order to proactively design interventions targeted at populations of users that may be less inclined to adopt and use new systems [129]. The model provides four core determinants of intentions to use information, such as performance expectancy, effort expectancy, social influence and facilitating conditions, and up to four moderators, such as gender, age, voluntariness and experience for each determinant.

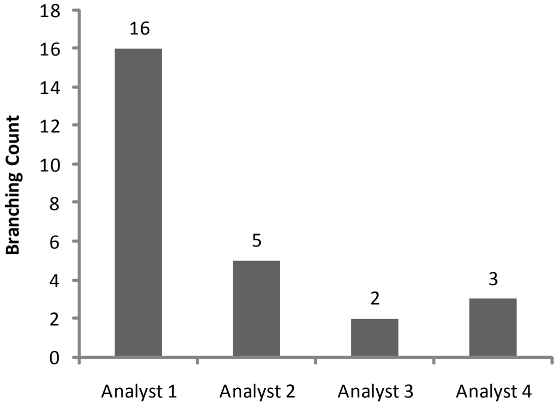


Figure 3.14: Branching pattern. A branch in the exploration process is created when an analyst revisits and reuses a past visualization state.

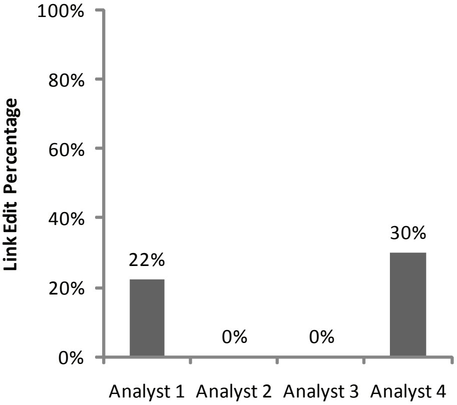


Figure 3.15: Link edit percentage. Analysts edited a link between an artifact and a visualization state when they found a better visualization state supporting their findings.

Performance expectancy focuses on the usefulness of the system; effort expectancy focuses on the degree of ease of use; social influence focuses on the degree to which an individual perceives that important others believe he or she should use the new system; and facilitating conditions focuses on the degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system.

For the preliminary assessment of Aruvi, we choose two determinants of intentions to use — performance expectancy and effort expectancy to reflect on the quality of the results and ease of use. Of the four moderators, gender, age and experience influence the performance expectancy and the effort expectancy.

Items for the performance expectancy include perceived usefulness of Aruvi to synthesize findings, improve the performance of the analysis, improve the productivity of the analyst and improve the effectiveness of the analyst. Items for the effort efficiency include ease of use for exploring the data, recording the findings, synthesizing the findings to build a case and disseminating the findings. The analyst is asked to rate these items on a 5-point Likert scale.

3.7.4 Analysts' Feedback

The questionnaire and the analysts' feedback are presented in Table 3.1. Since the number of the test subjects is small, statistical analysis is not possible; and the moderators such as gender, age and experience do not affect the summation of item responses to create a score for it. Figure 3.16 summarizes the results of the exit questionnaire on performance expectancy and ease of use. Overall, the analysts agreed that Aruvi improved their quality of results by effectively supporting information synthesis (capture and review findings) process.

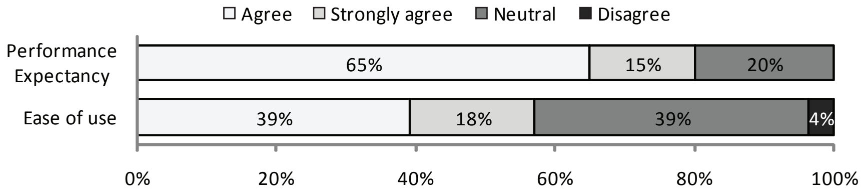


Figure 3.16: A summary of exit questionnaire results on performance expectancy and ease of use. The complete results are shown in Table 3.1.

The knowledge view was one of the key features that supported the sensemaking process by building a bridge between visualization and knowledge gained. They found recording the findings, linking them to the visualizations and organizing them very important for their analysis process, and the use of Aruvi improved the quality of their results. The knowledge view helped to visualize a variety of aspects, for instance, the analyst's hypotheses and assertions, and restructure the analysis to build a case. They also recorded their free thoughts apart from the analysis artifacts linked with visualizations.

Table 3.1: Questionaire Results

Questions	Analyst 1	Analyst 2	Analyst 3	Analyst 4
Performance expectancy				
I find Aruvi to be useful in the information synthesis during the exploration data	agree	agree	agree	agree
Using Aruvi would improve my performance in the analysis of the data?	neutral	agree	strongly agree	strongly agree
Using Aruvi increases my productivity in the analysis of the data?	agree	agree	neutral	strongly agree
Using Aruvi enhances my effectiveness in the synthesis of the information and helps to prove or contradict the claim.	neutral	agree	agree	agree
Using Aruvi would improve the dissemination of my findings and eventually reduce effort in collaboration	agree	neutral	agree	agree
Ease of use				
I know clearly how to use Aruvi for the exploration of data.	agree	agree	agree	strongly agree
I know clearly how to use Aruvi for the synthesis of information.	agree	neutral	neutral	neutral
I know clearly how to use Aruvi for the dissemination of findings.	neutral	neutral	agree	neutral
Interacting with Aruvi does not require a lot of mental effort.	neutral	agree	disagree	strongly agree
I find Aruvi to be easy to use for recording the findings.	strongly agree	agree	neutral	strongly agree
I find Aruvi to be easy to use for synthesizing the findings to build a case.	strongly agree	neutral	agree	agree
I find Aruvi to be easy to use in dissemination of findings.	agree	neutral	agree	neutral

Analyst 1 said "the knowledge view is simple and easy to use for grouping hypotheses, and for quickly constructing and visualizing the structure of all hypotheses." Also, she said that rearranging artifacts in knowledge view during the analysis helped her to restructure the analysis process to look for clear conclusions. Analyst 2 said "Aruvi is really cool to explore the software metrics rapidly and it will help me create an optimal method for analyzing the data in the future."

The analysts found the data view (interactive scatterplot and dynamic query interface) and the knowledge view easy to use. However, interacting with Aruvi without any previous training required some mental effort. They experienced difficulty in comparing past visualization states during a review process. The history tree representation also became incomprehensible for a lengthy analysis process, and did not help in gaining overview of the analysis process.

For analyst 2, the sequence in the navigation view was important, since it represents a workflow. He wanted to rearrange and purge certain visualization states in the navigation view to create an optimum analysis workflow template. This is particularly important for the analyst since this analysis has to be repeated for different datasets quite often. Since the history tracking module captures the dataset and visualization specification of the visualization states separately, workflow template extraction is possible. This is a promising use case for reusing visualization exploration processes.

Analyst 3 expressed difficulties in finding the relevant notes in the knowledge view and suggested a text based search to locate the notes within the knowledge view. He felt that the branching in the history tree showed the reuse pattern, but it did not clearly bring out his implicit thought process. Also, he wanted to group visualization states between axis changes in the scatterplot to get an overview of the analysis.

For analyst 1, the revisit from the knowledge view was easier and more meaningful than from the navigation view. However, the analyst used the navigation view to back track recent visualization states. The analyst also recorded notes on the revisited nodes. It supports the fact that knowledge creation is an unsystematic process; and the analyst wants to back track to see what has happened in the recent history to affirm a thought. Since the history tracking mechanism captures the visualization states automatically, the analyst can get access to the exhaustive list of visualization states via the navigation view. On the other hand, the knowledge view enables the analyst to record visualization states selectively. Hence, the analysts can opportunistically use the navigation view and knowledge view to revisit the visualization states for reviewing and validating their findings, and reusing the visualization to look for alternate views. Analysts 1, 2 and 4 expressed a need for export of the output of the knowledge view and visualizations as a report or presentation file. The analysts appreciated the possibility to save and restore their analysis.

3.8 Case Studies

We next present two case studies based on the analysis process of two analysts (1 and 3) who participated in the user study discussed in the previous section.

3.8.1 Software quality analysis

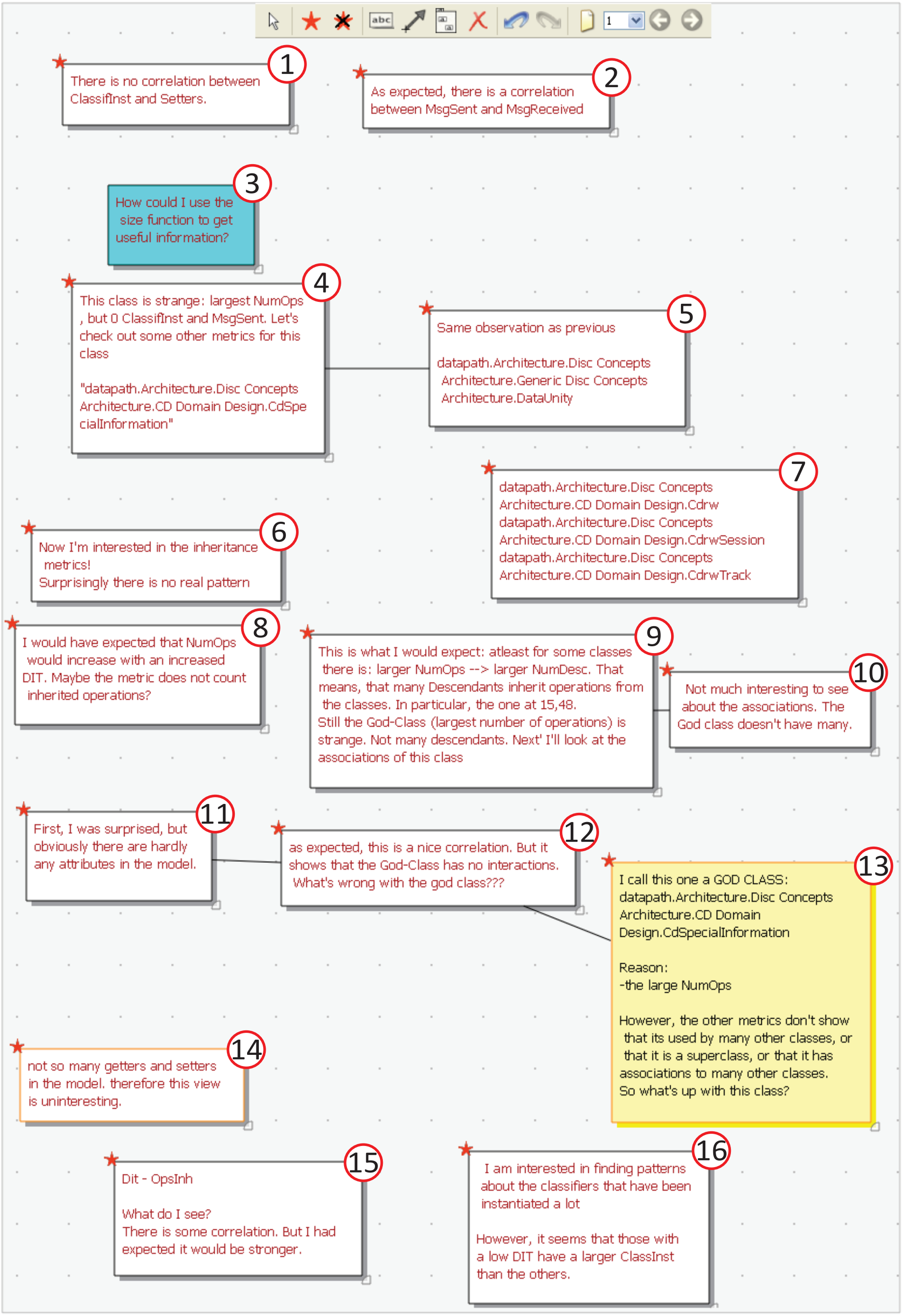
In this case study, we describe an analysis process where a software quality analyst explored software metrics data of a software project. The analyst explored the software metrics data to identify a so called *god class* (a single class that does everything and lacks any abstraction) and to understand the complexity in the interaction among classes in the project. A scatterplot attached to a dynamic query interface is used to understand the relationship between software metrics of the project.

The analyst first compared the *number of classifier instantiation* attribute against the *number of setter methods* and *number of messageSent* attributes. The analyst recorded his findings in notes 1 and 2 as shown in figure 3.17. The analyst recorded a reminder on how to use size encoding in the scatterplot using note 3. Subsequently, the analyst chose size encoding based on the *numOps* (number of operations) attribute. The analyst found immediately an outlier — a class with a large number of operations that was neither instantiated nor sent any messages (highlighted in Figure 3.19). The analyst suspects this class is a god class. The analyst recorded this finding using note 4, and wanted to further validate this assert.

As the analyst continued the exploration, he also recorded some more classes as god class suspects. These classes were recorded using notes 5 and 7. For validating the claim, the analyst compared the *depth of inheritance tree* (*DIT*) against the *class to leaf depth* (the longest path from the class to a leaf node in the inheritance hierarchy below the class) and *numOps* attributes. The findings were recorded using notes 6 and 8. The analyst was surprised to see a lack of positive correlation between the *DIT* and the *numOps*.

Next, the analyst investigated the class association. The analyst found that the suspected god class did not have many associations. The analyst elaborates this finding using notes 9 and 10. Then, the analyst studied the interaction pattern of the classes. First, the analyst compared the *number of variables* against the *numOps* (note 11). Next, the *number of messageSent* attribute was compared with the *number of messageReceived* attribute. Since the suspected god class did not have any interactions, it was confirmed to be a god class. The analyst elaborated his reasoning for identifying the god class using note 13, and wanted to investigate the class using an UML diagram, which is not supported in the prototype. Towards the end of the analysis, the analyst recalled the lack of correlation between the *DIT* and the *numOps* was because the inherited operations were not considered while counting the *numOps*. The analyst then compared the *DIT* against the *inherited operations* attribute (note 15).

After completing the analysis, the analyst organized his findings into four groups. The analyst grouped uninteresting views into the initial exploration group (see figure 3.18(a)); findings concerning the interaction pattern study into the interaction group (see figure 3.17(b)); findings related to the *DIT* attribute into the inheritance group (see figure 3.18(c)); and findings related to class association into the structure group (see figure 3.18(d)). The analyst expressed the organization of the findings into the interaction group clearly summarized the identification of the god class.



1

There is no correlation between ClassifInst and Setters.

2

As expected, there is a correlation between MsgSent and MsgReceived

3

How could I use the size function to get useful information?

4

This class is strange: largest NumOps , but 0 ClassifInst and MsgSent. Let's check out some other metrics for this class

"datapath.Architecture.Disc Concepts Architecture.CD Domain Design.CdSpecialInformation"

5

Same observation as previous

datapath.Architecture.Disc Concepts
Architecture.Generic Disc Concepts
Architecture.DataUnity

6

Now I'm interested in the inheritance metrics!
Surprisingly there is no real pattern

7

datapath.Architecture.Disc Concepts
Architecture.CD Domain Design.Cdrw
datapath.Architecture.Disc Concepts
Architecture.CD Domain Design.CdrwSession
datapath.Architecture.Disc Concepts
Architecture.CD Domain Design.CdrwTrack

8

I would have expected that NumOps would increase with an increased DIT. Maybe the metric does not count inherited operations?

9

This is what I would expect: atleast for some classes there is: larger NumOps --> larger NumDesc. That means, that many Descendants inherit operations from the classes. In particular, the one at 15,48.
Still the God-Class (largest number of operations) is strange. Not many descendants. Next' I'll look at the associations of this class

10

Not much interesting to see about the associations. The God class doesn't have many.

11

First, I was surprised, but obviously there are hardly any attributes in the model.

12

as expected, this is a nice correlation. But it shows that the God-Class has no interactions. What's wrong with the god class???

13

I call this one a GOD CLASS:
datapath.Architecture.Disc Concepts
Architecture.CD Domain
Design.CdSpecialInformation

Reason:
-the large NumOps

However, the other metrics don't show that its used by many other classes, or that it is a superclass, or that it has associations to many other classes.
So what's up with this class?

14

not so many getters and setters in the model. therefore this view is uninteresting.

15

Dit - OpsInh

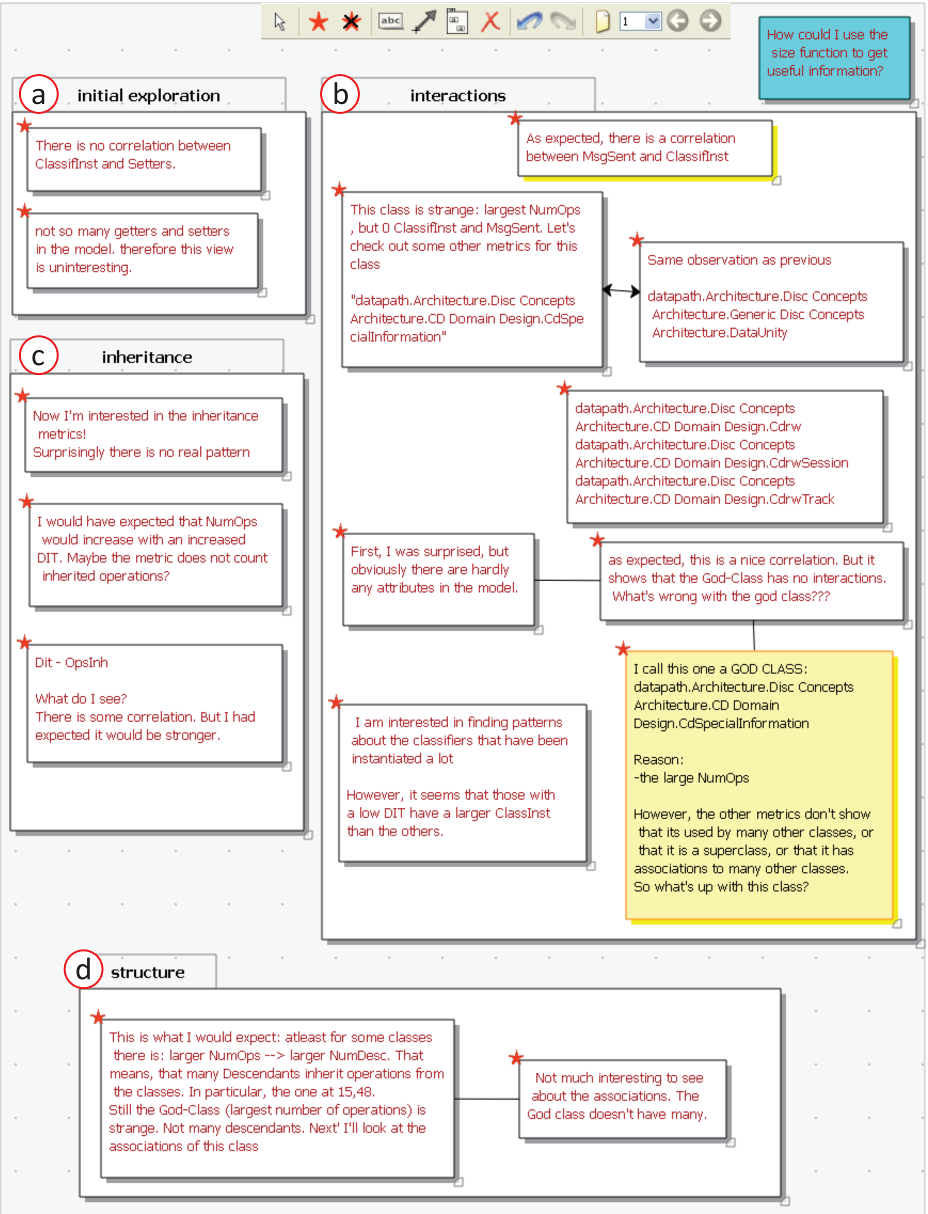
What do I see?
There is some correlation. But I had expected it would be stronger.

16

I am interested in finding patterns about the classifiers that have been instantiated a lot

However, it seems that those with a low DIT have a larger ClassInst than the others.

Figure 3.17: The knowledge view of the software quality analyst during the exploration process.



How could I use the size function to get useful information?

initial exploration

There is no correlation between ClassifInst and Setters.

not so many getters and setters in the model. therefore this view is uninteresting.

interactions

As expected, there is a correlation between MsgSent and ClassifInst

This class is strange: largest NumOps , but 0 ClassifInst and MsgSent. Let's check out some other metrics for this class

"datapath.Architecture.Disc Concepts Architecture.CD Domain Design.CdSpecialInformation"

Same observation as previous

datapath.Architecture.Disc Concepts Architecture.Generic Disc Concepts Architecture.DataUnity

datapath.Architecture.Disc Concepts Architecture.CD Domain Design.Cdrw
datapath.Architecture.Disc Concepts Architecture.CD Domain Design.CdrwSession
datapath.Architecture.Disc Concepts Architecture.CD Domain Design.CdrwTrack

First, I was surprised, but obviously there are hardly any attributes in the model.

as expected, this is a nice correlation. But it shows that the God-Class has no interactions. What's wrong with the god class???

I call this one a GOD CLASS:
datapath.Architecture.Disc Concepts Architecture.CD Domain Design.CdSpecialInformation

Reason:
-the large NumOps

However, the other metrics don't show that its used by many other classes, or that it is a superclass, or that it has associations to many other classes.
So what's up with this class?

I am interested in finding patterns about the classifiers that have been instantiated a lot

However, it seems that those with a low DIT have a larger ClassInst than the others.

inheritance

Now I'm interested in the inheritance metrics!
Surprisingly there is no real pattern

I would have expected that NumOps would increase with an increased DIT. Maybe the metric does not count inherited operations?

Dit - OpsInh

What do I see?
There is some correlation. But I had expected it would be stronger.

structure

This is what I would expect: atleast for some classes there is: larger NumOps --> larger NumDesc. That means, that many Descendants inherit operations from the classes. In particular, the one at 15,48.
Still the God-Class (largest number of operations) is strange. Not many descendants. Next' I'll look at the associations of this class

Not much interesting to see about the associations. The God class doesn't have many.

Figure 3.18: Findings organized by the software quality analyst after the exploration process.

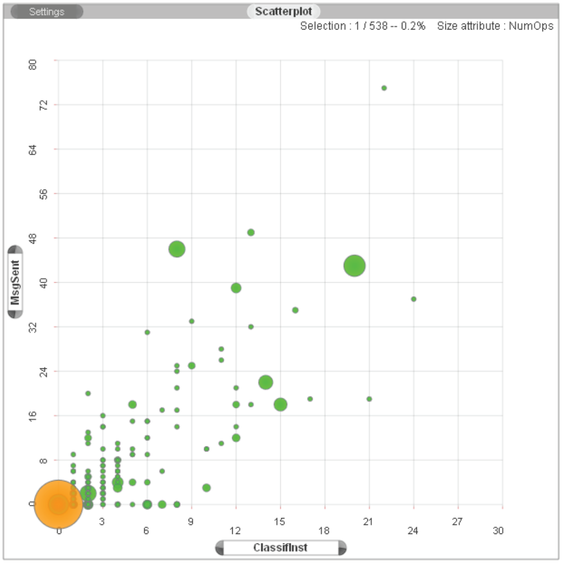
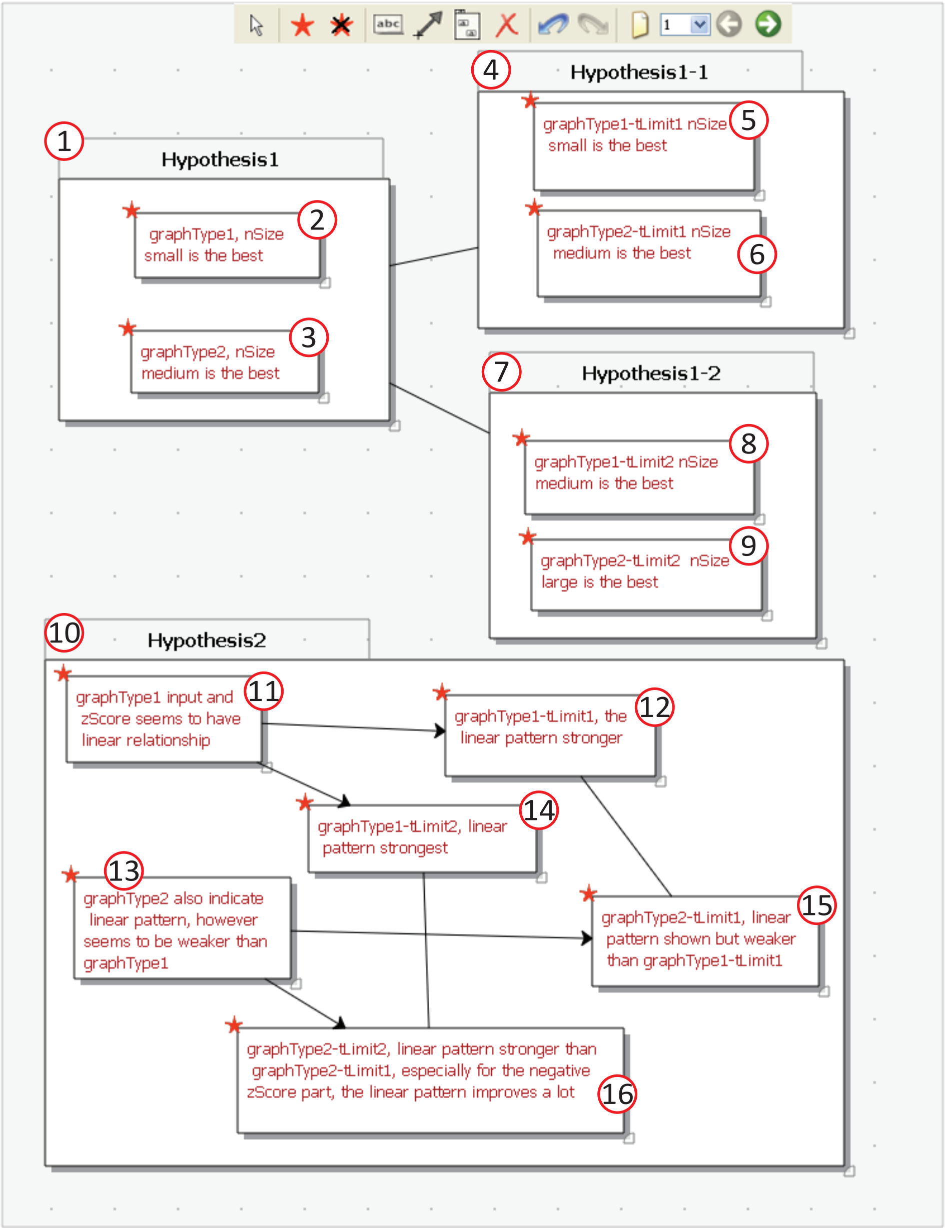


Figure 3.19: The scatterplot showing an outlier class (highlighted in orange) which the software quality analyst suspected to be a god class.

3.8.2 User experiment data analysis

A usability analyst was interested in understanding the results of perceptual user experiments. The analyst conducted a user experiment to evaluate how people perceive correlation from graphical representations of data. Two such representations, scatterplot and parallel coordinate plots, were used, and furthermore the number of samples (*nSize*), the correlation coefficient (*zScore*) and the time limit (*tlimit*) were varied. Users were requested to give their judgment of the correlation on a five point scale (*user input*). The analyst wanted to generate hypotheses by understanding the relationship between the control variables and the user observed correlation during the experiment. Based on this initial assessment the analyst wanted to choose appropriate statistical analysis methods that will help her to draw clear conclusions.

First, the analyst focused on the effect of the sample size on the correlation judgment. The findings were recorded using notes 2 and 3 (figure 3.20(2) and figure 3.20(3)) and were grouped into Hypothesis 1 (figure 3.20(1)). Further the analyst refined the hypotheses for each graph type considering the time limit control variable. The refined hypotheses were recorded using notes 5, 6, 8, and 9 and were summarized using group boxes 4 and 7. The analyst bookmarked the key visualization states using notes 2, 3, 11, and 14. These notes were often used to revisit the key visualization states and revise them to refine the hypotheses. This reuse pattern is clearly seen in the branching structure of the analyst's exploration presented in the navigation view (figure 3.21). Finally, the analyst linked notes 12 and 15, and 13 and 16 to represent a weak relation between the hypotheses generated on the two graphical representations.



graphType1, nSize small is the best

graphType2, nSize medium is the best

graphType1-tLimit1 nSize small is the best

graphType2-tLimit1 nSize medium is the best

graphType1-tLimit2 nSize medium is the best

graphType2-tLimit2 nSize large is the best

graphType1 input and zScore seems to have linear relationship

graphType1-tLimit1, the linear pattern stronger

graphType1-tLimit2, linear pattern strongest

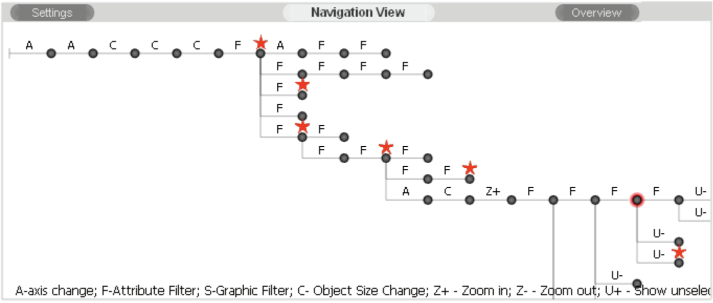
graphType2 also indicate linear pattern, however seems to be weaker than graphType1

graphType2-tLimit1, linear pattern shown but weaker than graphType1-tLimit1

graphType2-tLimit2, linear pattern stronger than graphType2-tLimit1, especially for the negative zScore part, the linear pattern improves a lot

Figure 3.20: The knowledge view of the user experiment data analyst. The numeric labels are used to describe her analysis process.

Next, the analyst investigated the correlation between the shown and perceived correlation. The findings were recorded using notes 11 to 16, and grouped into Hypothesis 2 (see figure 3.20(10)). During the interview, the analyst expressed that "the knowledge view is simple and easy to use for grouping hypotheses, and for quickly constructing and visualizing the structure of all hypotheses". The analyst rearranged the notes quite often in the middle of the analysis. She explained that rearranging helped to restructure the analysis process.



A-axis change; F-Attribute Filter; S-Graphic Filter; C- Object Size Change; Z+ - Zoom in; Z- - Zoom out; U+ - Show unselec

Figure 3.21: The navigation view shows an overview of the exploration process done by the user experiment data analyst.

3.9 Discussion

Currently, visual analytics designers focus on creating interactive visual interfaces to explore data using design models explained in Section 2.1.2 and Section 2.2.2. For interactive visualization, Shneiderman [114] provides a list of seven high level tasks — overview, zoom and filter, details-on-demand, relate, history, and extract — to support interactive data exploration. Extending these tasks to visual analytics, Keim et al. [78] emphasize on analyze first; show important; zoom, filter and analyze further; and details on demand. Interactions in visual analytics systems are designed to accomplish these basic tasks. For solving well-defined problems, users can compose these tasks in a structured way. However, for solving complex problems involving large data, they compose these basic tasks opportunistically, and often the structure of the analysis process tends to be complex.

Klein's recognition-primed decision making model [80] states that "the analysis process in a complex problem solving rarely arises straightforwardly, but rather results from a long and recursive process with back tracking and erratic switching among the following activities: thinking about ideas, production, reorganization, modification, and evaluation." However, to support these activities the user must be aware of 'what has been done and found'. Mica [52] claims that maintaining an awareness about the decision process is critical because "there is considerable evidence that a person's manner of characterizing a situation will determine the decision process chosen to solve a problem." It is necessary for the analyst to identify important elements of the situation and relationships between these elements for maintaining the awareness about the decision process [14].

We found an interesting usage pattern emerging from the problem solving process of the analysts participating in our user study. The analysts explicitly used the knowledge view and the navigation view to support their awareness of the analysis context. The analysts opportunistically exploited the possibility of linking the findings in the knowledge view to the visualization states in the navigation view. They selectively placed the important visualization states (for the analyst) in the analysis path as notes in the knowledge view. They used the knowledge view as a hypertext editor to refer to key visualizations. These notes were used to revisit and review the visualizations to get an overview of what has been done. The analysts also used those notes to revisit and reuse the visualization for searching alternative solutions or hypotheses. By connecting notes, they could not only structure their analysis output, but also build a semantically richer analysis structure than the automatically captured analysis structure shown in the history tree representation.

In general, we believe that our user studies reveal that incorporation of a knowledge view, consisting of a simple graphical editor with the possibility to link to visualization states, is a useful addition to a standard visual analytics tool. Also, the history tree representation is useful, but it does not seem optimal to provide an overview of the analysis. It only presents the structure of the exploration process, but fails to present important elements of the analysis and relationship among these elements. Hence, we need to identify better visual representations for the navigation view to automatically provide an overview of the analysis process. Figure 3.22 summarizes the topics discussed in this dissertation based on the feedback of the analysts and the reviewers (of the paper [117] based on which this chapter is written).

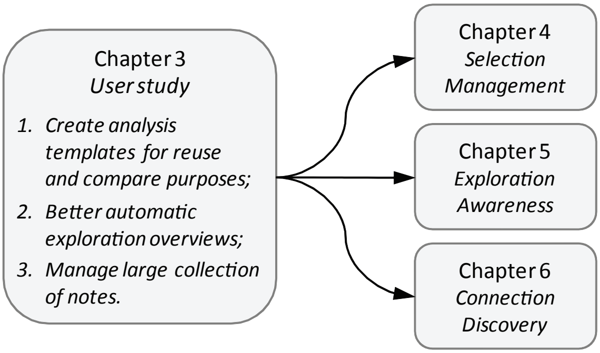

Chapter 3
User study
1. *Create analysis templates for reuse and compare purposes;*
2. *Better automatic exploration overviews;*
3. *Manage large collection of notes.*
Chapter 4
Selection Management
Chapter 5
Exploration Awareness
Chapter 6
Connection Discovery

Figure 3.22: The topics discussed in this dissertation based on analysts' and reviewers' feedback.

3.10 Conclusion

In this chapter, we presented a sensemaking framework for visual analytics that contains three integrated views: a data view, a navigation view, and a knowledge view. The sensemaking is facilitated by extending visualization support to externalize the mental models and link the analysis artifacts to the visualizations. It also enables the analysts to revisit the visualization states to review and validate the findings, and reuse these to look for alternate solutions or hypotheses. Finally, the analyst can organize the externalized analysis artifacts to build a case. Thus, users can capture, review, reuse and share an analysis process using the navigation view; and findings using the knowledge view. Four analysts participated in a user study with their own datasets. The perceived usefulness of Aruvi was discussed based on the usage pattern of the exploration sessions and the interviews conducted with the analysts. Analysts found that recording the findings, and linking them to the visualizations and organizing them were very important for their analysis process. Analysts agreed that the use of Aruvi improved the quality of their results.

During exploratory data analysis, selection techniques such as dynamic queries and brushing help users to progressively converge on interesting data items. Also, they can edit these selections, and thereby perform a divergent analysis. These convergent and divergent analyses using selection techniques are illustrated in the use case discussed in Section 3.6. In the next chapter, we present an approach to capture and reuse these selections more explicitly during exploratory data analysis; and enable analysts to effectively reason based on data selection.

Chapter 4

Select & Slice

> *Divide and conquer* — A political maxim advocated by the ancient Roman and French rulers, and widely used by the British colonization during the 18th and 19th century.

During exploratory data analysis, selection techniques such as dynamic queries and brushing are used to specify and extract items of interest. In other words, users define areas of interest in data space that often have a clear semantic meaning. We call such areas Semantic Zones, and argue that support for their manipulation and reasoning with them is highly useful during exploratory analysis. An important use case is the use of these zones across different subsets of the data, for instance to study the population of semantic zones over time. To support this, we present the Select & Slice Table. Using this table, users can capture, reuse and combine zones; and compare and trace items of interest across different semantic zones and data subsets. We present four case studies to illustrate the support offered by the Select & Slice table during exploratory analysis of multivariate data.

4.1 Selection Management

During interactive data exploration, users select data items to drill down or highlight items in the visualizations. For selecting these data items, they use interaction techniques such as dynamic queries [13] and brushing [20, 32], to specify conditions over functions of data attributes. During exploratory analysis, these selection techniques help users to progressively converge on interesting data items. Also, they can edit a selection specification, and thereby perform a divergent analysis.

Current visual analytics systems offer limited support to explicitly capture and reuse selections during an analysis. Often, brushing leads to selection of items, and when users change the visual mapping they can keep track of these selected items [65, 2, 9, 132]. When they specify a new selection, the previous selection is lost. Hence, data selection is often transient in these visualization systems. It is difficult for users to manually keep track of these selection specifications during a long analysis process. Also, they cannot effectively reuse selection specifications, and compare the results of these specifications. Therefore, for effective reasoning based on data selection, we argue that support for capturing and manipulating selection specifications is highly useful during an exploratory analysis.

Areas of interest in data specified by data selection usually have a clear semantic meaning, unless users select items by accident. We enable users to capture such areas of interest in data as *Semantic Zones* or simply *Zones*. A zone holds either a selection specification, or a set of items extracted using the selection specification. It has a label provided by a user. Figure 4.1 shows four zones: *rich nations*; *developing nations*; *poor nations*; and *India, Brazil, and Kenya*. Also, in current visual analytics systems, users cannot quickly slice and dice the selected items over different subsets of the data to study the distribution of these items. Examples of such tasks are 'how many nations in different continents belong to each zone?' and 'how did the nations move to different zones over time?'

A popular approach to slice and dice a multi-dimensional dataset is a pivot table. The pivot table provides an aggregate summary of a data attribute by cross-tabulating the dimensions of a dataset. A visualization spreadsheet is another approach that helps users to compare visualizations representing different data sets side-by-side [34]. It provides extensive cell manipulation operations similar to a spreadsheet. However, none of these interfaces can be directly used to manage and manipulate zones during an exploratory data analysis, for instance to see the contents of zones for different subsets of the data.

In this chapter, we present a table interface that enables users to capture and manipulate zones during an exploratory analysis (see Figure 4.2). The table interface is used in addition to a data view that contains interactive data analysis tools including visualizations. Firstly, users can externalize zones from the data view and archive these in the header along one axis of the table. The labels of zones are displayed in the header of the axis. Secondly, users can retrieve items from different data subsets. The data subsets are arranged along the other axis of the table; the labels of the data subsets are displayed in the header of the axis. A cell contains a set of items from a data subset that matches the specification of a zone. Thus, items of datasets are sliced based on the specifications of zones in the table. Hence we call this interface the *Select & Slice* table. Items in cells can be visualized in various ways, as a count, as an aggregation of a measure, or as a separate visualization, such that the table gives an overview of the relationship between zones and data subsets.

Next, users can edit specifications of zones using a zone editor attached to the Select & Slice table. During an analysis, they can reuse a zone specification by dragging its label from the table onto the data view. Then, users are enabled to drill down to a particular data set from the Select & Slice table in the data view. Next, they can logically combine the sets of items in the cells, and highlight the resulting items in the data view using simple mouse operations. Also they can study the distribution of items in the table using a set comparison operation and a keyword search. Thus, we adopt Shneiderman's information visualization mantra — overview first, zoom and filter, and details on demand — for manipulating zones during an exploratory analysis. Finally, we present four case studies that were conducted to understand the support offered by the Select & Slice table during exploratory analysis of multivariate data.

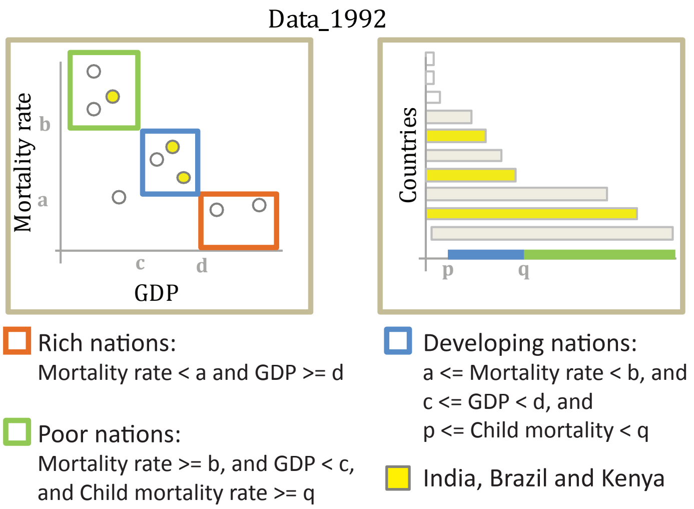


Figure 4.1: Four semantic zones shown in two visualizations. A zone has either a data selection specification or a set of items extracted using a data selection specification. It has a label provided by a user.

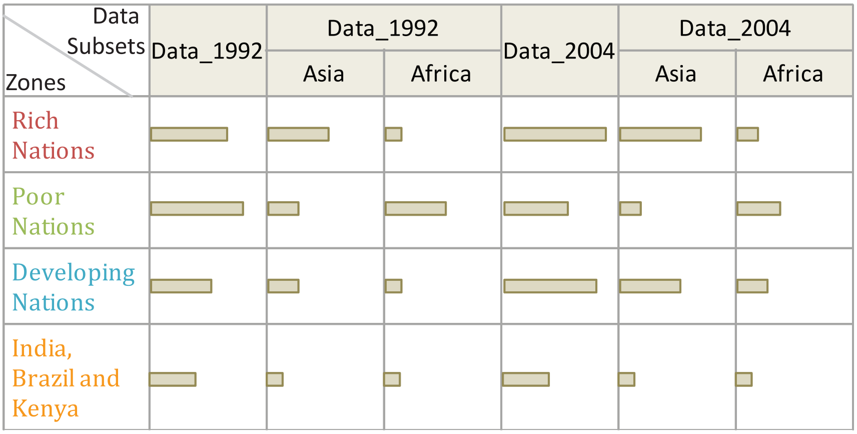


Figure 4.2: A Select & Slice table showing the distribution of items of the four zones across different subsets of two datasets (Data_1992 and Data_2004). The length of a bar in a cell represents the number of nations.

4.2 Related Work

First, we discuss existing techniques to capture and archive selections in visualization systems. Next, we present visualization techniques that are closely related to the Select & Slice table.

4.2.1 Selection Management

Several visualization systems help users to capture areas of interests in data specified through selection techniques. Visualization systems such as Aruvi [117], Cross-filtered views [135], Gapminder [2] and Flare [65] capture brushing as a declarative query, and reuse it when the view is transformed. QlikView [8] tracks the users' selection process and helps them to define alerts based on the data attribute criteria. Doleisch et al. [46] present a framework for capturing features using brushing, and archive these features using a feature definition language. They use a tree view to archive and edit the features. These archived features are used to steer 3D visualization of computational simulation data. Similarly, streamline predicates [108] are used for capturing flow structures while visualizing flow simulation data. In interactive analysis of simulation data [82], function graphs of attributes are used to specify areas of interest. These systems mainly focus on archiving and editing those regions of interests during exploratory analysis. They do not support reuse of selection specifications on subsets of data, and the comparison of the results of these specifications.

In Tableau [9], users can create and analyze subsets of data using computed sets. A computed set is used as a derived dimension in the analysis. However, the computed sets cannot be simultaneously sliced across different subsets of data. Visualization systems such as XMDV [91] and Mondrian [121] support brush editing. Users can change the logical composition of brushes during an exploration process. XMDV can simultaneously display multiple N-dimensional brushes to compare brush results. Using a similar approach, Elmqvist [51] supports multiple brushes in a scatterplot. Chen [32] uses a data-flow model to define multidimensional brushes. The number of brushes that can be simultaneously displayed in visualizations (XMDV and Mondrian), and tracked during animation (Aruvi, Gapminder and Flare) is limited. Moreover, in all these systems, users cannot simultaneously reuse these brushes on different subsets of data, and compare the results of these brushes side by side.

4.2.2 Visualization techniques

Visualization techniques help users to interactively explore multi-dimensional data. Examples of such techniques are interactive axis reconfiguration, tables, re-orderable matrix, multi-dimensional scaling, dimensional stacking and glyphs. The Select & Slice table uses a tabular approach to slice and dice items of datasets using zones. This approach is closely related to spreadsheets and pivot tables.

A spreadsheet displays a grid of cells. A spreadsheet cell contains a value, or a formula that defines the content of the cell by combining values of other cells in the spreadsheet. When the content of a cell is changed, the sheet is automatically re-calculated. In a visualization spreadsheet [34], cells contain visualization operators that transform data into views. When the content of a cell is changed, all views in the spreadsheet are automatically updated. In contrast to a spreadsheet, users cannot directly edit contents of cells in the Select & Slice table. They can only edit the specifications of zones and subsets of datasets to change the contents of cells; and cells provide an overview of the relationship between zones and data sets. As a result, spreadsheets offer much flexibility and focus on management and reuse of data flows; whereas the Select & Slice table aims at offering ease of use for the management and reuse of selection specifications.

A pivot table, found in spreadsheet programs such as Microsoft Excel and OpenOffice.org Calc, helps to slice and dice multi-dimensional data. A pivot table provides an aggregate summary of a data attribute by cross-tabulating the dimensions of a dataset. The pivot table has hierarchical clusters of data attributes along its row and column headers. Polaris [120] adopts a tabular layout similar to a pivot table; its cells have visualizations automatically chosen based on the composition algebra and the graphic design criteria. In contrast to the pivot table and Polaris, the Select & Slice table headers have zones along one axis of the table and subsets of data along the other axis of the table. Also, a cell contains items retrieved from a data subset that match the specification of a zone. It provides visual summaries of the items in various ways, as a count, an aggregation of a measure, or as a separate visualization. The pivot table shows grand summaries of the data field at the end of the rows and columns. Items in the cells of the Select & Slice table are not mutually exclusive, as the zones can define overlapping areas of interest in data. Hence, the table cannot show grand summaries at the end of rows and columns.

In summary, a spreadsheet offers much flexibility, but does not directly support handling of user defined semantic zones and subsets of the data; a pivot table is too rigid in the sense that along both dimensions of the table the data have to be partitioned. We argue that the solution that we provide, that is, a combination of user defined zones and dataset slicing, is often very useful for analysis and visualization purposes. In the following section, we describe the implementation of the Select & Slice table to support capturing and manipulating zones during an exploratory analysis.

4.3 Approach

To support reasoning based on data selection in visual analytics, we enable users to

- construct the Select & Slice table during an exploration process by
  - capturing the selection specifications or selected items from the data view as zones with user-defined labels;
  - retrieving items from different subsets of data using zones; and
  - visualizing the retrieved items in various ways, as a count, as an aggregation over a measure, or as a separate visualization.
- study the distribution of the items in the table;
- support drilling down to a particular subset of data from the table in the data view.

The Select & Slice table is implemented in Aruvi discussed in Section 3.5. The architecture of Aruvi was modified to simultaneously access multiple data sets from different databases during an analysis. Users provide a unique identifier to a dataset while loading it into Aruvi. The Select & Slice Table is implemented as a part of the knowledge view. So far, the knowledge view enabled users to capture interesting aspects of their exploration process by bookmarking visualizations, and recording and reordering findings using diagramming techniques. They can organize findings to build a case. Similarly, the Select & Slice table can also be used to build a case by manipulating zones, and by studying the distribution of items retrieved from datasets for zones.

We make use of a classic cars dataset from the 1983 ASA Statistical computing and graphics data expo (http://stat-computing.org/dataexpo/1983.html) to illustrate the features of the Select & Slice table. The dataset contains 406 cars with 9 attributes such as model name, mpg, number of cylinders and acceleration.

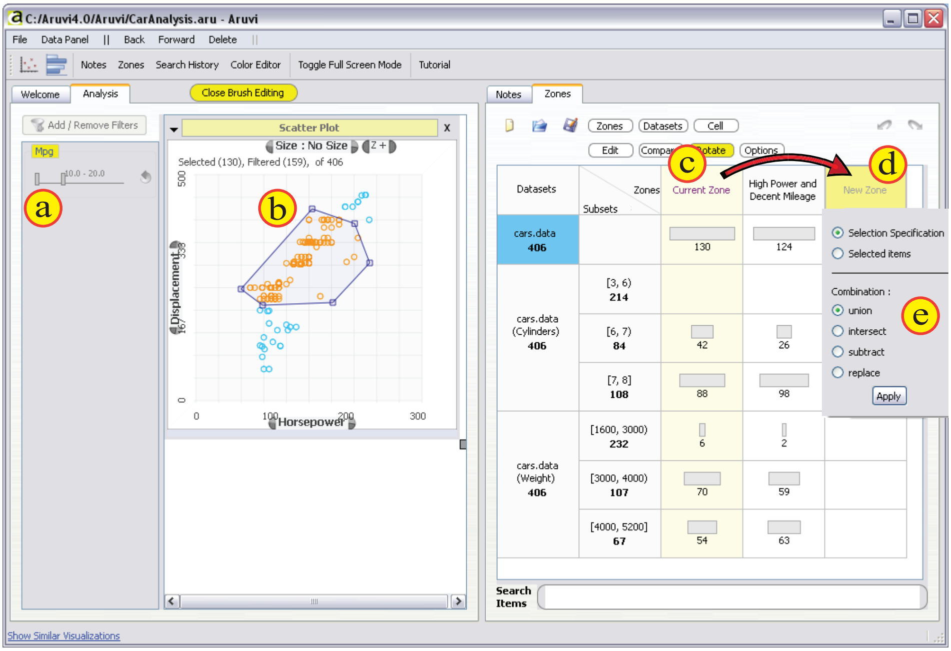
4.3.1 Constructing the Select & Slice Table

Encoding Selection

We encode selections specified by users in the data view using a SQL-like query language as in earlier systems (e.g., [100, 65, 86, 43]), and graphics operations such as object in polygon test. A selection specification consists of conditions over functions of data attributes. In Aruvi, users can specify a selection using dynamic query widgets and brushing. First, items are optionally filtered using dynamic query widgets. These dynamic queries are directly expressed using SQL clauses. Then, a brush can be used to select items in the visualizations. A brush is specified by picking items, by dragging a rectangle, or by drawing a lasso over items in the visualizations. Picking selects an item using its object id (primary key). A rectangle brush is expressed using SQL BETWEEN or IN operators. For a lasso selection, first its bounding box is expressed as a rectangle brush; then, the selected items are identified using an object in polygon test. The type of visualization determines how these SQL and graphics operators are applied on attributes to select items in the visualization. For instance, in a scatterplot, a rectangle brush is expressed as the intersection of range queries on x- and y- axes attributes; in a barchart, it is expressed as the intersection of a range query on the measure axis, and a set of items selected in the category axis. Finally, the current selection in the data view is defined by intersecting dynamic queries (Figure 4.3a) and brushing (Figure 4.3b).

Creating a new Select & Slice Table

When a new Select & Slice table is created, it is populated with a current zone and the current dataset as shown in Figure 4.4a. The current zone holds the current selection specification from the data view throughout an exploration process. The current dataset is highlighted in blue in the Select & Slice table. It also shows the number of items selected from the current dataset based on the current selection in the data view, using a bar representation.



Datasets	Zones / Subsets	Current Zone	High Power and Decent Mileage	New Zone
cars.data **406**		130	124	
cars.data (Cylinders) **406**	[3, 6) **214**			
	[6, 7) **84**	42	26	
	[7, 8] **108**	88	98	
cars.data (Weight) **406**	[1600, 3000) **232**	6	2	
	[3000, 4000) **107**	70	59	
	[4000, 5200] **67**	54	63	

● Selection Specification
○ Selected items

Combination :
● union
○ intersect
○ subtract
○ replace

Apply

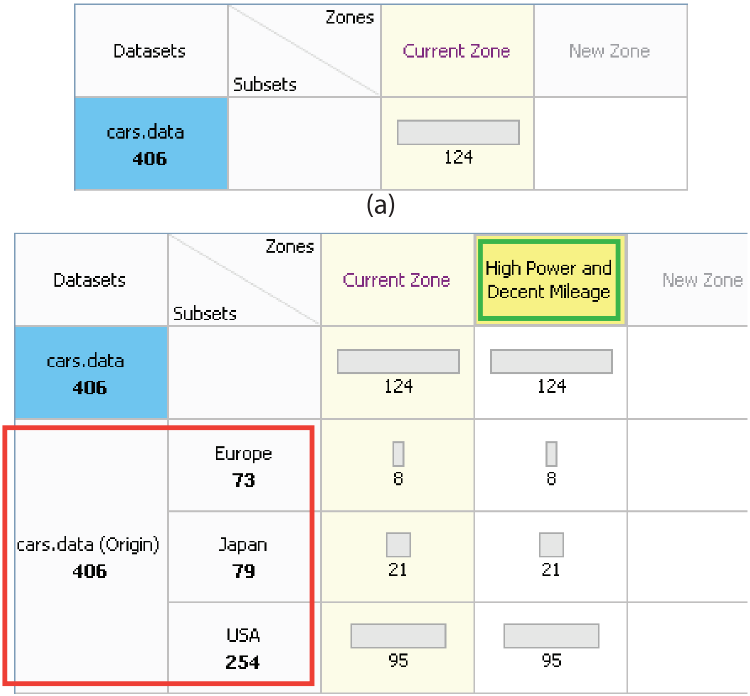
Figure 4.3: The Select & Slice Table is shown as a part of the knowledge view in the Aruvi visualization system. A filter (a) and a brush (b) are combined to define the current zone (c). Users can define a new zone by dragging the current zone, an existing zone or a cell on to the 'new zone' place holder (d). (e) The new zone composition menu.

Defining Zones

A new zone is defined by dragging the current zone header onto the 'New Zone' placeholder (Figure 4.3d). Next, users can choose to store either the selection specification or the selected items (Figure 4.3e); and provide a label for the new zone. Figure 4.4b highlights a newly defined zone in green.

Defining Datasets

Users can obtain more insight in the distribution of items in zones by defining subsets of datasets. For instance, in Figure 4.4b the original dataset is split up according to *Origin* countries. Users can subset a dataset based on one of its attributes. A nominal attribute can be used to subset data using either its unique domain values or groups of these. An ordinal attribute can be used to subset data using a clustering method such as equal intervals, quartile, percentile, standard deviation, unique values and custom intervals. A temporal attribute can be used to subset data by monthly, quarterly, yearly or custom intervals. Figure 4.5 shows the data subset definition interface in Aruvi. Also, they can change the current dataset in the data view by selecting a dataset in the table.



Datasets	Zones Subsets	Current Zone	New Zone
cars.data 406		124	

(a)

Datasets	Zones Subsets	Current Zone	High Power and Decent Mileage	New Zone
cars.data 406		124	124	
cars.data (Origin) 406	Europe 73	8	8	
	Japan 79	21	21	
	USA 254	95	95	

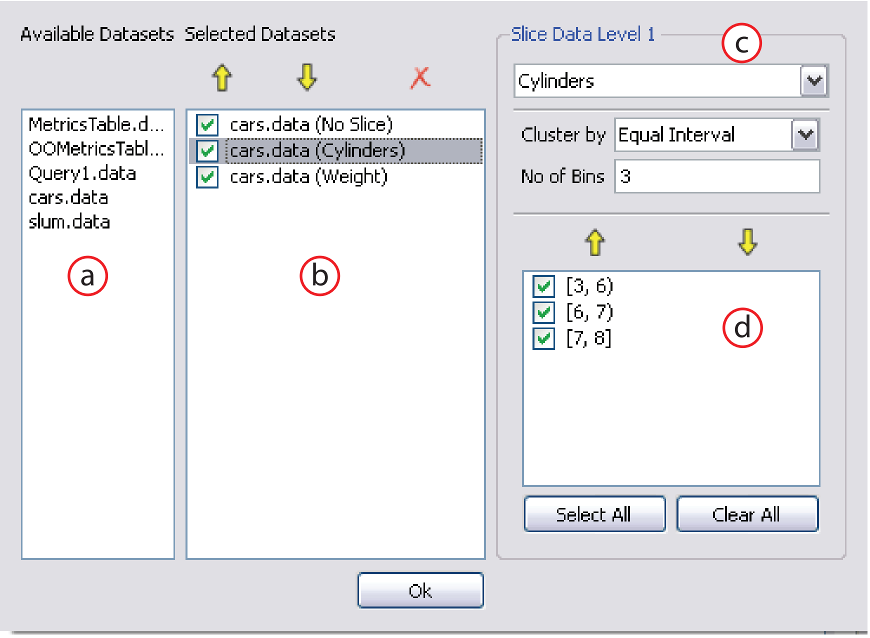
(b)

Figure 4.4: (a) A new Select & Slice Table. (b) The table with a new zone (highlighted in green), and three new subsets of the data based on attribute slicing (highlighted in red).

Cell Contents

Each cell contains a set of items. Users can request to show a summary (the number of elements, the average value of an attribute, etc.) or a visualization of all items. A bar or a bubble is used to visualize a summary of the items in a cell. First, the number of items, or a measure such as a total or an average value of an attribute is used to determine the length of a bar or the radius of a bubble in a cell. The lengths of bars and the radii of bubbles are normalized across cells of the table to simplify comparison across rows and columns. Bars can be aligned either to the center, or to the left of cells; bubbles are placed at the center of cells. A label showing the number of items is placed below a bar or a bubble. Users can choose either a linear or a logarithmic scale for mapping the number of items onto a bar and a bubble.

To show all items, the active visualization from the data view can be shown in each cell and the items in the cell are plotted in that visualization. In this way, a visualization matrix is created to provide an overview of items in the cells of the table. Currently, in Aruvi a scatterplot can be shown in each cell (Figure 4.6).



MetricsTable.d...
OOMetricsTabl...
Query1.data
cars.data
slum.data

☑ cars.data (No Slice)
☑ cars.data (Cylinders)
☑ cars.data (Weight)

Cylinders

Cluster by: Equal Interval
No of Bins: 3

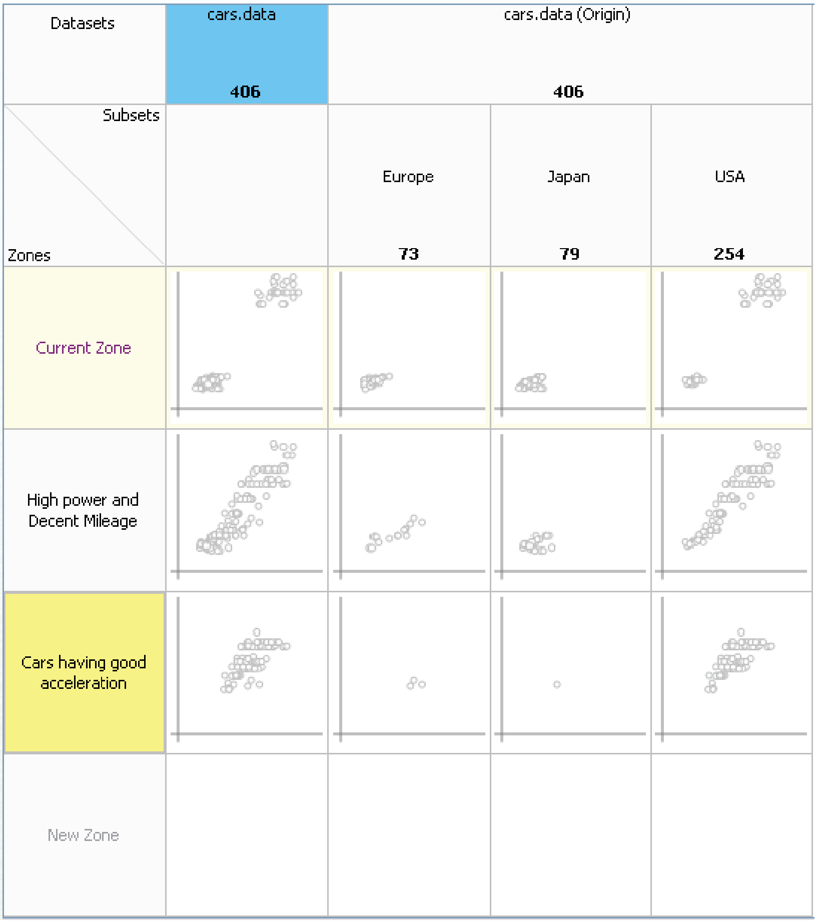
☑ [3, 6)
☑ [6, 7)
☑ [7, 8]

Figure 4.5: Data subset definition interface. (a) A list of available datasets using unique dataset identifiers provided by the user. (b) A list of subsets of datasets. The labels for the subsets are automatically generated by combining the dataset identifier, and the attribute name that is used for subsetting. Users can rearrange, show, hide or remove a selected dataset from the list. (c) A subset definition panel. (d) The subsets of a selected dataset. An ordinal interval is represented using a standard interval notation; where [3, 6) means $3 <= x < 6$.

Manipulating Zones

Dataset header elements hold selection specifications for data subsets. Cells contain the selection specifications of both zones and data subsets to retrieve items. Hence, each element of the headers and also each cell can be considered as a zone. So, we enable users to define new zones also by dragging a header element or a cell onto the 'New Zone' placeholder. Existing zones can be combined using one of the operations union, intersect, subtract and replace (Figure 4.3e). They can also reuse a zone definition (filters and brushes) in the data view by dragging a zone onto the current zone in the table.

Most operations on zones can be done using simple manipulations. For cases where detailed inspection and editing is needed, users can also manipulate zones using a zone editor. The zone editor has two components: a combination editor (Figure 4.7a) and a list of selection specifications (Figure 4.7b). The combination editor allows users to logically compose selection specifications using a parse tree representation. A parse tree completion assistant helps users to construct a valid combination of selection specifications. Below the combination editor, a list of selection specifications is shown. Users can



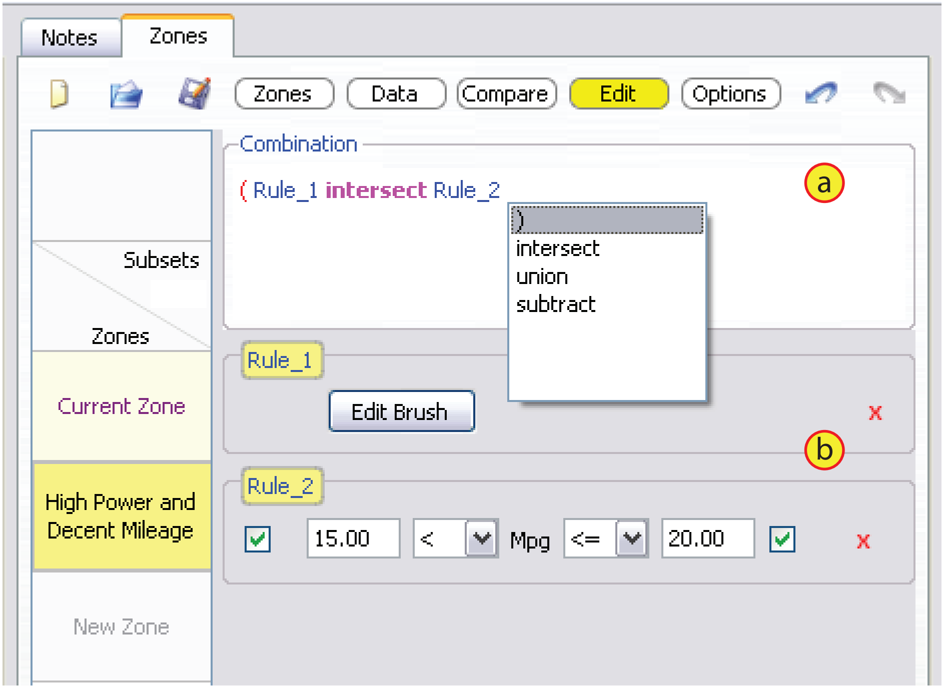
Datasets	cars.data 406	cars.data (Origin) 406		
Subsets Zones		Europe 73	Japan 79	USA 254
Current Zone				
High power and Decent Mileage				
Cars having good acceleration				
New Zone				

Figure 4.6: Scatterplot Matrix in the Select & Slice Table.

directly edit the selection specifications created using dynamic query widgets in the data view. For those selection specifications created using brushing, they can directly edit the corresponding brushes by restoring the original visualization state via the ‘Edit Brush’ button. An undo and redo mechanism is provided to the users for zone manipulation.

4.3.2 Studying Items Distribution

A set comparison operation and a keyword search are provided to study the distribution of items in the Select & Slice table. The zones and data subsets can be rearranged and rotated to support side-by-side comparison.



Combination

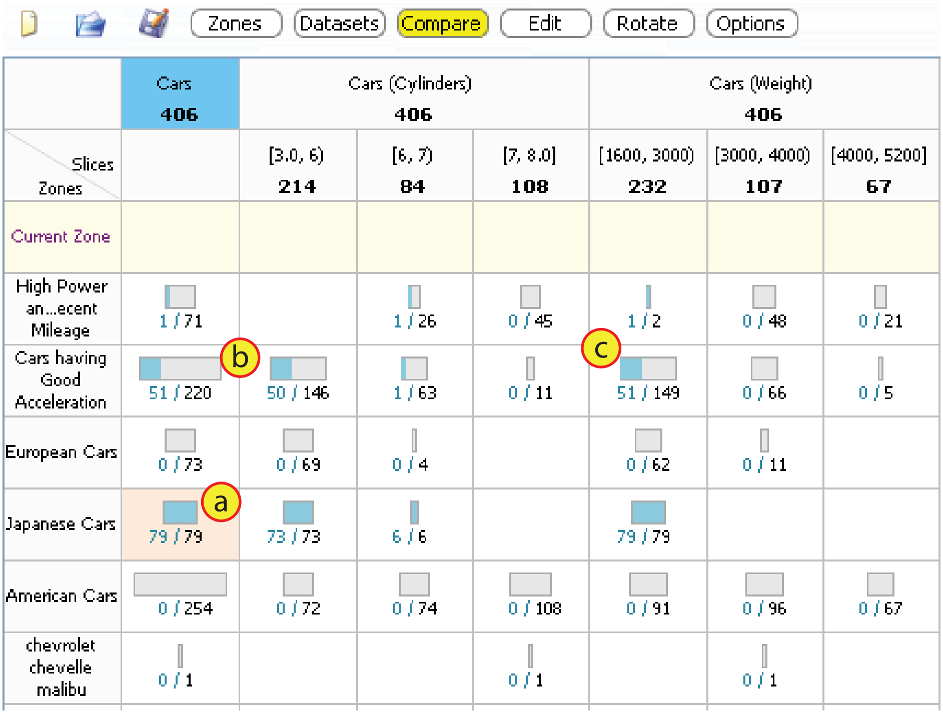
(Rule_1 intersect Rule_2

Rule_2: ☑ 15.00 < Mpg <= 20.00 ☑

Figure 4.7: Semantic zone editor. (a) Zone composition editor with a parse tree completion assistant. (b) A list of selection specifications (filters and brushes) that defines a semantic zone.

Set Comparison

A user can compare items of a cell with items of the other cells in the table. When the user double-clicks a cell, the Select & Slice table enters comparison mode. The selected cell used for comparison is filled with light red (Figure 4.8a). To identify the number of similar items in a cell with respect to the selected cell, the items of the cell are intersected with the items of the selected cell. The number of similar items in each cell with respect to the selected cell is shown as a ratio in blue below the bars or bubbles. Also the similarity ratio is visualized through a blue filling in the bars or bubbles. Using this comparison view, users can trace items across different zones and data subsets. For example, Figure 4.8 shows the distribution of Japanese cars (Figure 4.8a) across different zones and different subsets of the car dataset. One-fourth of the 'cars having good acceleration' (51 out of 220 cars) are Japanese in the dataset; and these Japanese cars (50 out of 51 cars) have between 3 and 6 cylinders (see Figure 4.8b). All these 51 Japanese cars weigh less than 3000 pounds (Figure 4.8c). This items distribution study shows Japanese car industry did not focus on producing powerful and heavy cars, but manufactured lightweight cars with good acceleration. For other aggregations apart from count, a blue filling and a gray filling in a cell represent a summary for the similar items and all items respectively.



	Cars 406	Cars (Cylinders) 406			Cars (Weight) 406		
Slices Zones		[3.0, 6) 214	[6, 7) 84	[7, 8.0] 108	[1600, 3000) 232	[3000, 4000) 107	[4000, 5200] 67
Current Zone							
High Power an...ecent Mileage	1 / 71		1 / 26	0 / 45	1 / 2	0 / 48	0 / 21
Cars having Good Acceleration	51 / 220	50 / 146	1 / 63	0 / 11	51 / 149	0 / 66	0 / 5
European Cars	0 / 73	0 / 69	0 / 4		0 / 62	0 / 11	
Japanese Cars	79 / 79	73 / 73	6 / 6		79 / 79		
American Cars	0 / 254	0 / 72	0 / 74	0 / 108	0 / 91	0 / 96	0 / 67
chevrolet chevelle malibu	0 / 1			0 / 1		0 / 1	

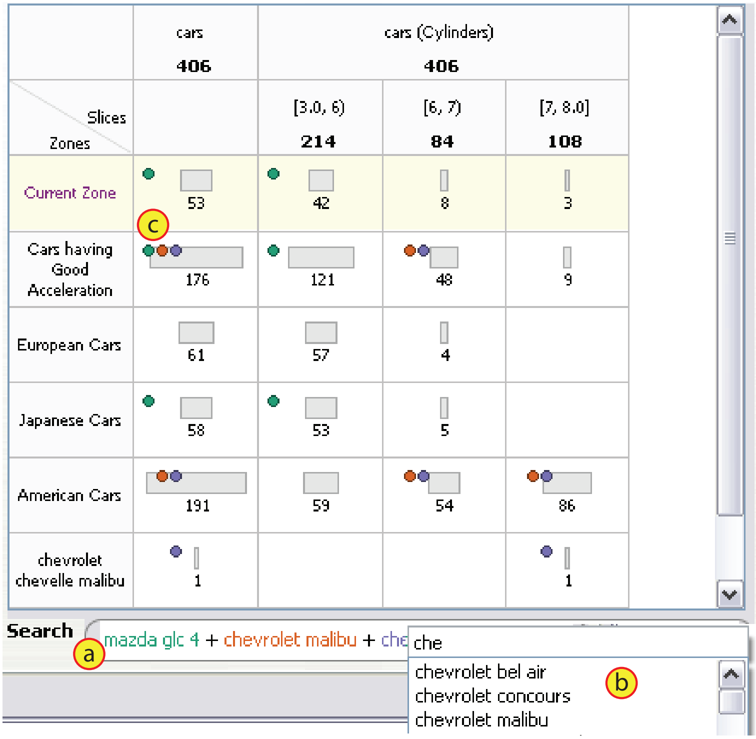
Figure 4.8: Set comparison view shows the distribution of Japanese cars (a). (b) One-fourth of the 'cars having good acceleration' are Japanese in the dataset; these Japanese cars (50 out of 51 cars) have between 3 and 6 cylinders. (c) All these 51 Japanese cars weigh less than 3000 pounds.

Keyword Search

Users can search for individual items in the table using a keyword search interface (Figure 4.9a), with an item suggestion list (Figure 4.9b). The keywords are separated by a '+' character and assigned a color. The search results are visualized using colored dots in cells. A dot is colored based on the corresponding keyword color in the search interface. Figure 4.9 shows a user searching for three cars: 'mazda glc 4', 'Chevrolet malibu' and 'Chevrolet chevelle malibu'. The search results are shown using colored dots in cells (Figure 4.9c). Currently, Aruvi does a wild card matching for a keyword. One dot is shown in a cell for a keyword even though many items in the cell can match the keyword. Using this keyword search, the user could infer that the three cars have good acceleration; and also identify their country of origin and cylinder specification.

4.3.3 Drill Down Analysis

During an exploratory analysis, users can compose a complex brush by selecting items in the table, and drill down to investigate these items in the data view. The brush is defined



Slices / Zones	cars 406	cars (Cylinders) 406		
		[3.0, 6) 214	[6, 7) 84	[7, 8.0] 108
Current Zone	● 53	● 42	8	3
Cars having Good Acceleration	●●● 176	● 121	●● 48	9
European Cars	61	57	4	
Japanese Cars	● 58	● 53	5	
American Cars	●● 191	59	●● 54	●● 86
chevrolet chevelle malibu	● 1			● 1

Search mazda glc 4 + chevrolet malibu + che

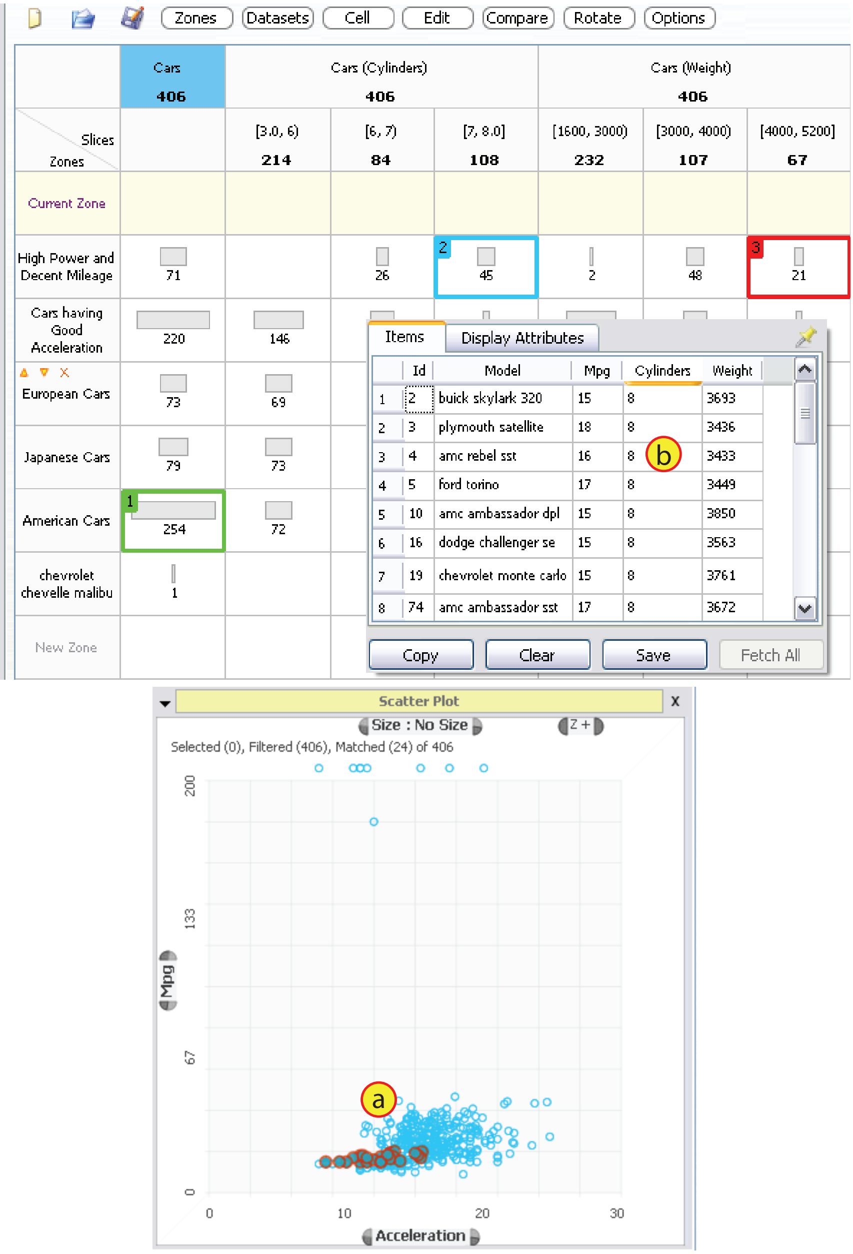
che
chevrolet bel air
chevrolet concours
chevrolet malibu

Figure 4.9: (a) Keyword search interface. (b) An item suggestion list. (c) Search results are visualized using colored dots in cells. A dot is colored based on its corresponding keyword's color in the search interface.

by logically combining the selected cells. Cells can be added, intersected or subtracted using click, shift+click, and ctrl+click; and these cells are marked green, red and blue respectively. When a user selects a cell, the selection status of that cell is toggled and the selection status of other cells is kept constant, similar to multi-selection mode in a list box widget. The order of the selection sequence is shown in the highlighted cells. The selection is cleared by pressing the escape key. In Figure 4.10, the selected cells in the Select & Slice table show 'American cars with 8 cylinders that are not heavy' (green $\cap$ blue $\setminus$ red). These cars are highlighted in the scatterplot (Figure 4.10a). Detailed information about items selected by the brush is shown in the table's context menu (Figure 4.10b). They can also archive these items along with detailed information as a comma-separated file to study them in other software systems or for reporting purposes.

4.4 Case Studies

We present analysis processes of four data analysts to illustrate the support offered by the Select & Slice table for an exploratory analysis. The analysts are experts from different



Slices / Zones	Cars 406	Cars (Cylinders) 406			Cars (Weight) 406		
		[3.0, 6) 214	[6, 7) 84	[7, 8.0] 108	[1600, 3000) 232	[3000, 4000) 107	[4000, 5200] 67
Current Zone							
High Power and Decent Mileage	71		26	2 45	2	48	3 21
Cars having Good Acceleration	220	146					
European Cars	73	69					
Japanese Cars	79	73					
American Cars	1 254	72					
chevrolet chevelle malibu	1						
New Zone							

Items | Display Attributes

	Id	Model	Mpg	Cylinders	Weight
1	2	buick skylark 320	15	8	3693
2	3	plymouth satellite	18	8	3436
3	4	amc rebel sst	16	8	3433
4	5	ford torino	17	8	3449
5	10	amc ambassador dpl	15	8	3850
6	16	dodge challenger se	15	8	3563
7	19	chevrolet monte carlo	15	8	3761
8	74	amc ambassador sst	17	8	3672

Selected (0), Filtered (406), Matched (24) of 406

Figure 4.10: Support for drill-down analysis. The selected cells 1, 2 and 3 highlighted in green, blue and red respectively compose a brush – ‘American cars with 8 cylinders that are not heavy’. These cars are highlighted in the scatterplot (a). (b) Detailed information about those cars.

domains such as software quality analysis, embedded systems and urban planning. They often use visualization tools for their day-to-day data analysis tasks.

First, analysts carried out their domain specific data analysis tasks using the Aruvi visualization system. Following that we conducted an informal interview to understand the usefulness of the Select & Slice table. We present our observations of their analysis processes, and discuss their feedback on the Select & Slice table.

4.4.1 Software Quality Analysis

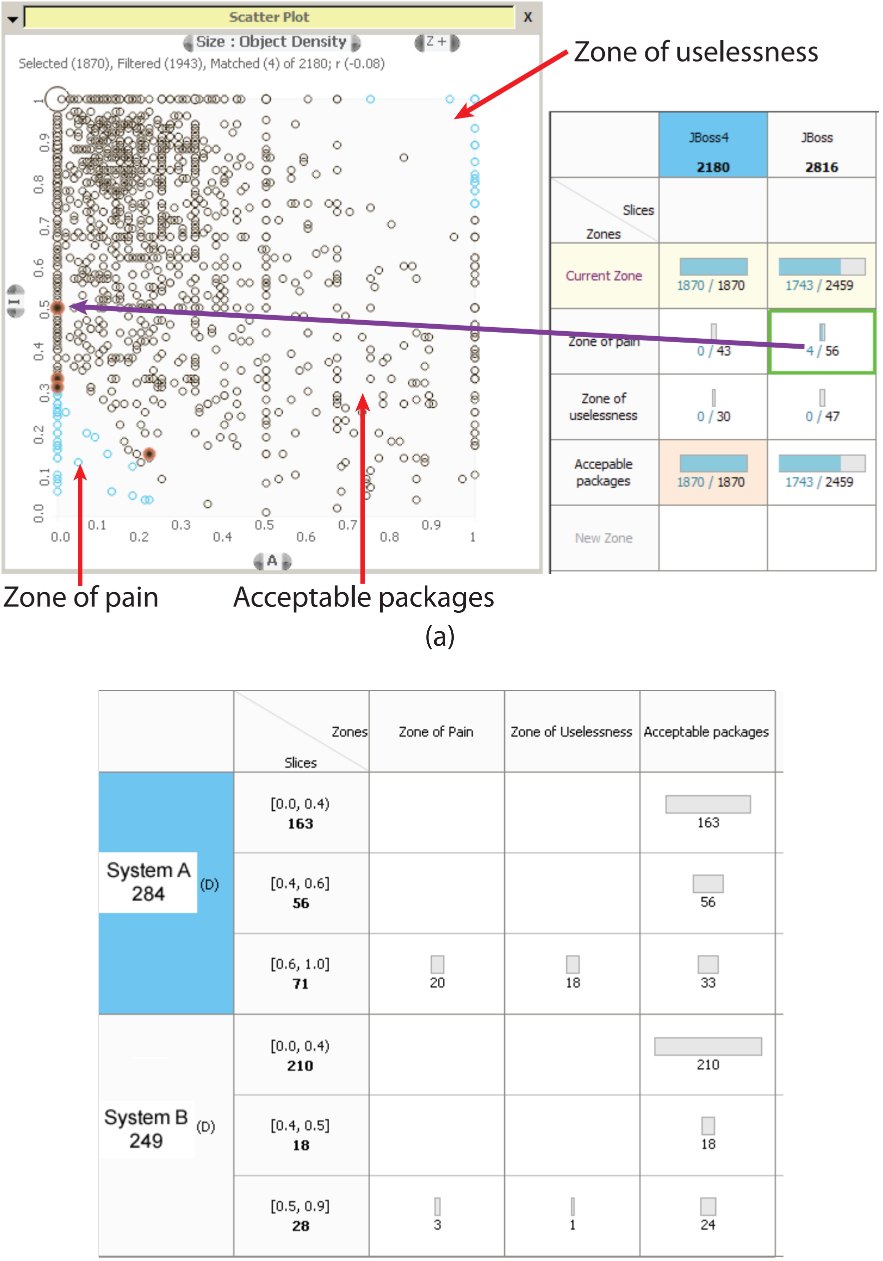
The first analyst is a software quality consultant at the Laboratory for Quality Software, TU/e, The Netherlands. He derives software metrics, package structure and call-graphs for software systems from source-code and visualizes them to check their design quality. There are ten package design principles for developing an ideal package structure for a software system [92]. Software quality analysts often use two metrics to study the quality of a package design: the stability metric (I), which measures the stability of dependencies, and the abstractness metric (A) [Roubtsov, personal communication]. There are three zones based on the relationship between A and I (Figure 4.11a). A *zone of pain*, where A and I are close to 0, contains packages that are rigid and cannot be changed or extended. A *zone of uselessness*, where A and I are close to 1, contains packages that are abstract and have no dependencies. The *acceptable packages* are close to the diagonal line connecting (A=0, I=1) and (A=1, I=0).

The analyst used Aruvi to compare two versions of JBoss, an enterprise application server. Initially, he loaded two datasets — *JBoss 4.0* and a recent version of *JBoss* (JBoss 4.3) into Aruvi. He started exploring the JBoss4 dataset using a scatterplot. He plotted A along the x-axis and I along the y-axis. Using this view, he defined three zones — *Zone of pain*, *Zone of uselessness* and *Acceptable packages* in the Select & Slice table. Using these definitions, he carried out two different analyses.

In the first analysis, he constructed a Select & Slice by slicing the two JBoss datasets with the three zones. He compared the recent version of JBoss (JBoss 4.3) against the previous version (JBoss 4.0) using the table. For this, he selected the acceptable packages of JBoss4, and switched to comparison mode. The Select & Slice table in Figure 4.11a shows this comparison. Based on this comparison, he studied the evolution of packages across two versions. He found that four of the acceptable packages from JBoss 4.0 have moved to the zone of pain in JBoss 4.3. He highlighted the four packages in the scatterplot (Figure 4.11a) that visualizes the JBoss 4.0 dataset. From this, he identified that one of the four packages was strongly affected (see purple arrow in Figure 4.11a); while the other three were already in the borderline. He hypothesized that the package might be strongly affected due to the changes made to incorporate some new features.

In the second analysis, he studied two finance management software systems from two different vendors — System A and System B (their names are sanitized), using the same approach as in the previous analysis. He reused the zones definition from the previous analysis. He compared the two systems by comparing the number of packages in the Zone of Pain and Zone of Uselessness. Based on these numbers, he found that system B has a good package design compared to system A.

The analyst usually follows a mathematical approach based on the normalized dis-



Selected (1870), Filtered (1943), Matched (4) of 2180; r (-0.08)

Zone of uselessness

Zone of pain

Acceptable packages

Zones \ Slices	JBoss4 **2180**	JBoss **2816**
Current Zone	1870 / 1870	1743 / 2459
Zone of pain	0 / 43	4 / 56
Zone of uselessness	0 / 30	0 / 47
Accepable packages	1870 / 1870	1743 / 2459
New Zone		

(a)

	Zones \ Slices	Zone of Pain	Zone of Uselessness	Acceptable packages
System A 284 (D)	[0.0, 0.4) **163**			163
	[0.4, 0.6] **56**			56
	[0.6, 1.0] **71**	20	18	33
System B 249 (D)	[0.0, 0.4) **210**			210
	[0.4, 0.5] **18**			18
	[0.5, 0.9] **28**	3	1	24

(b)

Figure 4.11: Software quality analysis. (a) Comparison of two different versions of JBoss, an enterprise application server. (b) Comparison of two different financial management software systems.

tance (D_n) to the diagonal line for identifying the acceptable packages. According to this approach [111], packages that have $D_n < (\mu_{D_n} + 2\sigma_{D_n})$ are acceptable packages. However, this approach cannot explicitly identify if a package belongs to a zone of pain or zone of uselessness. To verify this approach, he constructed a new Select & Slice table by slicing the three zones with six data subsets (3 subsets for both systems). He divided the two datasets based on their D_n attribute (D) into three bins, using the standard deviation clustering method. This table is shown in Figure 4.11b. He found that some of the packages are found acceptable in the Select & Slice table, even when $D_n > (\mu_{D_n} + 2\sigma_{D_n})$. Also, he could locate and visualize these packages in the scatterplot, to support this claim. Thus, in addition to validating the results using the mathematical approach, he could also explicitly identify the packages and understand their distribution using the Select & Slice table and the data view.

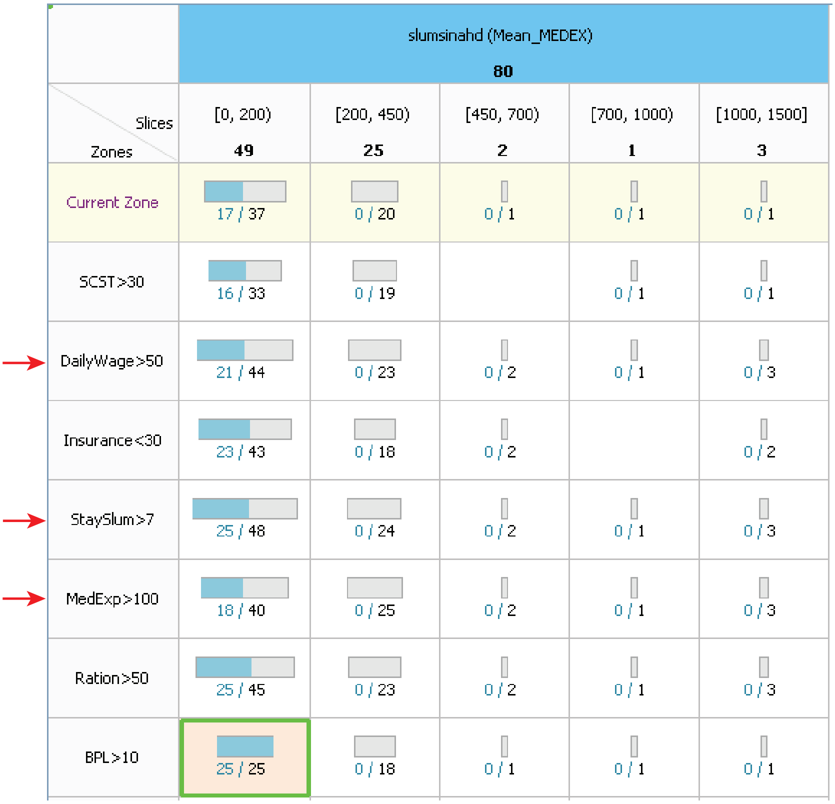
Afterwards, we asked the analyst to explain the key aspects of the Select & Slice table that made a difference in his analysis process. He said that "defining zones using lasso selection in the scatterplot to analyze data based on design principles was a quite handy and natural way of doing analysis. I could also verify the zones approach with our mathematical approach."

4.4.2 Social Data Analysis

The second analyst is an urban planner working at the Centre for Environmental Planning & Technology University, Ahmedabad, India. He investigated socio-economic data for slums in Ahmedabad using Aruvi. His analysis had two main goals: to understand the factors affecting the medical expenses of people in slums, and to understand the reason behind such trends.

For this analysis, he loaded the socio-economic data (*slumsinahd*) into Aruvi, and explored it using a scatterplot. During the exploration process, he identified 7 factors based on the demographics and socio-economic indicators to locate slums having poor living conditions. He used dynamic query widgets to specify these factors, and externalized these into 7 separate zones. They are the percentage of economical backward class people in a slum (*SCST* $>$ *30%*), the number of people having temporary jobs (*daily wage* $>$ *50*), the number of uninsured people (*insurance* $<$ *30*), the number of people who have stayed in slums over 7 years (*stayslum* $>$ *7*), the monthly medical expenses (*medexp* $>$ *100* Indian Rupee - INR), the number of people who have access to the public distribution system (*Ration* $>$ *50*) and the number of people below the poverty line (*BPL* $>$ *10*).

To understand the trends in the medical expenses, the analyst divided the dataset using mean monthly medical expenses into 5 custom interval bins. The Select & Slice table in Figure 4.12 shows an overview of the relationship between the 7 factors and the mean monthly medical expenses. He found that around 60% of slums (49 out of 80) have mean monthly expenses below 200 INR. Most of these slums fell under the poor socio-economic conditions described in zones such as daily wage (44 out of 49 slums) and stay in slums over 7 years (48 out of 49 slums). Then he compared the slums below the poverty line (BPL $>$ 10, highlighted in Figure 4.12) against the other factors in the first column. He found that for most of the slums below the poverty line (18 out of 25 slums) the monthly medical expenses constituted more than 50% of their monthly earnings (BPL



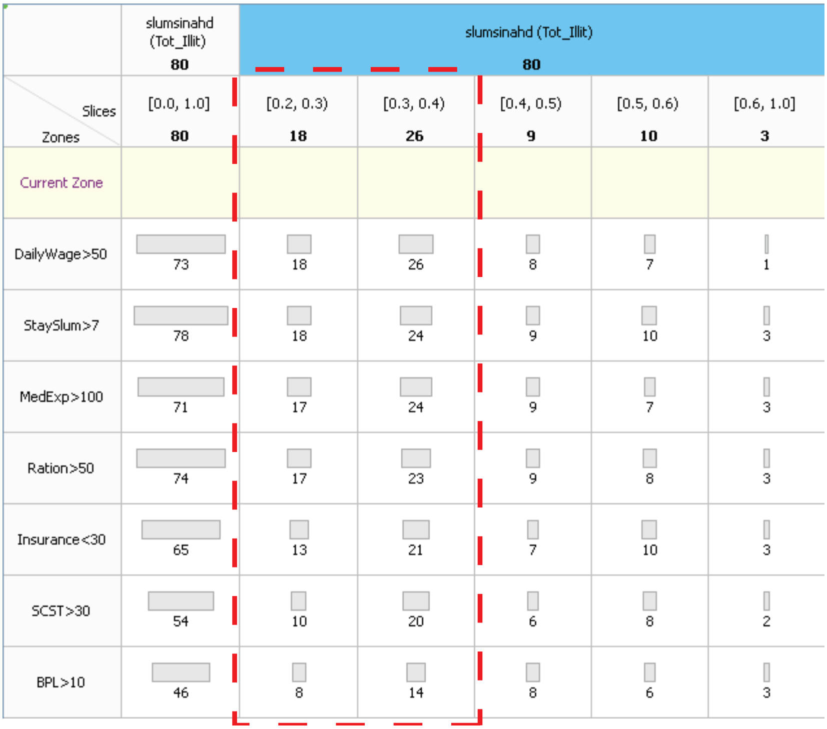
	slumsinahd (Mean_MEDEX) 80				
Slices Zones	[0, 200) 49	[200, 450) 25	[450, 700) 2	[700, 1000) 1	[1000, 1500] 3
Current Zone	17 / 37	0 / 20	0 / 1	0 / 1	0 / 1
SCST>30	16 / 33	0 / 19		0 / 1	0 / 1
DailyWage>50	21 / 44	0 / 23	0 / 2	0 / 1	0 / 3
Insurance<30	23 / 43	0 / 18	0 / 2		0 / 2
StaySlum>7	25 / 48	0 / 24	0 / 2	0 / 1	0 / 3
MedExp>100	18 / 40	0 / 25	0 / 2	0 / 1	0 / 3
Ration>50	25 / 45	0 / 23	0 / 2	0 / 1	0 / 3
BPL>10	25 / 25	0 / 18	0 / 1	0 / 1	0 / 1

Figure 4.12: Socio-economic data investigation: Medical Expenses trend analysis.

in Ahmedabad is at 436 INR per month. The monthly medical expenditure is about 200 INR).

By analyzing the table in Figure 4.12, he hypothesized that the high level of temporary jobs (like daily waged labor, unskilled labor) are because of the illiteracy prevailing in the slums. Subsequently, they are not able to improve their economic background as they do not have access to better education and training. Therefore, they are stuck in poor living conditions. However, the poor living conditions lead to high medical expenses. To prove these hypotheses, he projected these zones on the dataset divided using total illiteracy rate into 5 custom interval bins above 20% of total illiteracy rate (20% of the population is elderly and kids). The new Select & Slice table is shown in Figure 4.13. He found that most of the slums have between 20 and 40% total illiteracy rate. Also all these slums have a high number of people with temporary jobs and high monthly medical expenses. Based on this view, he could affirm his hypotheses. He also concluded that these slums are the intrinsic vulnerable slums which are vulnerable to even small fluctuations in the socioeconomic conditions.

During the informal interview session, the analyst explained the key differences made



	slumsinahd (Tot_Illit) 80	slumsinahd (Tot_Illit) 80				
Slices Zones	[0.0, 1.0] 80	[0.2, 0.3) 18	[0.3, 0.4) 26	[0.4, 0.5) 9	[0.5, 0.6) 10	[0.6, 1.0] 3
Current Zone						
DailyWage>50	73	18	26	8	7	1
StaySlum>7	78	18	24	9	10	3
MedExp>100	71	17	24	9	7	3
Ration>50	74	17	23	9	8	3
Insurance<30	65	13	21	7	10	3
SCST>30	54	10	20	6	8	2
BPL>10	46	8	14	8	6	3

Figure 4.13: Socio-economic data investigation: The relationship between illiteracy and the seven zones helped the analyst to identify the intrinsic vulnerable slums.

by the Select & Slice table in his analysis process. Usually, he uses Microsoft Excel for analyzing the data. He would study the effects of the factors one at a time; however, he could not analyze them simultaneously. Also, he noted that a pivot table cannot be used for this purpose, where items are partitioned over the row and column attributes. He said “possibly I could have done this in Microsoft Excel. However, I could have never done the analysis so quickly and without breaking my head. Slicing Zones by different subsets of data helped me to put all my conditions parallel and compare them simultaneously.” However, he felt that if these slum locations are plotted geographically, he could correlate the attribute values with other spatial accessibility functions, in order to make a better conclusion.

4.4.3 Wireless Sensor Network

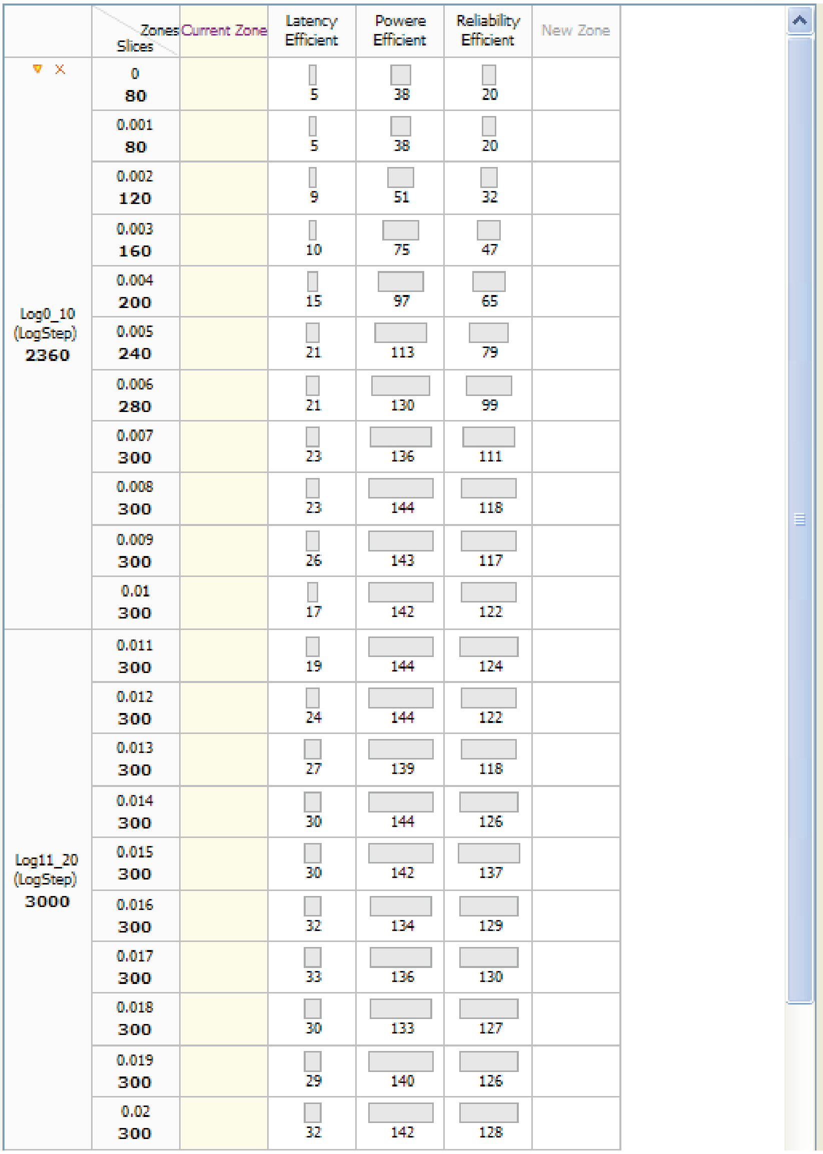
The third analyst is a graduate student at the Embedded Systems Institute, the Netherlands. One of his research goals is to identify optimal configurations for sensor nodes in a wireless sensor network. As the number of sensor nodes increases, the design space exploration for identifying optimal configurations becomes highly complex. For this, he and his colleagues [97] have come up with a set of configuration guidelines based on power, reliability and latency measures. He uses a genetic algorithm (GA) approach to seed the configurations based on a number of parameter on the nodes in the network. Each seed produces a set of configurations and the configuration's power, reliability and latency performance measures are derived. He visualizes these measures for each seed to inspect the performance of the configuration, and updates the GA seed parameters to seed a new configuration. For achieving optimal configurations, he typically seeds 400 configurations. During this design space exploration process, he has to maintain an overview of all the seed parameters and their performance. He maintains a note detailing the performance of each seed. However, he cannot compare on the performances of arbitrary seeds.

During this analysis, he loaded a dataset that contains the performance measures of 400 configurations for a sensor network having around 300 random sensor nodes into Aruvi. He plotted the dataset using two scatterplots: one comparing *latency* and *power* measures, and the other comparing *power* vs *reliability* measures. Using these scatterplots, he defined three zones: *Latency efficient*, *Power Efficient* and *Reliability Efficient*. He divided the dataset based on the *configuration_id* into 400 data subsets. Using the Select & Slice table, he retrieved items from these subsets for the three zones (Figure 4.14). By simply scrolling down the table and rearranging data subsets, he could understand the performance of GA and quality of the configurations seeded by the algorithm. Finally, he said that "I could quickly change the definition of the zones and compare the performances of the GA seeds, which I never could study previously. Moreover, I don't have to keep track of the performance details of each seed, as I could easily get an overview of them from the Select & Slice table."

4.4.4 Who are the best skaters?

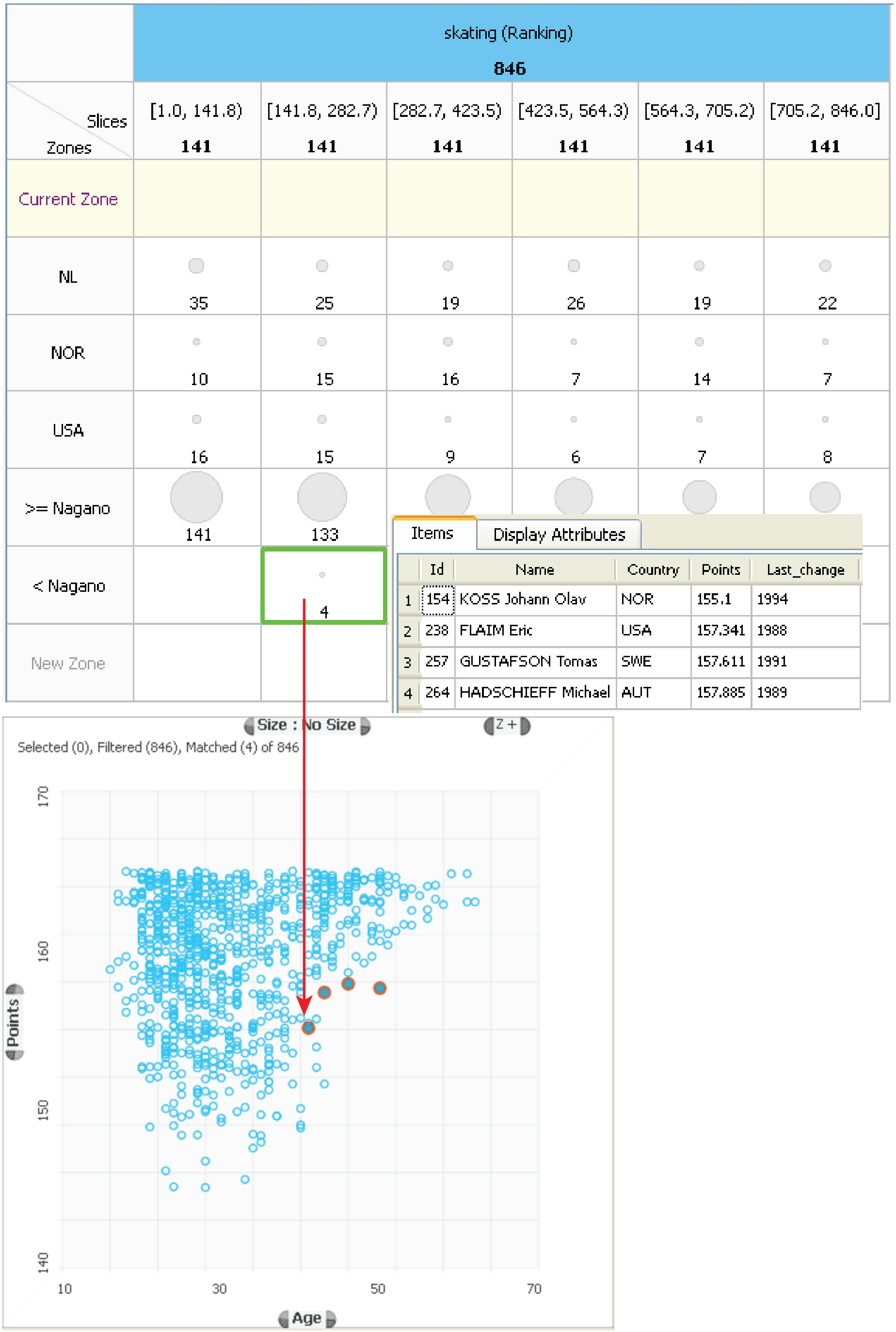
The fourth analyst, who is a fan of speed skating, investigated a speed skating dataset. He is also an assistant professor at the mathematics and computer science department, TU/e and teaches information visualization. The dataset contains male all round performers in both short and long distance speed skating. He was interested in identifying the best skaters before the clap skates were introduced to the international skating competitions. He was sure that clap skates were used at 1998 Winter Olympics in Nagano, Japan due to which many world records were broken. Also, there was an assumption that the skaters from Norway, the Netherlands and USA have high success rates at world championships.

To verify these hypotheses, he created five zones - three for the countries and one for skaters who appeared before the Nagano Olympics and one for skaters who appeared at and after the Nagano Olympics (see Figure 4.15). He projected these five zones on to the six equal interval bins of ranking attribute. He used a scatter plot for comparing points achieved (cumulative of times at 500m, 1000m, 5000m and 10000m in seconds) against the age of the skaters (see Figure 4.15). He compared the performance of the skaters from the three countries by alternatively highlighting skaters from the high ranking bin of these countries (first dataset column in the table) in the scatter plot. He found that though the Netherlands had more people in the high ranking bins, none of them converted them into a championship award. However, Norway with few participants was moderately successful. The rows '>= Nagano' and '< Nagano' in the table prove the drastic shift in the ranking after the introduction of clap skates in 1998.



	Zones Slices	Current Zone	Latency Efficient	Powere Efficient	Reliability Efficient	New Zone
Log0_10 (LogStep) 2360	0 80		5	38	20	
	0.001 80		5	38	20	
	0.002 120		9	51	32	
	0.003 160		10	75	47	
	0.004 200		15	97	65	
	0.005 240		21	113	79	
	0.006 280		21	130	99	
	0.007 300		23	136	111	
	0.008 300		23	144	118	
	0.009 300		26	143	117	
	0.01 300		17	142	122	
Log11_20 (LogStep) 3000	0.011 300		19	144	124	
	0.012 300		24	144	122	
	0.013 300		27	139	118	
	0.014 300		30	144	126	
	0.015 300		30	142	137	
	0.016 300		32	134	129	
	0.017 300		33	136	130	
	0.018 300		30	133	127	
	0.019 300		29	140	126	
	0.02 300		32	142	128	

Figure 4.14: Wireless sensor network - design space exploration analysis



Zones \ Slices	skating (Ranking) 846					
	[1.0, 141.8) 141	[141.8, 282.7) 141	[282.7, 423.5) 141	[423.5, 564.3) 141	[564.3, 705.2) 141	[705.2, 846.0] 141
Current Zone						
NL	35	25	19	26	19	22
NOR	10	15	16	7	14	7
USA	16	15	9	6	7	8
>= Nagano	141	133				
< Nagano		4				
New Zone						

Items | Display Attributes

	Id	Name	Country	Points	Last_change
1	154	KOSS Johann Olav	NOR	155.1	1994
2	238	FLAIM Eric	USA	157.341	1988
3	257	GUSTAFSON Tomas	SWE	157.611	1991
4	264	HADSCHIEFF Michael	AUT	157.885	1989

Selected (0), Filtered (846), Matched (4) of 846

Figure 4.15: Best speed skaters before 1998 Nagano winter Olympics.

The analyst carried out a very simple analysis using the zones. He also agreed that he could have identified the best skaters by simply applying dynamic queries. However, he said “to have criteria in the table (as zones) helped me to do the analysis comfortably and wouldn’t lose focus on my analysis. Moreover, the before Nagano and from Nagano rows clearly show the difference created by the clap skates.” He mostly played around by selecting items in the table and highlighting these in the scatterplot. He said “selecting cells and highlighting items in the scatterplot is like applying quick or preset dynamic queries during the analysis.”

We also found the analysts used only the scatter plot as they didn’t find a need to use a bar chart, the other visualization supported in the Aruvi visualization system. Only, analysts 1 and 3 manipulated their zones after defining them. Also, three of the analysts recommended that the percentage of items in cells similar to the items of the origin cell can be displayed as labels in the comparison view. Overall, from the case studies, we conclude that the analysts mainly engaged in defining zones, retrieving items from different data subsets for zones, studying items distribution in the table, and highlighting items in visualizations by selecting cells in the table during their analysis. Thereby, they used zones to reason with their domain-specific hypotheses during interactive data exploration.

4.5 Conclusion

In this chapter, we presented the Select & Slice table that helps to cross-tabulate semantic zones and data subsets. Semantic zones are areas of interest in data space specified through conditions over data attributes or as functions of data attributes that have a clear semantic meaning. Using the Select & Slice table, users can define and manipulate zones; and understand the relationship between zones and data subsets, visually and interactively. In addition, they can drill-down to a particular data subset, and investigate items of the table in the data view using drag & drop and other simple mouse operations. They can also get an overview of the distribution of items in the table using a set comparison operation, and a keyword search. Finally, we presented four case studies that illustrated the support offered by the Select & Slice table for exploratory data analysis.

In the next chapter, we present our approach and tools to support users for gaining exploration awareness based on their action trails. Using these tools, analysts can get an overview of key aspects of the exploration process; as well as search and retrieve parts of the analysis processes for reviewing purposes.

Chapter 5

Exploration Awareness

One faces the future with one's past. — Pearl S. Buck

In Chapter 3, we presented a sensemaking framework for visual analytics. Using this framework, we enabled users to capture interesting aspects of the analysis such as notes, zones and visualizations. In addition, we also automatically captured the analysis process of the users using action trails. We used a history representation (discussed in Section 3.4.2) to provide an overview of the analysis process. However, the history tree representation is overly abstract to comprehend the interesting aspects of a lengthy analysis process. Hence, users cannot review previous analysis processes.

When users want to continue an analysis performed in the past, either their own or a collaborator's, they need an overview of what has been done and found so far. Such an overview helps them to gain a shared knowledge about each others' analysis strategy and continue the analysis. We aim to support users in this process, and thereby support their *exploration awareness*.

In this chapter, we consider three linked processes: overview, search and retrieve for developing exploration awareness during an analysis. Support for these processes is added to the sensemaking framework for visual analytics discussed in chapter 3. To support these processes, we first present a user's information interest model that captures key aspects of the exploration process. Next, using these key aspects we provide interactive exploration overviews, and enable analysts to retrieve parts of past analyses using keyword and similarity based search mechanisms; to enable them to review past analyses. Finally, we present three case studies and discuss the support offered by the three linked processes for developing exploration awareness.

5.1 Introduction

Analysis is often a collaborative process [122]. During collaborative visual analysis, analysts need to develop exploration awareness — an overview of what has been done and found, by themselves or by their collaborators'. Such an overview helps them to establish a common ground for sharing their analysis, and defending their judgments. Analysts working at the same place and time can directly observe their collaborators' analysis strategy, however, during synchronous collaboration at different places or asynchronous collaboration at different places and times, it is difficult for them to understand their collaborators' analysis strategy. Also, when they resume their own analysis later, they may have limited recall of the key aspects of their previous analyses.

To understand the analysis strategy of a collaborator and continue his analysis, analysts must be enabled to get an overview of the key aspects of the previous exploration processes, and search and retrieve past visualizations for reviewing findings. While retrieving past visualizations, they have to be aware of what has been done and found around these visualizations to reason about the collaborators' analysis strategy. We summarize our aim to support the process of getting an overview of what has been done and found during an analysis as *increasing exploration awareness*.

Current visualization systems offer support for recovering from mistakes, archiving interesting visualizations, and reusing archived visualizations. Most of these systems offer limited support to users for retracing visualizations from past analysis that are neither bookmarked nor annotated. Keyword search is widely used to retrieve documents, images and videos on the Internet. A similar approach can be useful for retrieving visualizations from past analysis, by enabling users to retrieve visualizations using keywords, for instance based on labels of selected items, names of attributes used for axes, and filters employed during the exploration process. Query by example is another popular approach that helps to retrieve documents and images based on content similarity. In this chapter, we present models and tools that enable analysts to develop exploration awareness for reviewing a visual analysis by exploiting exploration overviews and keyword and similarity searches.

5.2 Related Work

We now consider previous work on exploration models and retrieval mechanisms in visualization and visual analytics tools.

5.2.1 Exploration Model

Heer et al. [66] provide an overview of design considerations for history models in information visualization. The interaction history can be modeled as a sequence of actions, states, or both. HARVEST [60] captures the user interaction history as an action trail. The action trail representation is optimized by grouping consecutive similar actions. Users can bookmark action trails, and revisit and reuse them. Jankun-Kelly et al. [71] model visualization states as sets of parameters, and actions as transformations of these parameters. They also present a derivation model to identify intermediate steps between two states.

Aruvi uses a hybrid action-state model to capture the interaction history as discussed in Section 3.4.2. A detailed state description is captured to provide readable information that highlights changes to a visualization state. A history tree representation is used to visualize the interaction history. Users can revisit and reuse a state, and attach notes to it. Tableau [66], a commercial visualization system, records states as VizQL statements and maintains an action log. User actions are grouped based on a custom classification, based on the Tableau visualization system. It also optimizes history management using an undo-as-delete metaphor and some chunking rules. A sequence of thumbnails is used to represent this optimized interaction history.

The above visualization state models focus on optimizing the archival of an exploration process. Their visualization state description only supports recreation of the visualization state. However, the description does not explicitly capture key aspects that represent the users' information interest during the exploration process.

5.2.2 Retrieval Mechanism

The history mechanisms in most data analysis tools offer support for recovering from mistakes [44] and analyzing exploration session logs [58, 85]. Commercial visualization tools such as Spotfire and MagnaView support archiving interesting visualizations and reusing them. However, they do not support retrieval of visualizations from the past analysis that are neither bookmarked nor annotated.

Vistrail [28] helps to retrieve the visualization dataflows based on the specifications to the dataflows and user actions. Tableau [66] offers support to retrieve visualizations from past analysis based on data fields used in visualizations and visualization types. However, it does not provide an overview of key aspects of an analysis for developing exploration awareness. Just a list of visualizations matching keywords seems to be too limited to fully support a review process [63].

Sense.us [67], a web based information visualization system supports asynchronous collaboration by sharing views and discussions. It uses a similarity mechanism to identify identical visualization views that have different parametric representations and attach discussions from other collaborators to those views. Yang et al. [140] discuss a similarity metric for visual queries in visualization systems for optimizing archival of interesting and repeated user queries while exploring large datasets. However, these approaches do not show an overview of similar visualization states for supporting a review process.

In the following sections, we present our approach and solutions that enable users to develop exploration awareness on a visual data analysis.

5.3 Approach

To support users to develop exploration awareness during visual data analysis, we look at how a reader develops awareness on a book's content. A table of contents, a list of figures and tables, and a keyword index provide an overview of key aspects of the book. In addition, the table of contents provides an overview of the structure of the book. These key aspects also have page numbers attached to them that help readers to easily transit from the overview to the corresponding detailed information inside the book.

When they want to review the book or retrieve specific pieces of information from it, they iteratively search through it using the overview of the key aspects and the links to the detailed information inside it. In a digital version of the book, a keyword search also helps to retrieve specific pieces of information. The keyword search results are shown with some metadata that help the readers to narrow down the search results. Hence, there are three linked processes: overview, search and retrieve that help the readers to develop awareness of a book. A visual data analysis that has to be understood is similar here to the contents of a book. It has detailed information about what has been done and found so far by users. To support developing exploration awareness on a visual data analysis, we argue that the users must be enabled to perform the three linked processes shown in Figure 5.1.

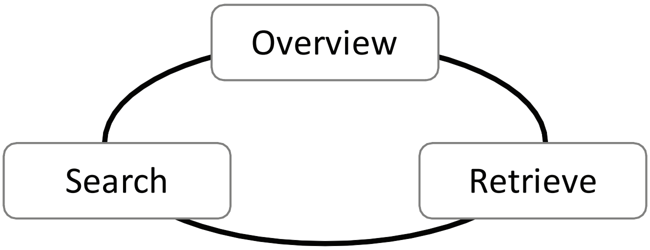


Figure 5.1: The three linked processes for supporting exploration awareness.

Hence, we argue that the users must be provided with the following components to support the three linked processes for developing exploration awareness in visual analytics:

- Overview: visual representation(s) that provide an overview of the structure and key aspects of the exploration process;
- Search: A keyword based and similarity based visualization retrieval mechanism; and
- Retrieve: visual representation(s) that provide an overview of the search results and help users to retrieve specific visualizations from the analysis.

In the following sections, we first present a user's information interest model that captures the key aspects of the exploration process. These key aspects are indexed during an exploration process. Next, we provide our solutions to present an overview of the exploration. Following that we present a keyword based and a similarity based visualization retrieval mechanism.

5.4 User's Information Interest Model

Gotz and Zhou [60] classify user actions in a visual analytic system into three broad categories: exploration actions; insight actions; and meta actions. Exploration actions alter the data and visualization specifications in a visual analytics system and create new visualization states. Insight actions enable users to record annotations, bookmark visualizations or organize notes recorded during an analysis. Meta actions enable users to revisit, undo, redo, delete or edit a past exploration or insight action.

We argue that exploration actions represent changes in a user's information interest, besides, producing new visualization views. For this, we take a closer look into the interactive visualization model [128] and information visualization pipeline [30].

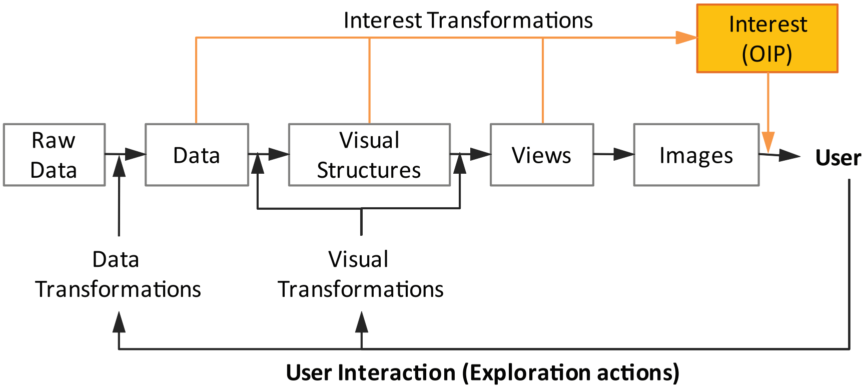


Figure 5.2: The visualization pipeline, based on Card's model [30], modified to show interest transformations. The Object Interest Profile (OIP) is used to highlight objects in the resulting images of an exploration action.

In interactive visualization, a dataset D is transformed into an image I based on a specification S given by the user. S consists of two components - data transformations (d) that denote what subset of the data has to be shown, and visualization transformations (v) that denote how it has to be shown. The former includes for instance filters and clustering used; the latter includes the type of plot(s) and visual encoding used, for instance the axes selected for a scatterplot. By specifying S, a user implicitly attaches some degree of interest to the objects of the dataset. We refer to these as *interest transformations* (see Figure 5.2). The degree of interest can range from none (items not selected for visualization) to high (items manually picked in visualization), and have values in between. We call a list of degrees of interest per object an Object Interest Profile (OIP). Hence, in addition to the image I_t, the specification S_t also transforms the OIP. Therefore, when users provide specifications S_t based on the current knowledge K_{t-1}, a visualization system generates I_t and changes OIP_t. Figure 5.3 shows the visualization state at time t.

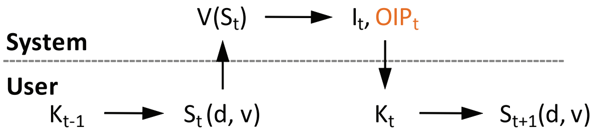


Figure 5.3: The visualization state in the user navigation at time t.

We use a simple model for the Object Interest Profile. It has three levels, based on the specifications to a visualization system through exploration actions. When data objects are not visible, because they are either filtered out through attribute criteria, moved out of the viewing area of visualization using zoom-in and pan actions, or collapsed in hierarchical or graph views, they are assigned a low interest value. When data objects are either directly selected on a visualization view or when clusters containing these objects are selected, they are assigned a high interest value. Visible, but non-selected objects are assigned a medium interest value. Both data and visual transformations can change the OIP of a dataset.

Thus, a user's exploration process can now be described at the system level as a set of specifications to a visualization system S = S$_n$(d, v): 0≤n≤t; a set of images I = I$_n$: 0≤n≤t; and a set of object interest profiles OIP = OIP$_n$: 0≤n≤t. Hence, we selected the following four key aspects to describe the exploration process:

1. the visualization and data transformations (S);

2. the data dimensions specified through S;

3. viewed objects (medium interest objects from OIP); and

4. selected objects (high interest objects from OIP).

Based on this user's information interest model, we have redesigned the history mechanism of Aruvi to capture the key aspects of an exploration process. In addition, we added support for multiple users within an analysis to facilitate asynchronous collaboration.

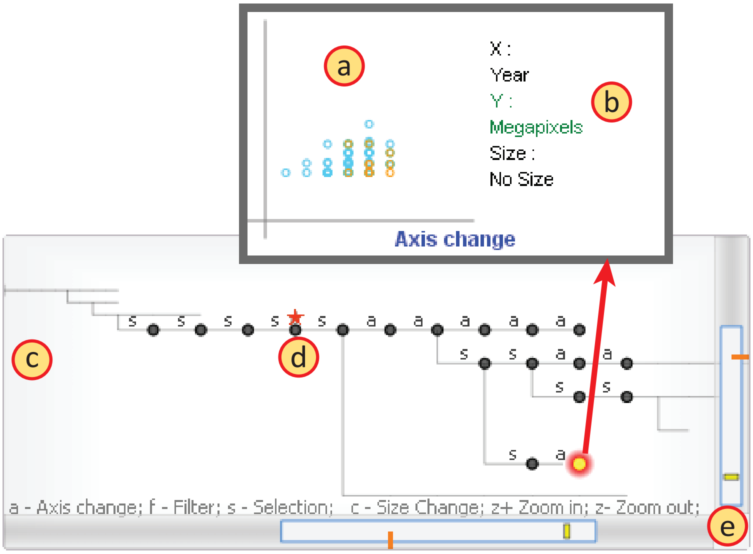
5.5 Exploration Overview

We have designed two different means to enable the user to get an overview of an exploration process: the structure overview, with an emphasis on the process; and the key aspects overview with an emphasis on the contents.

5.5.1 Structure Overview

The structure overview can be provided based on a user's action trail. In Aruvi (Chapter 3), a history tree representation is used for this. A branch in the history represents a revisit and reuse of a past visualization state. A node represents a visualization state and an edge between the adjacent nodes is labeled with the user action. In Tableau [66], a thumbnail strip is used to represent a user's action trail. Each thumbnail has a text description of the user's action. In Harvest [60], a sequence of action labels is used to represent a user's action trail. Each label has a thumbnail tool tip that represents the corresponding visualization state. A thumbnail strip provides an easy to understand overview, and we have added this feature to Aruvi as well.

A bare-bone history tree representation, with just labels for user interactions on the edges, is abstract. To support the user, we added a thumbnail tool tip to show the state more understandably (Figure 5.4(a)). The thumbnails are generated using a similar approach as presented in [66]. In addition to a text description of the user action, each thumbnail has a list of key aspects of the visualization state on the right. Key aspects that have changed compared to the previous visualization state are highlighted in green (Figure 5.4(b)).



X :
Year
Y :
Megapixels
Size :
No Size

Axis change

a - Axis change; f - Filter; s - Selection; c - Size Change; z+ Zoom in; z- Zoom out;

Figure 5.4: History tree representation. (a) Thumbnail tool tip for the visualization state highlighted orange. (b) A new key aspect (Megapixels) compared to the previous visualization state is highlighted in green. (c) Overview area. (d) Focus area. (e) Horizontal and vertical scrollbars help the user to change the focus area, and highlight the location of the current visualization state (yellow bar) and a note (orange bars).

During a large and complex exploration process, it is difficult to get an overview of the exploration process if all the visualization states are displayed as thumbnails or as a history tree. Based on feedback from users [117], we developed a focus + context technique in the history tree that enables a user to focus on a few visualization states around a certain visualization state with an overview of the entire exploration process. In the overview area, only the structure of the exploration process is shown without details about the visualization states (Figure 5.4(c)). In the focus area, details about visualization states and user actions are shown (Figure 5.4(d)). A vertical scrollbar and a horizontal scrollbar attached to the history tree (Figure 5.4(e)) enable the user to change the focus area. A yellow line on the scroll bar indicates the location of the current visualization state; orange lines on the scroll bar indicate visualization states with a note.

Complex analysis processes can involve an extended time period and many users. To support users to focus on a specific time period, they can specify this through a time interface (Figure 5.5(a)), and also they are enabled to focus on a subset of all users, via the users list (Figure 5.5(b)). Both these options can be used for all exploration awareness tasks. For the history tree, the focus provided is used to constrain the overview given.

5.5.2 Key Aspects Overview

We identified four key aspects of the exploration process based on our user's information interest model: visualization and data transformations, data dimensions specified through S, selected objects and viewed objects. The key aspects overview shows the most important items for each key aspect using tag cloud representations (Figure 5.6). The size of an item denotes its importance and is derived from its frequency of occurrence during the exploration process. The items are sorted in descending order based on their frequency of occurrence within a key aspect. The key aspects overview is provided for a time period specified through the time interface (Figure 5.5a) from a group of users' analysis selected from the users list (Figure 5.5b).

We display the four key aspects one below another, instead of integrating them in one list. With this, users can quickly see which visualizations tools were used most, which data aspects were investigated most with these tools and which data objects got most attention during that investigation. For example, Figure 5.6 from the first case study in section 5.8, tells us that this user used scatterplots to study bicycles, buses, a particular range of incomes, and that he focused on a specific set of (geometric) zones. Also, users can drill-down via this overview. By selecting items, the focus is limited to visualizations where these items are used, and the tag-clouds are adapted accordingly (see Figure 5.12a and Figure 5.12b).

Frequency counts give relevant information and show global patterns, but we acknowledge that they do have limitations. If a user has seen something special in just one view, and has not marked this item or made a note, the frequency counts do not help to find these back. Also, it can happen that part of the analysis was wrong. Concerning this, we assume analysts remove mistakes or wrong analysis path using the undo mechanism or directly deleting visualization states via the history tree representation. However, if analysts missed to identify wrong analysis paths, they can identify these mistakes in the key aspects overview, when an unexpected or unwanted item gets too much emphasis. In addition, some chunking rules are used to optimize the capturing of the visualization states such as grouping a quick succession of filter actions. This chunking helps to reduce the increasing importance of an item due to redundant specifications by the user.

5.6 Keyword based Search and Retrieval

We enable users to perform keyword search on all text that plays a role in visualization and data transforms, data aspects, viewed and selected objects; and notes in the knowledge view. Keywords are matched with labels of items and names of attributes used for axes, visualization types, filters, selection and filtered objects during the exploration process. States that match a keyword search are highlighted (in green) in the history tree view (Figure 5.7a), the thumbnail view (Figure 5.7b) and the exploration overview (Figure 5.8e). The keyword search is confined to a time period specified through the time interface (Figure 5.5a) and to a group of users selected from the users list (Figure 5.5b). In the search interface (Figure 5.8a), each keyword typed by a user is color coded, and keywords are separated by a plus symbol.

During a review process, users might be interested in combinations of keywords. In

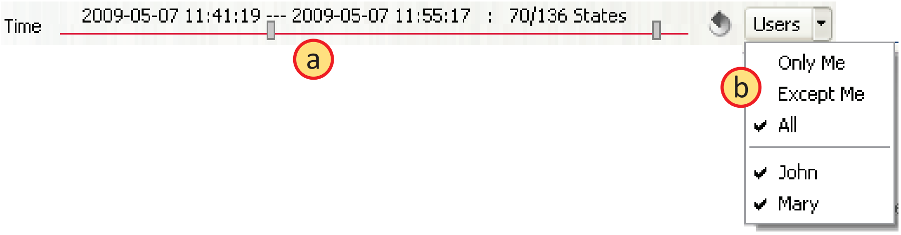
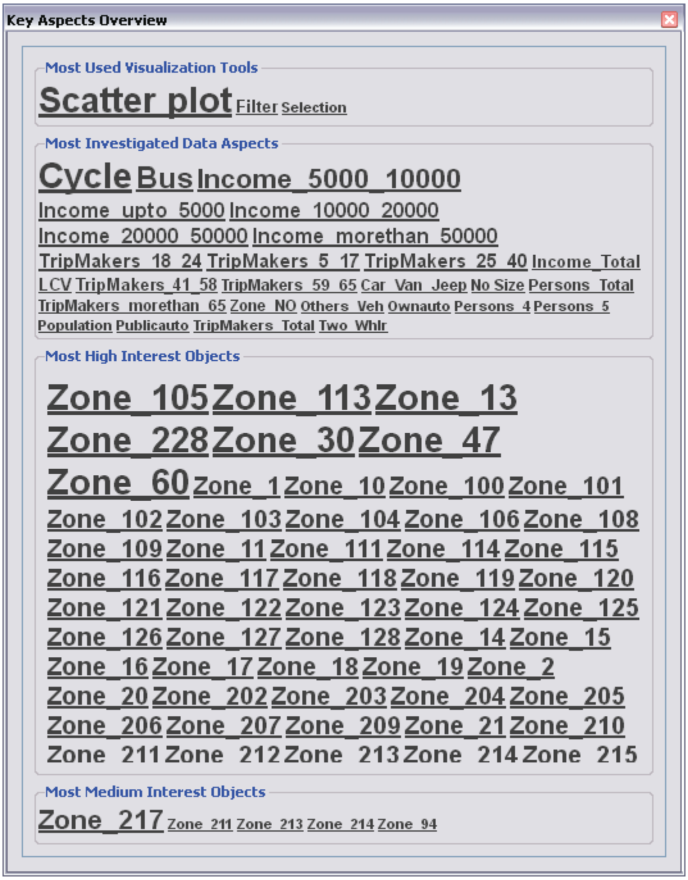


Figure 5.5: (a) Time interface. (b) Users list



Key Aspects Overview

Most Used Visualization Tools

Scatter plot Filter Selection

Most Investigated Data Aspects

Cycle Bus Income_5000_10000
Income_upto_5000 Income_10000_20000
Income_20000_50000 Income_morethan_50000
TripMakers_18_24 TripMakers_5_17 TripMakers_25_40 Income_Total
LCV TripMakers_41_58 TripMakers_59_65 Car_Van_Jeep No Size Persons_Total
TripMakers_morethan_65 Zone_NO Others_Veh Ownauto Persons_4 Persons_5
Population Publicauto TripMakers_Total Two_Whlr

Most High Interest Objects

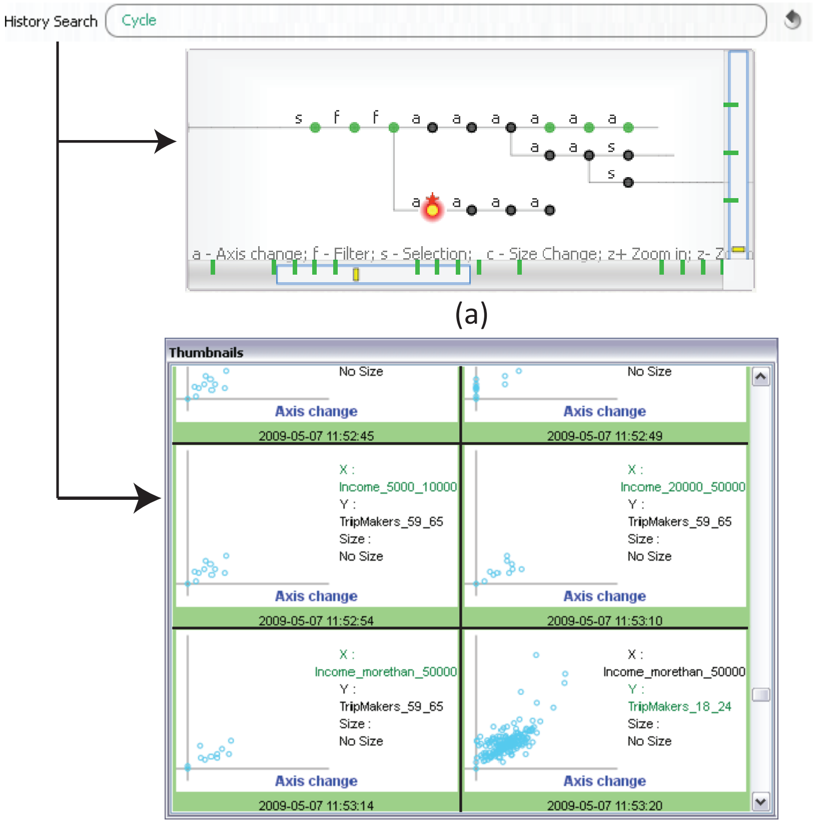
Zone_105 Zone_113 Zone_13
Zone_228 Zone_30 Zone_47
Zone_60 Zone_1 Zone_10 Zone_100 Zone_101
Zone_102 Zone_103 Zone_104 Zone_106 Zone_108
Zone_109 Zone_11 Zone_111 Zone_114 Zone_115
Zone_116 Zone_117 Zone_118 Zone_119 Zone_120
Zone_121 Zone_122 Zone_123 Zone_124 Zone_125
Zone_126 Zone_127 Zone_128 Zone_14 Zone_15
Zone_16 Zone_17 Zone_18 Zone_19 Zone_2
Zone_20 Zone_202 Zone_203 Zone_204 Zone_205
Zone_206 Zone_207 Zone_209 Zone_21 Zone_210
Zone 211 Zone 212 Zone 213 Zone 214 Zone 215

Most Medium Interest Objects

Zone_217 Zone_211 Zone_213 Zone_214 Zone_94

Figure 5.6: Key Aspects Overview.

the current form, the feedback given in the history tree and thumbnail views is limited. The history tree view offers few possibilities to show the occurrence of multiple keywords and their association to the key exploration aspects. The thumbnail view takes much screen space and addition of visual elements to show the occurrence of multiple keywords and their association to the key exploration aspects will clutter the thumbnail view. Therefore, we introduce a metadata view, based on a table metaphor, to visualize the search results and their association to the key exploration aspects.



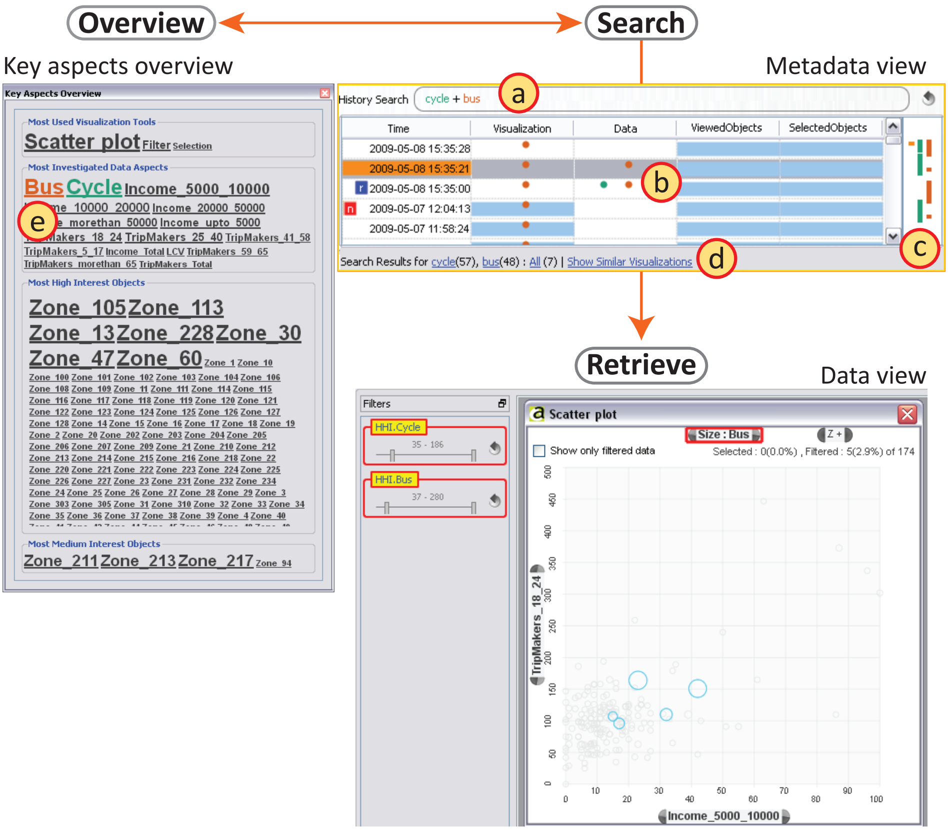
(a)

(b)

Figure 5.7: Visualization of the keywords search results. (a) The history tree and (b) the thumbnail view highlight visualization states containing the keyword *cycle* in green.

5.6.1 Metadata View

The Metadata view visualizes the changes to the visualization and data transformations and the influence of these changes to OIPs during an exploration process (see Figure 5.8). It has five columns: *time*, *visualization*, *data*, *viewed* and *selected objects*. Each row represents a visualization state. When users create new visualization states, rows are added at the bottom of the metadata view. Cells are filled in with light blue to indicate changes in key aspects due to user interaction.


Overview
Search
Key aspects overview
Metadata view
Key Aspects Overview
Most Used Visualization Tools
Scatter_plot Filter Selection
Most Investigated Data Aspects
Bus Cycle Income_5000_10000
Income_10000_20000 Income_20000_50000
Income_morethan_50000 Income_upto_5000
TripMakers_18_24 TripMakers_25_40 TripMakers_41_58
TripMakers_5_17 Income_Total LCV TripMakers_59_65
TripMakers_morethan_65 TripMakers_Total
Most High Interest Objects
Zone_105 Zone_113
Zone_13 Zone_228 Zone_30
Zone_47 Zone_60 Zone_1 Zone_10
Zone_100 Zone_101 Zone_102 Zone_103 Zone_104 Zone_106
Zone_108 Zone_109 Zone_11 Zone_111 Zone_114 Zone_115
Zone_116 Zone_117 Zone_118 Zone_119 Zone_120 Zone_121
Zone_122 Zone_123 Zone_124 Zone_125 Zone_126 Zone_127
Zone_128 Zone_14 Zone_15 Zone_16 Zone_17 Zone_18 Zone_19
Zone_2 Zone_20 Zone_202 Zone_203 Zone_204 Zone_205
Zone_206 Zone_207 Zone_209 Zone_21 Zone_210 Zone_212
Zone_213 Zone_214 Zone_215 Zone_216 Zone_218 Zone_22
Zone_220 Zone_221 Zone_222 Zone_223 Zone_224 Zone_225
Zone_226 Zone_227 Zone_23 Zone_231 Zone_232 Zone_234
Zone_24 Zone_25 Zone_26 Zone_27 Zone_28 Zone_29 Zone_3
Zone_303 Zone_305 Zone_31 Zone_310 Zone_32 Zone_33 Zone_34
Zone_35 Zone_36 Zone_37 Zone_38 Zone_39 Zone_4 Zone_40
Most Medium Interest Objects
Zone_211 Zone_213 Zone_217 Zone_94
e
History Search
cycle + bus
a
b
c
d
Retrieve
Data view
Filters
HHI.Cycle
35 - 186
HHI.Bus
37 - 280
Scatter plot
Size : Bus
Z +
□ Show only filtered data
TripMakers_18_24
Income_5000_10000
0
50
100
150
200
250
300
350
400
450
500
10
20
30
40
50
60
70
80
90
100

Time	Visualization	Data	ViewedObjects	SelectedObjects
2009-05-08 15:35:28	●			
2009-05-08 15:35:21	●	●		
r 2009-05-08 15:35:00	●	● ●		
n 2009-05-07 12:04:13	●			
2009-05-07 11:58:24	●			

Search Results for cycle(57), bus(48) : All (7) | Show Similar Visualizations

Selected : 0(0.0%) , Filtered : 5(2.9%) of 174

Figure 5.8: The orange arrows show the support for overview, search and retrieve. (a) Search interface. (b) Keyword search results. (c) An overview of the occurrence of the search results in the entire time of the metadata view. (d) A summary of the search results. Search results are highlighted in the key aspects overview (e) and in the data view.

The metadata view shows the evolution of the exploration process when the rows are sorted according to the time column. The time cell corresponding to the current visualization state is highlighted in orange. A thumbnail tool tip is shown for each visualization state similar to Figure 5.4a. The metadata view is linked to the history tree representation: when a user selects a visualization state in the metadata view, it is highlighted in the history tree representation. The time cell of that visualization state is furthermore marked with a ’r’ label in blue to represent the start of a new branch. Visualization states between two ’r’ labels represent a thread in the analysis. When a visualization state has a note attached to it, its time cell is marked with a 'n' label in red.

Matched keywords are visualized as colored dots in cells, which indicate that for a certain key aspect a match was found for a certain state (Figure 5.8b). The order and color of the dots in a cell follows the order and color of the keywords in the search interface. An overview of the occurrence of the search results in the entire time of the metadata view is shown at Figure 5.8c. Along with this overview, the order and colors of the circles associated to keywords enable users to quickly scroll though the metadata view to gain an overview of search results and their association to the key exploration aspects. Also they can visually apply logical combinations of keywords. A summary of the keyword search results is shown in Figure 5.8d. It has links to retrieve visualizations containing each keyword or all keywords. In addition the thumbnail tool tip helps them to easily narrow down on the search results without revisiting many visualization states.

When users revisit a visualization state that contains matched keywords, the visualization retrieval loop is closed by highlighting those keywords in the visualizations of that revisited state. If an axis label or a filter widget contains a keyword, it is drawn with a red bounding box; and if a data object matches a keyword, it is encircled in red (see the data view in Figure 5.8).

In the key aspects overview, items matching a keyword are identified using the keyword's color in the search interface (see Figure 5.8e). In addition to the time period and the group of users, the overview is confined to those visualization states that contain matched keywords. Users can add or remove keywords to the search interface by selecting or deselecting items in the key aspects overview respectively. In addition to users iteratively getting an overview of relationships among items in the key aspects overview as in Figure 5.12(a and b), they can retrieve past visualization states from the key aspects overview through the metadata view. This support for the linked processes: overview, search and retrieve as identified in Figure 5.1 is represented using the orange arrows in Figure 5.8.

5.7 Similarity based Search and Retrieval

Searching visualizations via keywords is just one option, another one is to retrieve states that are similar to a given visualization state. To this end, the key aspects of the visualization state are compared with the key aspects of other visualization states from the exploration process. One would like to express the similarity between states in a single number, but this cannot be done straightforwardly, because most of the visualization states are asymmetric to each other. For instance, Figure 5.9 shows two states A and B which are asymmetric to each other in terms of all key aspects - visualization, data and interest. State A has one scatterplot, two filters and four interesting data items. State B has one scatterplot, one barchart, three filters and ten interesting data items. The similar items between two visualization states in each key aspect are highlighted in red (Figure 5.9).

Suppose we want to compare the visualization aspects of A and B. Now, if we view B relative to A, there is a complete match (one of the scatterplots of B is same as the scatterplot of A) and we can rate the similarity between them to be high. However, if we view A relative to B, there is only a partial match (A has only one scatterplot that is same

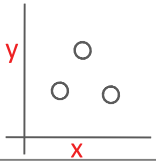
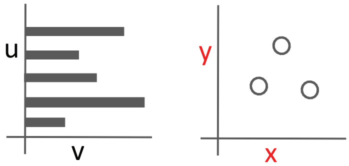
State	Visualizations	Data	Interest
A		Filters M and N	{a, b, c, g}
B		Filters M,N and D	{r, t, a, c, e, k, z, b, s, g}

Figure 5.9: Schematic representation of states A and B. The similar items between states A and B are highlighted in red.

as one of the scatterplots of B, and does not match with other visualizations of B) and the similarity between them is low. This disparity in similarity is also found when comparing other key aspects of A and B (Figure 5.9).

Our reasoning about the comparison of two asymmetric groups is similar to Festinger's theory of social comparison processes [54] on people's abilities: "people tend to compete with those who have similar abilities to themselves and not with those much higher or lower than themselves. These tendencies create a status structure, held in place by both higher and lower groups." If there is a similarity between a person from a lower group and a person from a higher group, then the lower group seeks to show a stronger influence. This is called minority influence [94]. In the above comparison of the visualization aspects of A and B, A is a lower group as it exhibits a minority influence in the comparison. We handle the asymmetry explicitly by comparing both B relative to A (forward comparison) and A relative to B (backward comparison).

For each visualization state, the key aspects are compared with the key aspects of other visualization states. We consider each key aspect of a visualization state as a composite object. Each such object contains a set of object instances. A scatterplot, a barchart, a treemap and other visualization methods are examples of *visualization* aspect objects. A range filter, a nominal filter and a cluster of attributes are some examples of *data* aspect objects. A set of primary keys of a dataset is an example of an *interest* aspect object. Each object has a type and a set of properties.

Now, two objects A and B can be compared as follows. Suppose, object A has a set of object instances A_i where each A_i has a type $T(A_i)$ and a set of properties A_{ip}. Object B has a set of object instances B_j where each B_j has a type $T(B_j)$ and a set of properties B_{jq}. The forward comparison S(A,B) is computed by comparing the properties of the set of object instances of the object B (B_{jq}) relative to the properties of the set object instances of the object A (A_{ip}), as below

$$\begin{aligned}
S(A,B) &= \sum_{i}^{|A|} S_i/|A| \\
\text{with } S_i &= \max_{i}^{|B|} S_{ij}(A_i, B_j) \\
\text{and } S_{ij} &= 0,\ if\ T(A_i)\ \neq T(B_j) \\
S_{ij} &= |\{A_{ik}|A_{ik} = B_{jk}\}|/|A_{ik}|, k = 1, \ldots, M(A_i),\ \text{otherwise.}
\end{aligned}$$

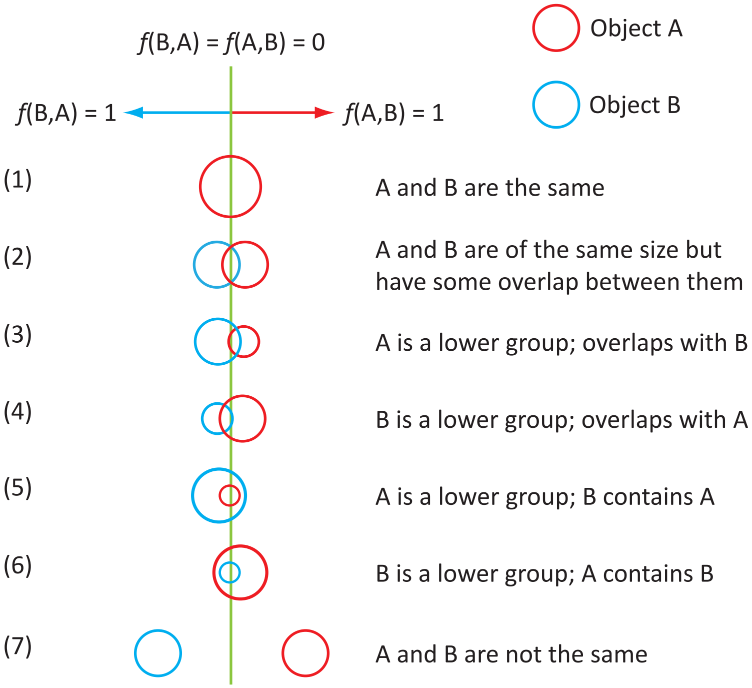
where $S(A, B)$ is the forward comparison between A and B; S_i is the forward comparison of object i of A against objects of B; S_{ij} is the forward comparison of object i of A against object j of B; and $M(A_i)$ is the number of properties of A_i.

Similarly, the backward comparison $S(B, A)$ is computed by comparing the object A relative to the object B. We use a Venn diagram to visualize the comparison between the two objects A and B. The object A is represented in red and the object B is represented in blue. The sizes of the two objects A and B are $|A_{ip}|$ and $|B_{jq}|$ respectively. Object A is placed on one side of a vertical axis; object B is placed on the other side of the vertical axis. In Figure 5.10, the vertical axis is in green; object A is placed on the right side of the vertical axis; and object B is placed on the left side of the horizontal axis. The distance of object A from the vertical axis $f(A, B)$ is $1 - S(A, B)$. The distance of object B from the vertical axis $f(B, A)$ is $1 - S(B, A)$. Figure 5.10 shows all possible results of the two objects comparison and their interpretation using the Venn diagram.

5.7.1 Similarity Search Results in the Metadata View

One obvious way of visualizing the similarity search results is the ranked list view. However, users have to be aware of why a visualization state is similar without having to revisit that state. We visualize the similarity search results in the metadata view which offers the advantages of the ranked list view, and has space to show users the reason behind the ranked list of similar visualization states using the Venn diagrams.

For each key aspect, the current visualization state is considered as the object A and is compared against every other visualization states' key aspect. The metadata view shows a Venn diagram to represent similarity in each cell of the key aspect columns (Figure 5.11). When objects are completely different, the Venn diagram (Figure 5.10(7)) is not shown. The time cell of the current visualization state is highlighted in orange. A summary of the number of visualization states that are similar to the current visualization state is shown below the metadata view (Figure 5.11(a)). The summary has hyperlinks to sort similar visualization states based on the key aspects' forward comparison values. 'Similar' hyperlink shows similar visualization states sorted by average similarity for all key aspects (Figure 5.11(a)), but also a sorted list for one key aspect can be obtained (Figure 5.11(b)).


Object A
Object B
f(B,A) = *f*(A,B) = 0
f(B,A) = 1
f(A,B) = 1
(1)
A and B are the same
(2)
A and B are of the same size but have some overlap between them
(3)
A is a lower group; overlaps with B
(4)
B is a lower group; overlaps with A
(5)
A is a lower group; B contains A
(6)
B is a lower group; A contains B
(7)
A and B are not the same

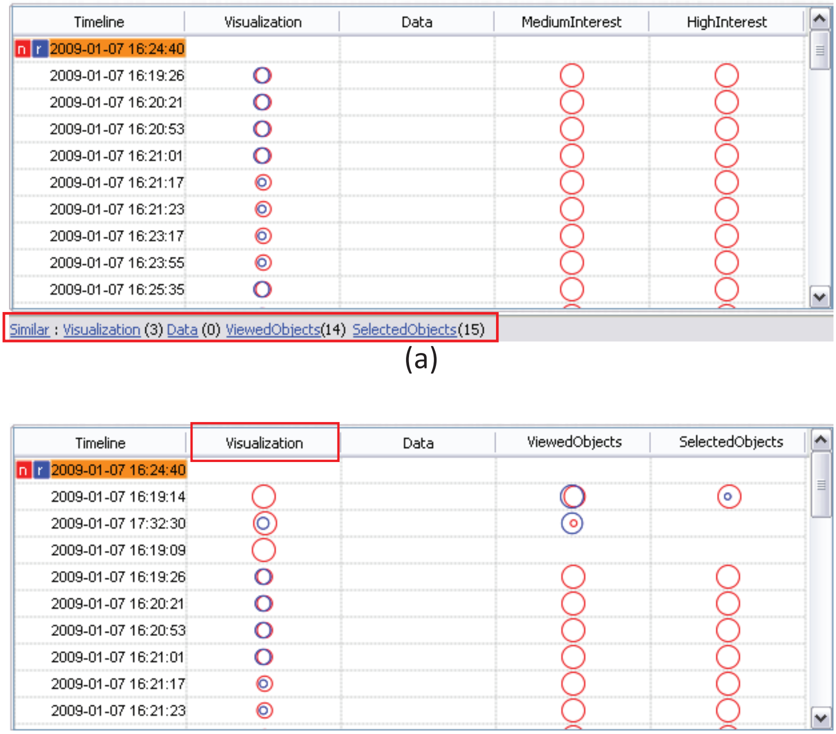
with *f*(A,B) = 1 - S(A,B), and *f*(B,A) = 1 - S(B,A);
where S(A,B) is the forward comparison between A and B, and
S(B,A) is the backward comparison between A and B.

Figure 5.10: Venn diagrams for interpreting similarity between two visualization states.

5.8 Case Studies

We conducted three case studies to assess to what extent the three linked processes: overview, search and retrieve help to develop an awareness of the past visual analysis during asynchronous collaboration.

The first case study considers the asynchronous collaboration between two analysts where the second analyst has to continue a past analysis of the first analyst. Two urban transport researchers participated in the case study. The first analyst works at a university and the second analyst works for a city corporation in India. They often use simple visualization tools such as Microsoft Excel for their analysis. An urban zone level transport dataset is used for the analysis. The dataset contains the zone wise information about public transport usage, private transport usage, number of passengers categorized according to age, income level and household, and demographics for about 250 zones. We explained the features available in Aruvi to the analysts and answered queries regarding those features during the analysis. The first analyst investigated the dataset for thirty minutes using Aruvi. Since the exploration was at the early stage, he did not record any notes in



Timeline	Visualization	Data	MediumInterest	HighInterest
n r 2009-01-07 16:24:40				
2009-01-07 16:19:26	○		○	○
2009-01-07 16:20:21	○		○	○
2009-01-07 16:20:53	○		○	○
2009-01-07 16:21:01	○		○	○
2009-01-07 16:21:17	◎		○	○
2009-01-07 16:21:23	◎		○	○
2009-01-07 16:23:17	◎		○	○
2009-01-07 16:23:55	◎		○	○
2009-01-07 16:25:35	○		○	○

Similar : Visualization (3) Data (0) ViewedObjects(14) SelectedObjects(15)

(a)

Timeline	Visualization	Data	ViewedObjects	SelectedObjects
n r 2009-01-07 16:24:40				
2009-01-07 16:19:14	○		○	◎
2009-01-07 17:32:30	◎		◎	
2009-01-07 16:19:09	○			
2009-01-07 16:19:26	○		○	○
2009-01-07 16:20:21	○		○	○
2009-01-07 16:20:53	○		○	○
2009-01-07 16:21:01	○		○	○
2009-01-07 16:21:17	◎		○	○
2009-01-07 16:21:23	◎		○	○

(b)

Figure 5.11: Metadata view showing similar visualization states sorted by (a) average similarity for all key aspects and (b) visualization aspect similarity. Red circle represents the current visualization state. Blue circles represent the visualization states in each row.

the knowledge view. The analysis was archived to a file and the analysis file was handed over to the second analyst. Figure 5.6 shows the key aspects overview of the analysis of the first analyst.

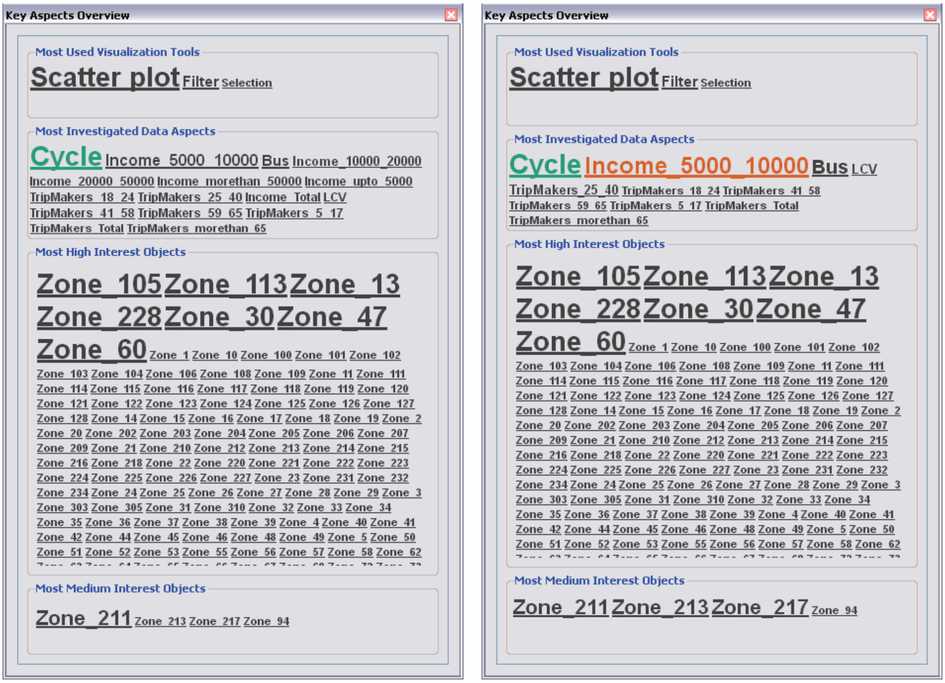
The second analyst was asked to understand the highlights of the analysis using the key aspects overview and search mechanisms. He used mostly the key aspects overview. It became clear to him that the first analyst used only the scatterplot visualization to understand the relationship between attributes. Occasionally, the first analyst applied filters and investigated/identified few interesting zones. Then, he looked at the most investigated data aspects overview. Since 'cycle' and 'bus' are the most prominent items, followed by the income category items and trip makers, he hypothesized that the first analyst tried to understand the relationship among the usage of different modes of transport and the passengers' income levels and age in all zones. To confirm his hypothesis, he selected the 'cycle' item to get an overview of the key aspects when the first analyst investigated the cycle users (see Figure 5.12(a)). Then, he selected the next most frequent item ('Income_5000_10000') in the 'most investigated data aspects' to refine the key aspects overview (see Figure 5.12(b)). From this overview, he identified that the first analyst tried to understand the relationship between 'Income_5000_10000' and different passengers' age categories ('trip_maker_*'). Similarly, he checked for other income levels in Figure 5.12(a). 'Bus' and 'LCV' (Longer Combination Vehicles - long bus) were also prominent next to 'cycle' and 'income' level in most of the cases as in Figure 5.12(b). He continued to iteratively select items and see the relationship among the key aspects. Finally, he concluded that the first analyst tried to understand the relationship between different income levels and age categories of the passenger. Further, he investigated the effect of modes of transport (mainly cycle, bus and LCV) on these relationships. He also understood from interactively refining the overview that the first analyst initially started investigation with a larger number of zones and subsequent narrowed down to seven interesting zones. He was curious why the first analyst narrowed down to these seven interesting zones. Then he continued to investigate the effect of demographics and other modes of transport.

After the session with the second analyst, we asked the first analyst if the second analyst's interpretation of his analysis was right. The first analyst confirmed this. He further explained to us his analysis strategy in detail, and answered a few questions such as why he narrowed down to seven zones and why he investigated only cycle and bus users.

The second case study considers collaboration between two analysts who have worked independently aiming at the same analysis goal, and are sharing their results with each other. The two analysts (one male and one female) that participated in the case study are graduate students in computer science who have had exposure to information visualization tools through academic course work. They investigated a food nutrition dataset for finding a meat replacement food plan. The dataset contains a food group description and the amount of nutrients such as minerals, vitamins, fat, proteins, carbohydrates and energy content for about 1500 food items.

After receiving a training on the features of Aruvi, they explored the dataset for around 45 minutes. Figure 5.13 shows a visualization state and the exploration overview of one of the analysts taken during her analysis. They used scatterplot and barchart visualizations during their analysis. After conducting their individual analysis, they came together and discussed their analysis results. During this collaboration, each user took turns to present his/her analysis to the other collaborator using their notes in the knowledge view. After the presentation, the other collaborator interrogated the analysis. We found that users mostly used the keyword search and the key aspects overview to retrieve key visualization states from the past during the presentation and interrogation sessions. Multiple keywords were used to search for the co-occurrence of some data items. For instance, one user was asked if meat and dairy products had similar calcium content. The other user retrieved a visualization state where he investigated the calcium content of the meat and dairy food products and presented an overview of calcium content in them.

The food dataset contained a variety of similar food items, for instance, different types of cheese. During the analysis session, one of the users used the keyword search to locate all varieties of cheese in the scatterplot; and could see their distribution while studying protein and fat contents of the food items. This keyword search helped her to understand if all cheese products had similar protein and fat contents. At the end of the discussion,


Key Aspects Overview
Most Used Visualization Tools
Scatter plot Filter Selection
Most Investigated Data Aspects
Cycle Income_5000_10000 Bus Income_10000_20000
Income_20000_50000 Income_morethan_50000 Income_upto_5000
TripMakers_18_24 TripMakers_25_40 Income_Total LCV
TripMakers_41_58 TripMakers_59_65 TripMakers_5_17
TripMakers_Total TripMakers_morethan_65
Most High Interest Objects
Zone_105 Zone_113 Zone_13
Zone_228 Zone_30 Zone_47
Zone_60 Zone_1 Zone_10 Zone_100 Zone_101 Zone_102
Zone_103 Zone_104 Zone_106 Zone_108 Zone_109 Zone_11 Zone_111
Zone_114 Zone_115 Zone_116 Zone_117 Zone_118 Zone_119 Zone_120
Zone_121 Zone_122 Zone_123 Zone_124 Zone_125 Zone_126 Zone_127
Zone_128 Zone_14 Zone_15 Zone_16 Zone_17 Zone_18 Zone_19 Zone_2
Zone_20 Zone_202 Zone_203 Zone_204 Zone_205 Zone_206 Zone_207
Zone_209 Zone_21 Zone_210 Zone_212 Zone_213 Zone_214 Zone_215
Zone_216 Zone_218 Zone_22 Zone_220 Zone_221 Zone_222 Zone_223
Zone_224 Zone_225 Zone_226 Zone_227 Zone_23 Zone_231 Zone_232
Zone_234 Zone_24 Zone_25 Zone_26 Zone_27 Zone_28 Zone_29 Zone_3
Zone_303 Zone_305 Zone_31 Zone_310 Zone_32 Zone_33 Zone_34
Zone_35 Zone_36 Zone_37 Zone_38 Zone_39 Zone_4 Zone_40 Zone_41
Zone_42 Zone_44 Zone_45 Zone_46 Zone_48 Zone_49 Zone_5 Zone_50
Zone_51 Zone_52 Zone_53 Zone_55 Zone_56 Zone_57 Zone_58 Zone_62
Most Medium Interest Objects
Zone_211 Zone_213 Zone_217 Zone_94
Key Aspects Overview
Most Used Visualization Tools
Scatter plot Filter Selection
Most Investigated Data Aspects
Cycle Income_5000_10000 Bus LCV
TripMakers_25_40 TripMakers_18_24 TripMakers_41_58
TripMakers_59_65 TripMakers_5_17 TripMakers_Total
TripMakers_morethan_65
Most High Interest Objects
Zone_105 Zone_113 Zone_13
Zone_228 Zone_30 Zone_47
Zone_60 Zone_1 Zone_10 Zone_100 Zone_101 Zone_102
Zone_103 Zone_104 Zone_106 Zone_108 Zone_109 Zone_11 Zone_111
Zone_114 Zone_115 Zone_116 Zone_117 Zone_118 Zone_119 Zone_120
Zone_121 Zone_122 Zone_123 Zone_124 Zone_125 Zone_126 Zone_127
Zone_128 Zone_14 Zone_15 Zone_16 Zone_17 Zone_18 Zone_19 Zone_2
Zone_20 Zone_202 Zone_203 Zone_204 Zone_205 Zone_206 Zone_207
Zone_209 Zone_21 Zone_210 Zone_212 Zone_213 Zone_214 Zone_215
Zone_216 Zone_218 Zone_22 Zone_220 Zone_221 Zone_222 Zone_223
Zone_224 Zone_225 Zone_226 Zone_227 Zone_23 Zone_231 Zone_232
Zone_234 Zone_24 Zone_25 Zone_26 Zone_27 Zone_28 Zone_29 Zone_3
Zone_303 Zone_305 Zone_31 Zone_310 Zone_32 Zone_33 Zone_34
Zone_35 Zone_36 Zone_37 Zone_38 Zone_39 Zone_4 Zone_40 Zone_41
Zone_42 Zone_44 Zone_45 Zone_46 Zone_48 Zone_49 Zone_5 Zone_50
Zone_51 Zone_52 Zone_53 Zone_55 Zone_56 Zone_57 Zone_58 Zone_62
Most Medium Interest Objects
Zone_211 Zone_213 Zone_217 Zone_94

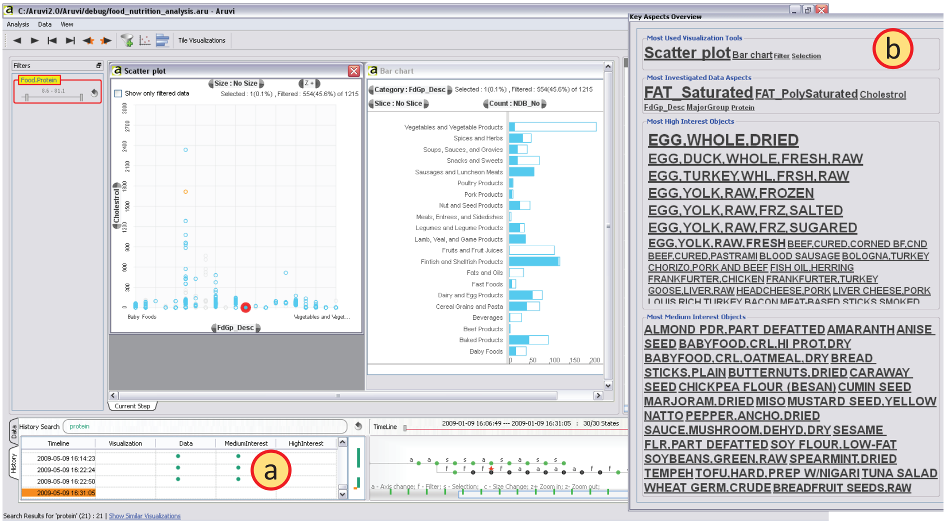
(a) (b)

Figure 5.12: (a) and (b) Key aspects overview is provided with respect to the items iteratively selected by the user.

the users understood each other's analysis strategy. One user was focusing on selecting the food items that have similar characteristics as meat products, while the other user was selecting food items by considering vegetables and dairy products that are rich in vitamins, minerals and protein contents; moderate in energy content and low in fat and cholesterol contents.

The third use case considers continuation of an analysis process after a break. An analyst who is an information visualization researcher reviewed his past analysis. He analyzed the food nutrition dataset (used in the second use case) to find a balanced food plan for anemia. He explored the dataset using the Aruvi visualization system for thirty minutes after receiving a training on its features. During this exploration process, he encountered some interesting facts and patterns that helped him to select some food items. At the end of the analysis, the analyst listed food items that are rich in iron content. During the analysis, he only used a scatterplot visualization. He revisited his analysis the next day. During the review process, he found that the final list did not contain one of his favorite food items, an apple. He was curious to find why he inadvertently left out his favorite food item. Also, he wanted to make sure if the final list also appeared while he was investigating zinc content.

For the first question, the user searched for 'apple' and inspected the keywords trail in



Timeline	Visualization	Data	MediumInterest	HighInterest
2009-05-09 16:14:23		●	●	
2009-05-09 16:22:24		●	●	
2009-05-09 16:22:50		●	●	
2009-05-09 16:31:05				

Search Results for 'protein' (21) : 21 | Show Similar Visualizations

Figure 5.13: Aruvi Prototype. (a) The metadata view enables users to search and retrieve past visualizations. (b) The key aspects overview.

the metadata view. He found that apple was removed from the investigation when he used iron and copper filters. He became aware of the fact that apple had low iron and copper content as he always thought apples were good for anemia. For the second question, he searched for 'zinc' and selected show similarity in the metadata view. Based on the Venn diagram representation, he could only judge a partial match. However, he was expecting a matching list of food items to be presented.

5.8.1 Limitations

The case studies mainly focused on extracting analysis strategies used by analysts in past analyses which can be one of the many reasons why analysts want to review a past analysis. We developed three linked processes: overview, search and retrieve to enable users to develop exploration awareness on a past visual data analysis. The tasks required for extracting the analysis strategy did not require all the three processes. Analysts only used the key aspects overview for extracting the analysis strategy. The 'most used visualization tools' and the 'most investigated data aspects' helped analysts to understand the analysis strategy. The 'most selected object' showed the focus of analysts on particular data items that are a part of their analysis strategy.

We could gather only limited support for the search and retrieve processes due to the analysis goals which only required analysts seek an overview of the analysis process. In the second case study, one of the analysts used the keyword search during the analysis process to understand the distribution of items matching keywords in the scatterplot. The similarity search was used in only one instance and we assume it will be used only in complex analysis processes. Overall, we found that the key aspects overview was the dominant feature that helped the analysts to understand their collaborators' analysis strategies. However, we could not gather much support for the three linked processes: overview, search and retrieve to develop exploration awareness on a visual data analysis due to the limitation of the case study design. In future, we will study the three linked process in detail.

5.9 Conclusion

In this chapter, we presented three linked processes: overview, search and retrieve to support developing exploration awareness during a review process. For this, we first presented a user's information interest model that captures key exploration aspects such as visualization and data transformations, data aspects in those transformations, viewed (medium interest) and selected (high interest) objects. An exploration overview is provided using a key aspects overview and a history representation. A keyword based visualization retrieval mechanism was discussed. The metadata view is used to visualize the search results and their association to the key aspects. The visualization retrieval loop is closed by highlighting the keywords in the visualizations once a visualization state that contains matched keywords is revisited. Furthermore, a similarity based visualization retrieval mechanism that retrieves visualization states based on the content similarity to the current visualization state was discussed. Three case studies were discussed. These case studies revealed the support offered by the framework to develop exploration awareness during asynchronous collaboration.

When analysts interactively explore complex datasets over multiple sessions, they may uncover a large number of findings. They must often connect findings discovered at various points of time for effective reasoning process. During long analysis sessions, they may not notice all implicit connections between these findings as it is often difficult for them to recall the past findings, views and concepts that are most relevant to their current line of inquiry. In this chapter, we presented tools that help users to develop exploration awareness for reviewing purposes by proactively searching on the key aspects of a visual data analysis. In the next chapter, we describe our approach to support connection discovery using a related notes, visualization and concepts recommendation system based on a context based retrieval mechanism. In this way, we aim at supporting automated connection discovery among findings, visualizations and concepts investigated during a visual data analysis.

Chapter 6

Connection Discovery

> *You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future.* — Steve Jobs

During visual analysis, users must often connect insights discovered at various points of time. This process is often called *"connecting the dots."* When analysts interactively explore complex datasets over multiple sessions, they may uncover a large number of findings. As a result, it is often difficult for them to recall the past insights, views and concepts that are most relevant to their current line of inquiry. This challenge is even more difficult during collaborative analysis tasks where they need to find connections between their own discoveries and insights found by others. In this chapter, we describe a context-based retrieval algorithm to identify notes, views and concepts from users' past analyses that are most relevant to a view or a note based on their line of inquiry. We then describe a related notes recommendation feature that surfaces the most relevant items to the user as they work based on this algorithm, and conclude with a case study.

6.1 Introduction

Interactive visualizations allow users to investigate various characteristics of a dataset and to reason based on patterns, trends and outliers. During complex visual analyses, users must derive insights by connecting discoveries made at different stages of an investigation. However, during a long investigation process that can span hours, days or even weeks, it becomes difficult for users to recall the details of their past discoveries. Yet these details may form the key connections between their past work and current line of inquiry. We believe that the difficulty in recalling past work often leads users to overlook important connections. The challenge, therefore, is to develop techniques that assist in *connection discovery* by uncovering connections to users' past work that would normally go unnoticed.

To address the challenge of recalling past work, users often externalize interesting findings or new hypotheses using either annotations on top of visualizations or through bookmarks in electronic notes. These notes help users to manually revisit and review their past analysis. However, as the number of notes and annotations grows larger, users again have difficulty recalling the details of each previous discovery. Therefore, users must be enabled to more easily retrieve *views* (visualization states with one or more visualizations), *notes* and *concepts* (including data characteristics investigated in the views and entities from notes) from their past analyses. These related views, notes and concepts can then help them to find interesting connections within their analysis.

In this chapter, we describe a context-based retrieval algorithm that retrieves views, notes and concepts from a user's past analysis related to a view or a note based on their line of inquiry. Whenever users create a view or record a note, we derive a context description for the view or note from their line of inquiry. Our algorithm then uses these context descriptions to retrieve the most relevant views, notes and concepts from past analyses.

Using our context-based retrieval algorithm, we have implemented a related notes recommendation feature in HARVEST, a web based visual analytic system. As users create new views during their analysis, HARVEST dynamically applies our algorithm to recommend the most relevant notes from past analyses. An overview of related notes is presented as a ranked list of notes along with a thumbnail of associated views in the note-taking interface. An overview of related concepts is also shown using a tag cloud. Both overviews are updated after each exploration action. We evaluate the related notes recommendation feature of HARVEST through a case study and discuss the implications of our approach. Specifically, we believe that the related notes recommendation feature helps users to maintain greater awareness of relevant information and assists in connection discovery during visual analysis.

6.2 Connection Discovery

We encounter a lot of information during daily activities. We process that information to learn new things, perform tasks or make decisions, and store that processed information in our memory. However, our memory is limited in its ability to store and recall relevant information from the past [38]. To overcome these limitations, we have learnt to work around by taking notes, capturing pictures and videos, or associating with a local environment [68]. In addition, we also create to-do lists and automatic reminders using personal information management systems [98]. These external attention pointers help us remember information that would otherwise be forgotten. Thus, we try to connect the dots using these attention pointers and make sense of information encountered in our daily activities.

Also, when we read a text, we process information from it to understand the story conveyed by its authors. For this, we need to connect the dots at various parts of the text and make sense of it. A good text provides relevant attention pointers in the text that help a reader to connect the dots. For example, authors of academic text use cross-referencing as a reminder that helps readers to locate relevant pieces of information from other locations. Similarly, authors of fiction text use sequences of events or people and context descriptions as attention pointers that help readers to connect the dots.

During a visual analysis, analysts encounter much information by interactively exploring large datasets using visualizations. They also formulate some interesting findings during this exploration process. Due to the volume of information discovered during a long analysis task, they often externalize interesting findings or new hypotheses using either annotation on top of visualizations or through bookmarks in electronic notes. They organize these findings into a case and present them to others [67, 117]. They must often connect insights discovered at various points of time and make sense of them [65]. However, during a long investigation process that can span hours, days or even weeks, it becomes difficult for users to recall the details of their past discoveries. Therefore, it is difficult to connect the dots during a visual analysis. Hence, we think it will be helpful for the users to retrieve notes, views and concepts that are related to a given view or note based on their line of inquiry. Also, during a visual analysis, the most relevant items from past analyses related to their current line of inquiry can be recommended for maintaining awareness of relevant information and to assist in connection discovery.

6.3 Related Work

First, we present a number of sense making models that highlight the critical role of connection discovery during information analysis. We then discuss work related to connection discovery during visual analysis.

6.3.1 Sense Making Models

Kuhlthau [83] considers a sense making process as an information search process in which a person is forming a personal point of view [45]. She identifies six stages in an information search process from a user's perspective: initiation, selection, exploration, formulation, collection, and presentation. She modeled the cognitive, affective and action aspects involved in these six stages by conducting longitudinal user studies involving various public library users, students and academic researchers. Finding relevant information to the current topic is one of the important actions during the exploration and collection stages. These actions help to avoid premature closure of an information search process.

Similarly, Ellis [48] classifies information seeking activities into eight categories: starting, chaining, browsing, differentiating, monitoring, extracting, verifying, and ending. She models the process of connection discovery in the information search process in two categories: chaining and monitoring. Chaining involves following a referential connection between information sources. Monitoring involves maintaining awareness by tracking related information sources.

Pirolli and Card [103] identify two major loops in the sense making process during an intelligence analysis task: the information foraging loop and the sense making loop. They found that analysts look back into the processed information obtained (evidence file) during the information foraging loop from the sense making loop to search for evidence or relations that support a hypothesis. If no supporting information is found, analysts continue to forage new information.

6.3.2 Visual Analysis

In general, to support the sensemaking process in visual analytics (discussed in chapter 3) [117], users can be provided with three types of linked views: a data view, a knowledge view and a navigation view. The data view has interactive visualization tools; the navigation view provides an overview of the exploration process, for instance, a history tree and action trails; and the knowledge view helps to record and organize notes. Currently, during an analysis, the connection discovery process is supported by exploiting the relationships shared between either views and notes, or entities in notes.

Using Links between Views and Notes

Several information visualization tools support links between views and notes. In Aruvi, users can externalize findings using notes along with links to the views. They can revisit views via notes and review and revise their analysis. To support the review process, it also provides an overview of key visualization and data aspects in an exploration process using a user's information interest model (discussed in chapter 5). They can also retrieve visualizations from the past analysis using keyword and similarity search mechanisms (Chapter 5).

Sense.us [67], a web site supporting asynchronous collaboration across a variety of visualization types, supports view sharing, discussion, graphical annotation, and social navigation. It has a doubly-linked discussion mechanism that supports situated conversation about visualizations. For this, both data and view parameters of visualization states are indexed and associated with the corresponding comments. Thus, during an asynchronous collaboration, all comments associated with a view are retrieved.

Using Entities

A combination of text analytics and information visualization has been widely used to analyze massive textual data. Text analytics is used to extract entities from the text and the relationship between those entities is visualized. The Have Green framework [138] uses an interactive graph visualization to represent concepts and relationships extracted through its analytical capabilities. In Jigsaw [119], multiple coordinated views are used to visualize the connections between entities extracted from a collection of text documents. A graph view is used to visualize text documents and entities shared among these documents. In addition to graph visualization, a list view is used to show the connection between entities. A scatterplot view is used to explore pairwise connections between entities. However, in Have Green and Jigsaw text analysis is used on the input data, but not applied to a user's notes.

Analyst's Notebook [70] visualizes the relationships among entities extracted from a user's notes using graph visualization. In Entity Workspace [24], users can record notes or place text snippets, entities and their relationship from notes and documents are extracted and a document-entity graph is constructed. Using this graph model, analysts can re-find facts quickly, notice connections between entities, abstract information structure and identify documents and entities to explore further. During a collaborative analysis, the most valuable notes from other analysts related to the current topic (text) are recommended to an analyst using an entity graph. Thus the entity workspace identifies related entities and helps analysts to connect the dots while investigating a text document corpus. Also, in InsightFinder [33], users' notes are used to build a context model. Using this context model, the most relevant page units are recommended to them while browsing the Internet.

During a visual analysis, users formulate findings after some exploration as identified in Pirolli and Card's sensemaking model and Kuhlthau's information seeking process model. For connection discovery in visual analysis, approaches based on links between views and notes or entities in notes are not sufficient. The users' line of inquiry has to be considered in combination with view and data parameters of views and entities in notes. We now present our approach to connect the dots in visual analysis, by considering the users' line of inquiry, view and data parameters of views, and entities in notes in an integrated way.

6.4 Approach

To support the connection discovery process during a visual analysis, we enable users to retrieve views, notes and concepts from past analyses related to a view or note. Figure 6.1 shows our approach. Whenever they create a view of their data (in the data view) or record a note (in the knowledge view), we derive a context description for the view or note from their line of inquiry. Our algorithm then uses these context descriptions to retrieve the most relevant views and notes from past analyses. The context description is derived from a model of visual analytic activity called *action trails* [60]. Action trails represent users' analytic activity as graphs of semantic analytic steps, or actions. Actions can be classified into broad categories: *exploration actions*, *insight actions*, and *meta-actions*. An exploration action alters the visualization specifications in a visual analytics system and creates a new view. Insight actions record or organize notes and views, while meta-actions (e.g., revisit, undo, redo) allow users to review and structure their lines of inquiry.

Action trails contain valuable information about the concepts that are most relevant to a user's analysis and how the user's interests evolve over time. We therefore extract a set of concepts from the action trail to form the context description for each view or note. We extract two types of concepts. Action concepts are derived from the attributes associated with exploration actions (e.g., data and view parameters). Entities are concepts extracted from a user's notes and represent items such as people, places or companies. For each concept associated with a view or note, we derive *concept weights* from the user's action trail to determine its degree of salience at the time the view or note was created.

For a view or note focused on by the user, we compute the relevance score to existing views and notes by comparing the context descriptions of existing views and notes with that of the given view or note. Using the relevance score, the related views and notes are retrieved. An overview of the related concepts is also provided. Thus, this context-based retrieval algorithm surfaces the most relevant information from the past analyses of the users based on their line of inquiry during a visual analysis.

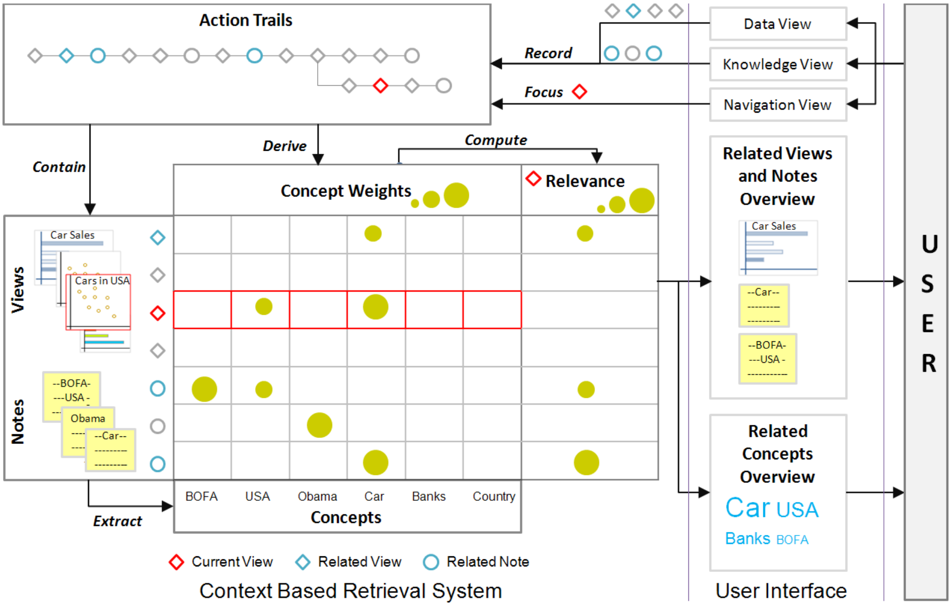
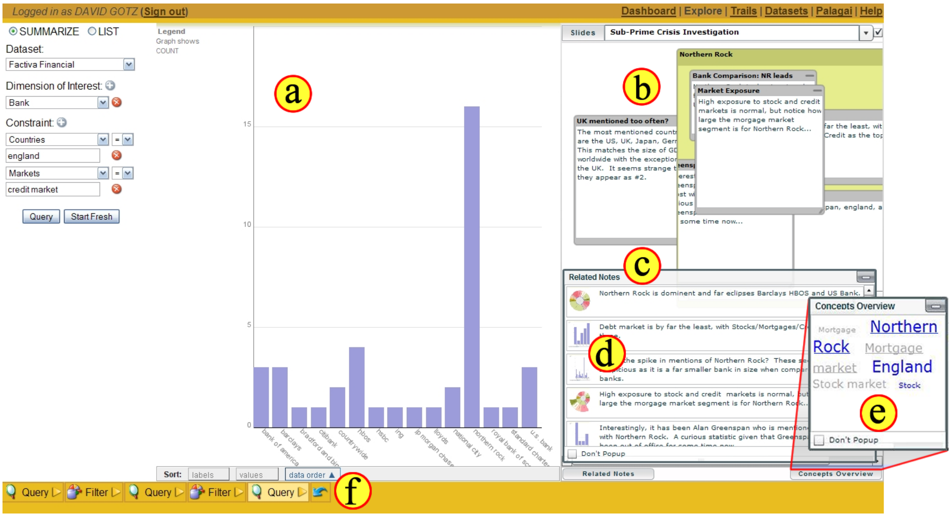


Figure 6.1: A context-based retrieval system that retrieves related notes, views and concepts for a view or a note based on the users' line of inquiry. This retrieval system is used to support the connection discovery process during a visual analysis.

Using this context-based retrieval algorithm, we have implemented a recommendation feature in HARVEST, a web based visual analytics system which is shown in Figure 6.2. The recommendation feature shows a list of related notes (Figure 6.2(c)) along with thumbnails of the view displayed while recording those related notes (Figure 6.2(d)) to the current view (Figure 6.2(a)). Also, it provides an overview of related concepts using a tag cloud (Figure 6.2(e)). In the following sections, we describe the context-based retrieval algorithm (Section 6.5) and present the design considerations (Section 6.6) and implementation details (Section 6.7) of the recommendation feature in HARVEST.

6.5 Context-based Retrieval Algorithm

In this section, we describe the details of our context-based retrieval algorithm. First, we present a visual analysis use case. Next, we support our argument for a context description based on action concepts and entities from action trails with the use case. We then use the context description as the basis for the relevance metric used to identify related views, notes, and concepts.



High exposure to stock and credit markets is normal, but notice how large the morgage market segment is for Northern Rock...

The most mentioned countr are the US, UK, Japan, Gerr This matches the size of GD worldwide with the exceptior the UK. It seems strange t they appear as #2.

far the least, wit Credit as the top

pan, england, a

some time now...

Northern Rock is dominent and far eclipses Barclays HBOS and US Bank.

Debt market is by far the least, with Stocks/Mortgages/C these.

the spike in mentions of Northern Rock? These se picious as it is a far smaller bank in size when compar banks.

High exposure to stock and credit markets is normal, but large the morgage market segment is for Northern Rock...

Interestingly, it has been Alan Greenspan who is mention with Northern Rock. A curious statistic given that Greenspa been out of office for some time now.

Figure 6.2: A user investigating a finance dataset in HARVEST, a web based visual analytics system. (a) The data view shows a visualization created by the steps shown in the user's action trail (f). (b) A note-taking interface. (c) A ranked list of related notes. (d) Thumbnail of the view displayed while recording those related notes. (e) Related concepts overview - An overview of related entities from notes (underlined) and related action concepts from action trails.

6.5.1 Use Case

Figure 6.3 shows a portion of an action trail for an analyst investigating product sales data. She starts her analysis by focusing on sales that are more than $50,000 (Figure 6.3(1)). She compares sales of each product using a scatterplot visualization and bookmarks it (Figure 6.3(2)). Then, she studies quarterly sales of the products by aggregating the sales represented on the y-axis of the scatterplot based on a quarterly time period (Figure 6.3(3)). Next, she uses a tree map to visualize the sale figures in various regions (Figure 6.3(4)). Further, she clusters the products by their category to get an overview of the sales performance by product category in various regions (Figure 6.3(5)). This view triggers her to reconsider the products sales comparison that she investigated some time back. She therefore revisits the comparison view she bookmarked earlier. Then she narrows down to the east and south regions (Figure 6.3(6)). This revisit and reuse of a view creates a branch in her action trail.

She further slices the products in the x-axis of the scatterplot by their category; and slices sales in the y-axis of the scatterplot by quarterly period (Figure 6.3(7)). This slicing creates a scatterplot matrix showing sales of various product categories in different quarters of the year. She finds out that product categories A, C and D have shown profit consistently in the east and south regions. She records this finding using a note. Then, she continues her analysis by studying yearly sales (Figure 6.3(8)) and sales distribution across regions using a map (Figure 6.3(9)).

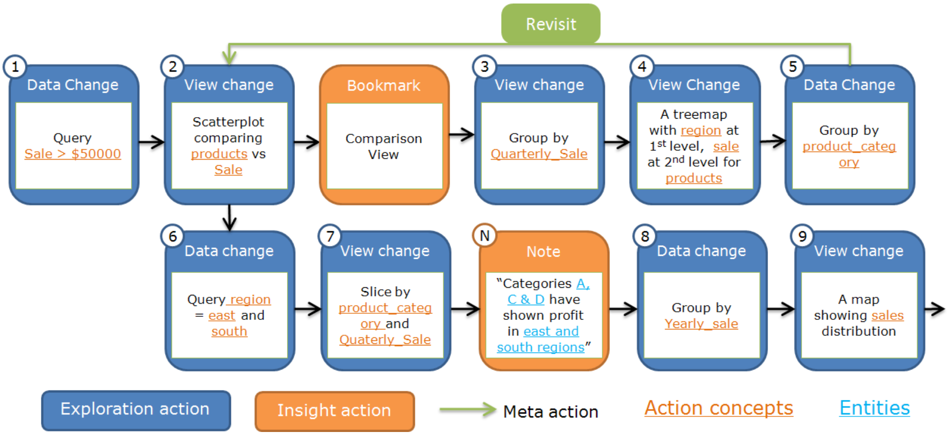

Revisit
1 Data Change
Query Sale > $50000
2 View change
Scatterplot comparing products vs Sale
Bookmark
Comparison View
3 View change
Group by Quarterly_Sale
4 View Change
A treemap with region at 1st level, sale at 2nd level for products
5 Data Change
Group by product_category
6 Data change
Query region = east and south
7 View change
Slice by product_category and Quaterly_Sale
N Note
"Categories A, C & D have shown profit in east and south regions"
8 Data change
Group by Yearly_sale
9 View change
A map showing sales distribution
Exploration action
Insight action
Meta action
Action concepts
Entities

Figure 6.3: Part of an action trail for an analyst investigating product sales. Exploration actions are represented with a blue box; insight actions such as bookmarking and note-taking are represented using an orange box; meta-actions such as revisit are represented using a green line with an arrow.

6.5.2 Action Concepts as Context

In the products sales use case, the user started her analysis with general sales data and moved on to investigate quarterly and yearly sales trends. Region was another aspect considered in the investigation; she focused on all regions, then narrowed down to the east and south regions, and finally moved on to see the actual geographical sales distribution. She also investigated the sales of individual products as well as product categories (groups of products).

The action concepts associated with this action trail (e.g., the east region and product category) correspond to the user's information interests. However, some of the action concepts were more predominant at certain times than others. For instance, she was interested only in sales of more than $50,000 throughout the investigation. In contrast, she shifted her focus among other action concepts such as quarterly sales, product categories, and regions. Her interest in these action concepts varied over time. Therefore, during an exploration process, users' evolving information interests can be viewed as a time-varying set of weighted action concepts taken from their action trails.

A set of weighted action concepts is associated with each view and note to represent its context description. The weight for each action concept represents its degree of salience at the time the view or note was created. The metrics used for calculating the weight from the action trails are motivated by the spreading-activation construct that is used in many theories for retrieving information from long term memory [16, 37]. In these theories, knowledge is encoded as a network structure, consisting of nodes representing concepts and links representing associations among concepts. During a retrieval process, this network structure is used to identify knowledge relevant to a current focus of attention and facilitate processing of associated items. The two basic points emphasized in these theories are (1) activation is modeled as a spreading function, and (2) activation decays exponentially with the distance it spreads over a network structure [37].

Back Trace and Forward trace

Action trails represent a network structure consisting of views and notes. This network structure holds concepts and their relationships investigated during an analysis. To extract related action concepts for a view or a note, we use a spread function from the view or note over the network structure presented in the action trails. A trace spreads through the network structure of an action trail to reflect that a view or note can be created by a confluence of different lines of inquiry. Figure 6.4(a) shows a back trace of exploration actions for a view using the structure of the analyst's action trail shown in Figure 6.3.

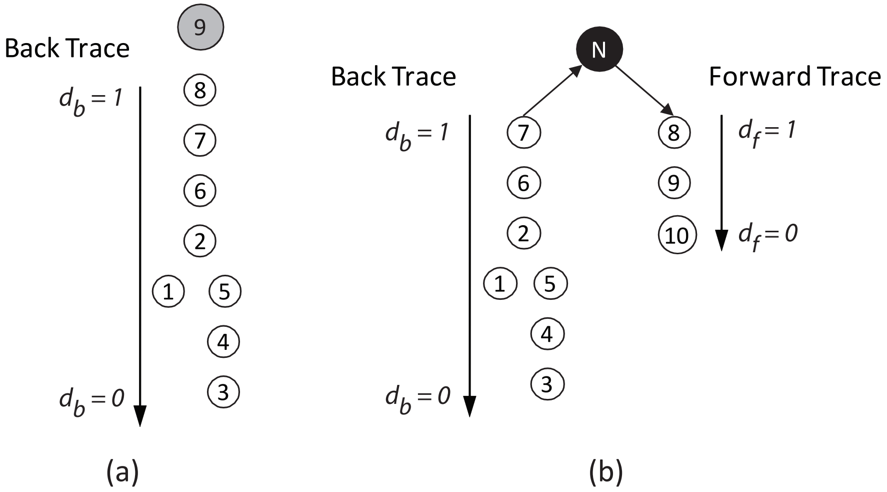


Figure 6.4: (a) Back trace of exploration actions for view 9 in Figure 6.3. (b) Back trace and forward trace of exploration actions for the note N in Figure 6.3. d_b and d_f are the normalized weight for each exploration action in the back trace and forward trace respectively.

We can trace both forward and backward through the action trail for a note. To determine if a backward or forward trace is appropriate, we determine the type of insight behavior being performed by the user. Based on our observation of how users record notes in Aruvi [117], we distinguish six categories of notes taking.

Finding. Findings are usually obtained after a sequence of exploration actions. Hence, a back trace of exploration actions will give related action concepts for this note. A note with a link to a view is categorized as a finding.

Hypothesis. Users record some assertions or hypotheses that they want to confirm during an investigation. These notes influence subsequent actions. Hence, a forward trace of the exploration actions will give related action concepts for this note. A note without a link to a view is categorized as an hypothesis. Users can also formulate hypotheses after exploring data. In this case, we assume users record such hypotheses as findings with a link to a view.

Snippet. Users can collect some relevant information from outside a visual analytics system (e.g., a snippet from the Internet). In this case, either a sequence of exploration actions might have triggered them to look for some external information or they may be preparing for an investigation by gathering some external information. Hence, in this case, both back trace and forward trace is required to derive related action concepts (Figure 6.4(b)). A note created by copying contents from the Internet or other digital documents, and without a link to a view is categorized as a snippet.

Edit. During the exploration process, users can edit a previously recorded note. In this case, we combine the related action concepts from the previous line of inquiry associated with the note and the related action concepts from the current line of inquiry. Currently, we consider only edits that add a new entity or new sentence to the notes.

Reassociation. Sometimes, users can remove a link between a note and a visualization and reassociate the note to a new visualization. In this case, the related action concepts from the previous line of inquiry are replaced with those from the current line of inquiry.

Multiple Association. Some users requested multiple visualizations created at different instance during an analysis to be associated with a note. In this case, the related action concepts from the line of inquires of each visualization are combined.

In addition to choosing the trace direction, we must also determine how far to trace along the trail. The boundary of a trace is difficult to determine algorithmically from an action trail because it depends on the semantics, and is subjective. So far we apply a simple threshold to determine the boundary: either until n unique action concepts are extracted, or when the start or end of an action trail is reached. After experimenting with various values, we use a threshold of $n = 10$ in our current prototype. Thus, the outcome of the trace is a list of related action concepts from the local neighborhood of the action trails.

Related Action Concept Weight

We derive weights for a set of related action concepts extracted by tracing the action trail based on the following factors:

- **Recency**
  Proximity of an exploration action to a view or a note in an action trail is used to weigh an action concept. In Figure 6.4, d_b and d_f are the normalized weight for each exploration action in the back trace and forward trace respectively based on

the length of the trace. This normalization compensates for the variation in length for each trace.

- **Specificity**
  During an exploration process, analysts may be focused on all values of an attribute (e.g., sales in all regions) or they may focus on specific values of those attributes (e.g., sales in the east and south regions). Hence, if an action concept references specific values within the dataset, then it is given more weight than those which reference generic characteristics. In our current prototype, a specific concept is given a specificity weight s_{c} that is twice the weight of a generic concept (e.g., all regions).

Based on the factors above, the weight W_{c} for an action concept c is as follows

$$W_{\mathrm{c}} = s_{\mathrm{c}} \times \left(w_{\mathrm{b}} \times \sum_{i=1}^{b} d_{\mathrm{i}} + w_{\mathrm{f}} \times \sum_{i=1}^{f} d_{\mathrm{i}} \right),$$

where s_{c} is the specificity weight of the action concept c; b and f are the length of the back and forward traces respectively; d_{i} is the normalized weight based on recency of an exploration action for back trace or forward trace; (with $d_{\mathrm{i}} = 0$, if c is not specified in an exploration action); w_{b} and w_{f} are the weights for back and forward traces respectively; (with $w_{\mathrm{f}} = 0$, for a view or a finding; $w_{\mathrm{b}} = 0$, for a hypothesis). For each note, related action concepts are extracted and a weight for each action concept is computed based on the structure of the user's action trail. As the exploration process evolves, the set of related action concepts for each note and their weights are updated based on the above categories.

6.5.3 Related Entities from Notes

In the above use case, the analyst recorded a note (in Figure 6.3) that contains entities such as product categories (A, C and D) and regions (east and south) and relationships among them. These entities and relationships also represent her information interest at the time of recording that note in addition to the action concepts that lead to this note. Thus, entities extracted from notes also represent a user's information interest in addition to the related action concepts.

We use text analysis tools to extract entities (e.g., people, places, and organizations) from the user's notes [53]. Often, these entities are of the same types found in the dataset being visualized. An extracted entity has three properties: a type, the covered text and its canonical form. For example, a user might type 'BOFA' in a note to refer to 'Bank of America'. The text analysis tool would detect this phrase as an entity of type 'Bank' with covered text 'BOFA' and canonical form 'Bank of America'. For each type, we also defined a generic canonical form (e.g., 'Generic Bank') to capture general references (e.g., 'Bank' or 'Lender').

A weight can be associated with each entity extracted from a note based on its properties and frequency of occurrence (n) within the note . We associate a weight (w_{e}) to the covered text e: $w_{\mathrm{e}} = n$, if e is a canonical form; $w_{\mathrm{e}} = 0.5n$, if e is a type; and $w_{\mathrm{e}} = 0.25n$, if e is a generic canonical form.

6.5.4 Retrieving Related Views, Notes and Concepts

A view or a note has a context description based on the related action concepts (c) from the action trails and entities (e) extracted from notes. For a given view or a note (B), we can compute a relevance score $d(T)$ for a target view or a note from past analyses (T) as follows

$$d(T) = \sum_{i=1}^{m} (W_B(c_i) \times W_T(c_i)) + \sum_{i=1}^{p} (w_B(e_i) \times w_T(e_i)),$$

where m is the number of related action concepts for the base view or note and p is the number of entities from the base note; with p = 0, if B is a view; $W_T(c_i) = 0$, when c_i is not a related action concept for the target view or note (T); and $w_T(e_i) = 0$, when e_i is not an entity of a target note or the note attached to a target view T. Thus, a ranked list of related views and notes for a given view or note is obtained based on the context descriptions extracted from the action trails.

Next, we derive the related concepts for B. An overview of the related concepts is provided using a tag cloud as shown in Figure 6.2(e). The weights of the action concepts from the context description of B are used to determine the font height for displaying each action concept in the tag cloud. The weight $W(e_i)$ for a entity e_i is computed as

$$W(e_i) = \sum_{k=1}^{n} d(T_k),$$

where n is the number of relevant notes. $d(T_k) = 0$, when the note T_k does not contain the entity e_i.

The weights of the action concepts and entities are normalized before they are used to determine the font height. Entities are underlined while action concepts are not underlined. Since concepts can be represented in multiple words, an alternate coloring scheme is used to distinguish concepts in the tag clouds.

In the above use case, when the analyst explores the geographic distribution of the sales (Figure 6.3(9)), we can retrieve related views and notes from her past analysis. Previously, she investigated sales in all regions using a tree map (Figure 6.3(4)). This view may be one of the most relevant views for her investigation on the geographic distribution of the sales. Using the above context-based retrieval algorithm, we retrieve such related views and notes for a given view or note.

6.6 Recommending Relevant Information

Our algorithm can be used to recommend related views, notes and concepts based on a user's ongoing exploration process. This recommendation can help the user by showing them information they may have overlooked. However, it is also critical to avoid overwhelming the user with too many recommendations. To avoid this, we must automatically recommend only the most relevant information to balance the cost of distracting their attention.

Of the three components—views, notes and concepts—we argue that notes play the most critical role in connection discovery during a visual analysis by acting as a reminder that helps to recall key aspects such as views and concepts during the foraging process (Figure 6.5).

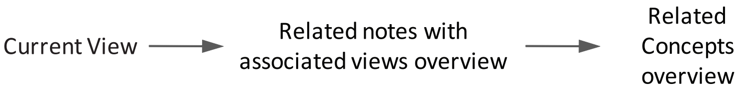

Current View
Related notes with associated views overview
Related Concepts overview

Figure 6.5: Relevant information for connection discovery during the information foraging process in a visual analysis.

To validate this argument, we interviewed two business analysts who do some visual analysis using simple visualization tools such as Microsoft Excel. Both analysts take notes during the analysis process and refer back to it throughout the analysis, when preparing a report, while sharing analysis with others or when starting a new related analysis. The first analyst stated "I take notes to help me remember what I have learnt . . . I would refer to the notes to figure out what I think and what I do. The notes help me remember how I performed a task during the analytic process, for instance, how I derive this insight, how I generate this chart."

The second analyst explained that she records how she manipulated a dataset along with findings in her notebook. She documents in detail especially when she has to create a report for transferring operations to other analysts. She maintains a big notebook and organizes notes with titles that summarize them. While recording any new findings, she tries to locate earlier notes that are most relevant to the particular topic and just add new findings into the old notes. When she creates the detailed report, it is pretty much like starting a new task from her, because usually she forgets what and how she did the analysis. She says "but I have my initial report to help me remember. It is not easy to remember how I did by just looking at the visualizations in the report. I need to click on a few (spreadsheet) cells to remember what it is about."

For the two analysts, the notes acts as a bridge between the analysis executed in the system and their cognitive process. The notes act as reminders to key aspects of the exploration process, such as views or concepts. Hence, in our current prototype, we recommend only related notes along with a thumbnail of the visualizations that led to the formulation of those notes during the exploration process. Figure 6.2 shows recommendations of related notes for the current view (Figure 6.2(a)) based on the user's current line of inquiry. If the users are interested in locating views and concepts with similar context description, they can explicitly request that information.

6.7 Connection Discovery in HARVEST

We have added our recommendation algorithm to HARVEST [59], a web based visual analytics system that supports exploration of large unstructured datasets. It has an action tracking mechanism that automatically captures and displays (Figure 6.2(f)) user's analysis behavior as an action trail [60]. Using the action trail interface, users can archive their trails, as well as revisit and reuse past views. In addition, we extended HARVEST by adding a new note-taking interface that allows users to record notes and organize them into groups and slides (Figure 6.2(b)).

Related notes are surfaced through the note-taking interface. When a user records a note, the system augments it with a context description. Then, as the user creates a new view in HARVEST, the recommendation algorithm dynamically derives a context description for the view from the current action trail, and compares it with the context descriptions attached to the user's notes. Based on this comparison, the system computes a relevance score for each note and presents a ranked list of related notes through the note-taking interface (Figure 6.2(c)). A thumbnail of the visualization that was displayed while the user originally recorded each note is also shown (Figure 6.2(d)). An overview of concepts extracted from notes (underlined) and views is shown (Figure 6.2(e)) on-demand. With the note-taking interface, users can either explicitly request related notes at anytime or have the system automatically recommend them after each exploration action.

The integration of our algorithm into the HARVEST system allows it to dynamically surface the most relevant notes from earlier stages in an analysis as users continue the exploration process. We believe that this related notes recommendation feature in HARVEST helps users maintain awareness of relevant information and assists in connection discovery during visual analysis. To evaluate this approach, we now present a case study and discuss its result.

6.8 Case Study

We conducted a case study to explore the implications of recommending related notes during a visual analysis. We were quite interested in looking at the circumstances in which users wanted to access related notes during their tasks. For this, we observed the analysis process of a research analyst working for a major financial services company. He is familiar with data analysis tools such as Microsoft Excel but had never used HARVEST before. He investigated a financial dataset in HARVEST, and recorded notes using its note-taking interface.

The financial dataset consists of around 1000 financial news articles from the New York Times published between August and September of 2007. These articles were selected from a collection of news and business articles provided by Factiva, a division of Dow Jones & Company. The content of the articles was processed by a text analysis tool to identify key entities in the financial domain such as banks, investment firms, markets (e.g., stock, mortgage, credit, debt), financial instruments (e.g., bonds, securities, funds, etc.), government agencies, important persons, and countries.

The research analyst investigated the financial dataset by exploring the relationships among the entities using visualizations in HARVEST. His investigation spanned for two sessions each lasting for 30 minutes and one week apart. For the first session, we turned off the recommendation feature, and for the second session we turned it on. The analyst was told to explore the financial dataset to understand the status of the financial sector around the time when the articles were published. He was allowed to explore and analyze data freely without any task restriction. We recommended to make use of the note-taking interface for recording hypotheses and discoveries during the analysis. At the end of the sessions, his exploration trail was bookmarked and saved. We closely observed the analyst's analysis process, and conducted a short interview at the end of each session.

Session 1 - Without related notes recommendation

In the first session, the analyst typically alternated between analyzing data along different dimensions using various criteria (by issuing queries and interacting with the visualized results), and taking notes to record his thoughts and discoveries. He used separate notes to record (1) what he expected to see from the data (hypotheses), and (2) what he actually saw and thought was/were the reason(s) to explain such trend or pattern in the visualizations (findings). He then grouped notes related to the same topic (e.g., about a specific investment firm). During this session, he created 10 notes and organized them into 4 groups.

During an interview held afterwards, the analyst expressed that the note-taking facility was quite useful. We asked further about the usefulness of identifying related notes in the analysis process. He agreed that it would be useful. He stated that he would like the system to recommend the related notes immediately after the system displays the chart of the newly requested data. He indicated that such recommendations will inform him of what has already been explored, and give him some ideas of how to explore the new result. He also felt it could save time by helping him avoid duplicate work, and by allowing him to start new tasks by building on previous analyses. He felt that these time savings would allow him to go deeper into his analysis.

The analyst also expressed that automatic recommendation of related notes can be useful since he wouldn't have to spend time reading through all his notes to find the few that might be relevant. It can also help him to aggregate insights and discoveries from previous notes more easily. In addition, he mentioned that he would very much appreciate if the system could recommend related notes from a collection of notes shared among other analysts.

These statements were encouraging and affirmed our approach to surface the related notes during an exploration process.

Session 2 - With related notes recommendation

After a week, the analyst continued his analysis of the same Factiva dataset. The archived exploration trail was restored in HARVEST, including the notes previously created by the analyst. We enabled the related notes recommendation in HARVEST during this session and observed his analysis process. The analyst started by revisiting previous views using the exploration trail for recalling what he did during the last session. Then he continued to explore the data using criteria that were not used during the first session.

When the system provided a recommendation for the first time, he read the content of the recommended notes carefully. He also tried to identify the states in the action trail that were associated with the notes without revisiting it using the thumbnails. Later, he only glanced over the recommendation list and focused his attention on the recommended notes that were newly added to the list. During this session, he added new content to four existing notes, all of which were recommended by the system. Interestingly, he didn't create any new notes.

After the analysis session, we conducted a short interview to understand how the related notes recommendation in HARVEST impacted his analysis. When we asked about the relevancy of the recommend notes, the analyst said "They were relevant in the sense that the concepts mentioned in the recommended notes were related to the data I was inspecting. For example, when I was looking at the information about one bank, the system recommended a note I created previously about another bank, which I thought was useful. I think note recommendation could also help me find some of my previous notes related to my current analysis, which I might not realize or totally forget about."

The analyst liked the thumbnail associated with each recommended note because it helped him quickly remember the context of this note. He felt the option of showing/hiding the recommendation quite useful; and said "... so if I didn't want to be distracted during my data analysis I could always hide it, and make it appear later when I needed it."

Towards the end of the interview, the analyst suggested a few improvements to the system. He would have liked the related concepts to be highlighted in the recommended notes so he could quickly determine if a recommended note is either useful or not without having to read through the whole note. He currently felt it was difficult to revisit views from the notes and to revisit notes from the action trail; and asked for an efficient way to revisit visualizations without having to lose sight on the current analysis process. He also expressed that it would be better if the notes are displayed with a thumbnail of the linked visualization states in the knowledge view, similar to how thumbnails are displayed next to notes in the related notes recommendation list.

6.9 Discussion

We performed the case study to understand the circumstances in which related notes recommendation is found to be useful. Initially, we assumed that the recommendation would be relevant only for longer analysis processes with a large number of notes, hence our study design of two sessions spread out over a week. However, the research analyst who participated in the case study performed the analysis in two short sessions and just recorded 10 notes with relatively small action trails. Still, the analyst created ZERO new notes in the second session, always editing old notes recommended by the system. It is exactly what we want to encourage — *'connections between insights instead of a bunch of small individual insights'*. We believe that with the related notes recommendation, users will more often do editing, re-association and multi-association of notes during an analysis. Thus, the related notes recommendation helps to create awareness of relevant information from the past with respect to the analyst's current line of inquiry and encourages connection discovery during visual analysis.

In addition, the identification of related notes and views using the context description provides a new way of retrieving visualization from past or other collaborator's analysis. This approach, in addition to keyword and view similarity based search methods [116], can help analysts review past analyses.

Retrieval of related items can also be helpful during the sensemaking process. In HARVEST, we used the context-based retrieval algorithm to recommend related items during the information foraging process. Whenever analysts created a new view, the related notes, views and concepts are retrieved and shown. Similarly, recommendation of related views, notes and concepts can be made when they select or modify an existing note, related views, notes and concepts to that note can be looked up and recommended to them. Thus, analysts can locate related notes within the note-taking interface when relevant information is either scattered spatially or distributed in different discussion threads. After locating the related notes, users may be interested in combining them into a group or a note.

6.10 Conclusion and Future Work

In this chapter, we described a context-based retrieval algorithm that retrieves views, notes and concepts from users past analysis related to a view or a note based on their line of inquiry. Whenever users create a view of their data or record a note, we derive a context description for the view or note from their line of inquiry. Our algorithm then uses these context descriptions to retrieve the most relevant views and notes from past analyses.

Using our proposed approach, we have implemented a related notes recommendation feature in HARVEST, a web based visual analytic system. As users create new views during their analysis, HARVEST dynamically applies our algorithm to recommend the most relevant notes from past analyses. An overview of related notes is presented as a ranked list of notes along with a thumbnail of associated views in the note-taking interface. An overview of related concepts is also shown using a tag cloud. Both overviews are updated after each exploration action. Finally we presented a case study in which a research analyst investigated a dataset using the HARVEST system. Our observations of the analyst's analysis process and his feedback support our argument that the identification of related notes, views and concepts is helpful in connection discovery during visual analysis.

Given our findings, there are several areas for future work. From the navigation structure represented in the action trail, it is possible to identify the relationship among the action concepts. Also, the relationship among entities can be derived based on the spatial distribution of notes and text analytics as in some text analysis tools such as Jigsaw [119] and Entity Workspace [24]. Hence, in the future, the relationship among action concepts and entities can be derived from the action trails, and studied using interactive graph visualization. We believe this can clearly bring out the information structure that evolves during the user's exploration process, and can provide a better overview of the implicit connections among concepts during a visual analysis.

In the next chapter, we provide concluding remarks about the work discussed in this dissertation, and recommend future work.

Chapter 7

Conclusion

> *Katrathu kaimann alavu, kallathathu ulaga alavu (What you know is as big as the size of your palm, what you do not know is as big as the size of the universe)* — Thiruvalluvar, Thirukkural, around 200 BC.

In the preceding chapters, we presented generic models and tools for supporting the sensemaking process in visual analytics. In Section 1.2 we presented the requirements for supporting an effective sensemaking process during interactive data exploration within a visual analytics application. Our approach to support the sensemaking process in visual analytics enabled analysts to capture aspects of interest while interactively exploring data; and to support analytical tasks such as reviewing, reusing and sharing these.

This final chapter contains concluding remarks about the work presented in this dissertation. First, we summarize the main contributions of this dissertation to support the sensemaking process in visual analytics. Secondly, we present implications of these contributions, and discuss opportunities for future work.

7.1 Contributions

Interactive visual exploration of data can lead to many discoveries in terms of relations, patterns, outliers and so on. It is difficult for the human working memory to keep track of all findings during a visual analysis. Also, synthesis of many different findings and relations between those findings increases the information overload and thereby hinders the sensemaking process further.

In this dissertation, support for the sensemaking process in visual analytics was investigated. The key research question, introduced in Chapter 1, was: *How to support users in their sensemaking process during interactive exploration of data?*

To answer this question, we mainly focused on how to support users to capture, reuse, review, share, and present the key aspects of interest concerning the analysis process and the findings during interactive exploration of data. First, we presented a sensemaking framework in Chapter 3 that contains three linked views: a data view, a navigation view and a knowledge view. Using this framework, the analysis process was automatically captured based on an analyst's action trails in the data view. Also, it provided an opportunity for the analyst to manually keep track of his/her analysis process using notes in the knowledge view. We showed that enabling analysts to capture findings along with provenance proved to be an important support for the sensemaking process. They could build a case by organizing the findings. Thus, they could effectively ground their analysis; and defend their judgment using the provenance information.

Secondly, we enabled analysts to capture data selections as semantic zones during an analysis, and to reuse these zones on different subsets of data. Data selection techniques such as dynamic queries and brushing help users to progressively converge on interesting data items. Also, they can edit these selections, and thereby perform a divergent analysis. A zone contains the specification of a data selection with a label provided by analysts. In Chapter 4, we presented a Select & Slice table that helped analysts to gain an overview of the distribution of items across different zones and subsets of data. It provided an opportunity to compare the distribution of items side-by-side, and to build a case. The Select & Slice table is a good example of effective capture and reuse of an interesting aspect of the exploration process. Analysts commented that the *capture*, *reuse* and *compare* tasks supported by the Select & Slice table was a natural way of doing analysis with data selection. Otherwise, reasoning with selections was a laborious task.

Finally, exploration overviews and searching techniques based on keywords, content similarity, and context helped analysts to develop awareness over the key aspects of the exploration concerning the analysis process and findings. On one hand, they can proactively search analysis processes and findings for reviewing purposes as described in Chapter 5. On the other hand, they can use the system to discover implicit connections between findings and the current line of inquiry, and recommend these related findings during an interactive data exploration, as discussed in Chapter 6.

An overview of models and tools to support the sensemaking process described in this dissertation is shown in Table 7.1. The interesting aspects concerning an analysis investigated in this dissertation are the analysis process and findings. The four main tasks for supporting the sensemaking process in visual analytics are capture, review, reuse, and share and present. The table shows models and techniques developed to support these tasks while handling analysis processes and findings.

Capture. An analysis process in the data view is captured as an action trail (Chapter 3). The action trail is a hybrid state-action model that captures the visualization and data specifications, as well as the object interest profile of visualization states. It supports a branching history model. Findings can be recorded as notes in the knowledge view (Chapter 3). These notes can be linked to a visualization state in an action trail. These notes can be organized into groups, and connected using arrows using diagramming techniques. Data selections are captured as semantic zones (Chapter 4).

Reuse. Visualization states archived in action trails can be revisited and reused via the navigation view and the knowledge view (Chapter 3). Users can select a node in the history tree (Section 3.4.2), or a note bookmark in the knowledge view to revisit a visualization state. Zones can be reused to select items from different subsets of data in the Select & Slice table (Chapter 4).

Table 7.1: An overview of models and tools developed to support the sensemaking process in visual analytics in this dissertation.

Interesting Aspects of Analysis	**Capture**	**Review**	**Reuse**	**Share and present**
Analysis process	Action trails using history tree representation (*Chapter 3*)	Exploration Awareness through key aspects overview, and keyword and similarity retrieval mechanism (*Chapter 5*)	Visualization specifications (*Chapter 3*)	Action trails (*Chapter 3*)
Findings	Notes and causal relationships using diagramming techniques (*Chapter 3*)	Review visualizations via notes (*Chapter 3*); knowledge diagrams (*Chapter 3*); and identify related notes (*Chapter 6*).	Visualization specifications (*Chapter 3*)	Knowledge maps (*Chapter 3*)
	Semantic Zones (*Chapter 4*)	using the Select & Slice table (*Chapter 4*)	Semantic Zones (*Chapter 4*)	Semantic Zones and Select & Slice table(*Chapter 4*)

Review. We developed a user's information interest model to extract the key aspects of the exploration process: visualization and data transforms, and medium interest (viewed) and high interest (selected) objects. A tag cloud representation was used to provide an overview of these key aspects of the exploration process. We enabled users to perform keyword search on all text that plays a role in these key aspects. Also, we enabled users to retrieve visualization states based on content similarity. Using these tools, analysts can proactively develop awareness about what has been done and found in an analysis, and review it. We also provide a simple keyword search to retrieve notes. These techniques are described in Chapter 5. In Chapter 6, we developed a context based retrieval mechanism to assist in connection discovery during an analysis by uncovering connections to users' past work that would normally go unnoticed. In Chapter 4, we developed comparison and keyword search functionalities to the Select & Slice table for reasoning based on data selection.

Share and present. Action trails, notes, zones, Object Interest Profile, and Select & Slice tables can be archived and shared. Except action trails, other items can be exported as HTML, image, rich text or CSV files, and reviewed in other applications.

We implemented these models and tools described in Chapter 3, Chapter 4 and Chapter 5 in Aruvi; and Chapter 6 in HARVEST. Using Aruvi and HARVEST, we studied the implications of these models on a user's sensemaking process. Data analysts from different domains such as software quality, finance, embedded systems, and urban planning used these tools to carry out some of their data analysis tasks. We adopted the short-term and long-term case studies approach to study support offered by these tools for the sensemaking process. The observations of the case studies were used to evaluate the models.

In conclusion,

- The four tasks: capture, reuse, review and share (of key aspects of the exploration process) are vital to support the sensemaking process of the user. They help him to opportunistically mix the information foraging and the sensemaking loops;

- The sensemaking framework with the three views: data view, knowledge view and navigation view, enables users to keep track of their analysis and findings by supporting the above four tasks;

- The history tree representation is a good model for automatically capturing the analysis; however, the key aspects overview gives a better overview of the analysis to users;

- The note taking mechanism that enables users to record findings with a link to visualization states is a basic necessity for supporting the sensemaking process;

- During exploratory data analysis, selection techniques such as dynamic queries and brushing help users to progressively converge on interesting data items, and by editing they can perform a divergent analysis. The Select & Slice table is a better way to capture and reuse selections, and to compare the results of selections. Also, it helps users to rapidly explore multi-dimensional datasets with a similar data structure.

7.2 Future Work

The sensemaking framework, presented in Chapter 3, focused on supporting the analytical activities during interactive exploration data in the data view. The knowledge view and the navigation view supplemented the sensemaking process in the data view. One of the users commented that "it is absolutely intuitive to have the data view and knowledge view side-by-side; and an overview of the exploration process to track back." These additional views supported the sensemaking process in the data view well. In addition, these views helped analysts to extend their analytical activities beyond the data view in terms of note-taking. However, we also observed that our approach so far has limitations. We discuss these issues in the following subsections. The opportunities for extending the sensemaking framework to address these issues are shown in Figure 7.1.

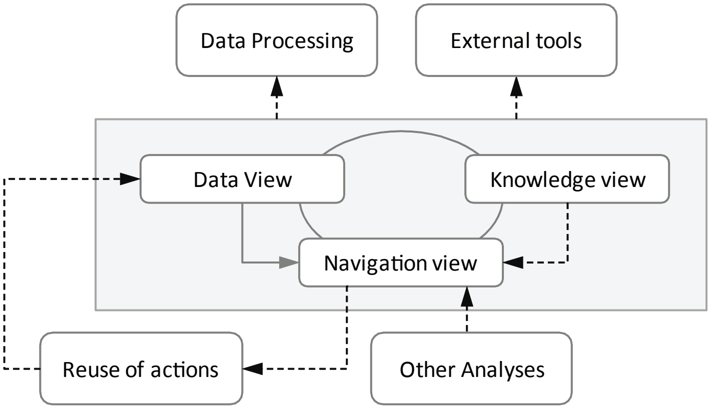


Figure 7.1: Opportunities for extending the sensemaking framework for visual analytics highlighted using dash lines.

More Tools

More visualization tools such as treemaps, graphs and parallel coordinate plots can be developed as a part of the data view. Currently, Aruvi supports SQL based databases. Support for handling hierarchical data structures and XML-based databases can be added to enhance opportunities provided by Aruvi for data exploration.

External Tools

When analysts used Aruvi to explore data, they sometimes used other tools such as Microsoft Excel, ESRI ArcGIS, and the Internet for extracting data and verifying the results. We assumed in the sensemaking framework that all tools used are a part of the data view. In Aruvi, we implemented a simple data view consisting of interactive scatterplots and barcharts attached to a dynamic query interface. But, during a data analysis, analysts make use of multiple tools to solve or understand a problem at hand.

When analysts carry out an analysis using multiple tools, they may need to combine and make sense of results from these tools. For instance, tools used by a software quality analyst for dynamic program analysis are different from tools used for the static code analysis. The analyst has to combine the results from these analyses to build a case about a software system.

The action trails captured by the history tracking mechanism of Aruvi do not include activities from external tools; the key aspects of the exploration process, (visualization and data transformations, view objects and selected objects) are captured only from the data view of Aruvi. Therefore, the exploration overviews, the search and retrieve mechanism, and the connection discovery are limited to the analytical activities within the data view, and do not cover the entire analysis process of an analyst.

One of our users complained that “you cannot capture input and output from products such as Matlab and Microsoft Excel, and archive them as a part of the action trails shown in the navigation view.” There is much analytics software that is specialized in either visualization or automated data analysis. Re-implementing these software tools as a part of the data view, for instance in Aruvi is impossible; a better route is to provide functionality such that external tools can be coupled. One option is to adopt a socket programming approach such that an external tool state is captured as a part of the action trails or notes, and later restore it in the external tool. If that is not possible, a weaker coupling can be implemented, such as capturing screenshots from external tools with comments in the notes view.

Data Provenance

Often data is large and complex, and data preprocessing is an important part of the analysis process. Analysts have to spend much time on data pre-processing, including data cleaning, removing redundant data, and adapting to an input format supported by the application. In a production environment, data preprocessing is also a part of the analytical activity.

Adding data manipulation tools in the data view in addition to visualization tools is practical for analysts. During data analysis, most of them prefer a spreadsheet-like manipulation of columns, so that they can define an attribute based on mathematical combinations of attributes of a dataset. In Aruvi, users can create a derived column at anytime and use it during an analysis. However, Aruvi does not keep track of changes to a dataset. It only records the data and visualization specifications given by the users, and the resulting object interest profile. Thereby, the size of an analysis file is kept minimal, but data changes cannot be recreated.

Of the three views in Aruvi, the data view has minimal tools, such as interactive scatterplots and barcharts attached to dynamic query interfaces to support the analysis process. Though these visualizations are useful to study the distribution of items in a dataset, they do not help in analyzing large datasets with tens of attributes. Analysts mostly cleaned their datasets using Microsoft Excel and derived subsets of a large database before working in Aruvi. They manually kept track of the processes involved in data manipulation using notes in the knowledge view.

Reuse of Actions

We found a common behavior which reoccurred during the case studies conducted to evaluate the sensemaking framework (Chapter 3) and the Select & Slice table (Chapter 4). Analysts often captured visualization states and selections not only for recording findings and building a case, but also for reusing these during an analysis process. Each visualization state is used as a kind of a macro to recreate a visualization state. Similarly, data selections were declaratively captured as semantic zones. Analysts could reuse the zones on different subsets of a dataset or on a dataset that has a data structure similar to the original dataset. However, the reuse of visualization states was restricted to revisiting. Often, our analysts expressed interest to have a mechanism for capturing an analysis template from the action trails, such that they can reuse the analysis template on different subsets of the data or on a similar dataset; and compare results of the analyses. In general, capturing an action trail of the exploration process is a minimal exploitation of the actions. Flexible reuse of actions, and comparison of the results of these actions have to be facilitated.

Awareness of Note-taking Activity

We found that analysts spent considerable time on note taking activities during an analysis. Recording and grouping notes, creating flow diagrams, and organizing these diagrams by topic is a large part of the analytical activity. The creation of flow diagrams by connecting notes and multi-level grouping of notes are the most popular features of the notes view. These features did not exist in popular note-taking applications such as Microsoft OneNote[1] and EVERNOTE[2].

Inclusion of note-taking in the system has direct implications on the user's information interest model presented in Chapter 5. We argued there that the key aspects of the exploration process include visualization and data transformations, and viewed and selected objects. However, the concepts (entities) in the notes and the relationship among these concepts, based on the structure of the knowledge diagrams, are also key aspects of the exploration process. We considered entities from notes in the context based visualization and notes retrieval mechanism for supporting connection discovery (Chapter 6). The concept overview (see Figure 6.2e) combines entities from notes and the key exploration aspects from a data view; and presents an overview of the current line of inquiry.

Several analysts expressed interest to capture the analytical activities while taking notes during an analysis. Currently, when analysts revisit a past visualization state, the notes view is not restored. They would like to have an option to revisit the notes view as it looked when the visualization state was captured. With this, they can review their past notes, and revise their current line of inquiry. The key analytical activities in the notes view include creating, editing, grouping, and connecting notes. Aruvi does not capture these activities as a part of the action trails; it just supports an undo-redo mechanism to recover from mistakes. HARVEST captures the creation of notes along with the visualization specifications as a part of the action trails. However, it cannot restore the past notes view

[1] http://office.microsoft.com/en-us/onenote/default.aspx

[2] http://www.evernote.com/

when a past visualization state is revisited. The analytical activities in the notes view can be either captured as a part of action trails from the data view or in an independent action trail. For independently capturing an action trail from the notes view, we can adopt the same approach used for capturing action trails from the data view; and link these action trails using the *user name* and *time* attributes.

When there are many notes in the knowledge view, an overview of the notes structure is required to develop awareness of the note-taking activity. A structural overview of the note-taking activity can be provided by automatically extracting a concept map based on the structure of the knowledge diagrams.

Awareness of other analyses

In Aruvi, we adopted a file based approach to archive the analysis process and findings. Initially, all information related to one analysis of one dataset was stored in one file. However, we found that the file-based approach fell short to support the sensemaking process on a larger scale, concerning multiple analysis tasks of multiple datasets.

Initially, findings were archived along with the analysis process. Analysts could review the findings by opening the entire analysis file. However, they wanted to quickly review the notes first, and then open the entire analysis to investigate further. Also, they wanted to share the notes view across different analyses. To support this, we enabled analysts to work with multiple datasets at the same time in Aruvi. We enabled analysts to export the notes view and the Select & Slice table as images, HTML, SVG and rich text documents. Thus, analysts can review the findings outside Aruvi; but, they cannot revisit the visualizations from the exported documents.

We found that the notes view became the central working area during a sensemaking process. We learned that archiving the findings and analysis processes of the analysts in a centralized archive with links between them can be useful. Figure 7.2 shows the support offered by a centralized archive for the sensemaking process in visual analytics. Using a centralized archive, an overview of analysis processes using notes can be shown to analysts, and they can drill down to an analysis process, or switch between analyses seamlessly. The exploration awareness techniques (Chapter 5) and the connection discovery (Chapter 6) can effectively exploit a centralized archive to support collaborative sensemaking processes without having to shift focus on explicitly loading analyses of different users.

Presentation

Currently, we only enable analysts to record notes using diagramming techniques in the knowledge view. Though the knowledge diagrams are useful, they currently use only keyboard and mouse inputs. Oral notes can help users to more quickly record findings than typing in the knowledge view. Also, there are other advanced input mechanisms such as stylus and ink recognition technologies that can be used to construct knowledge diagrams. The knowledge view can also be designed as a tangible user interface similar to the designer's outpost [81] that is used for supporting collaborative design processes.

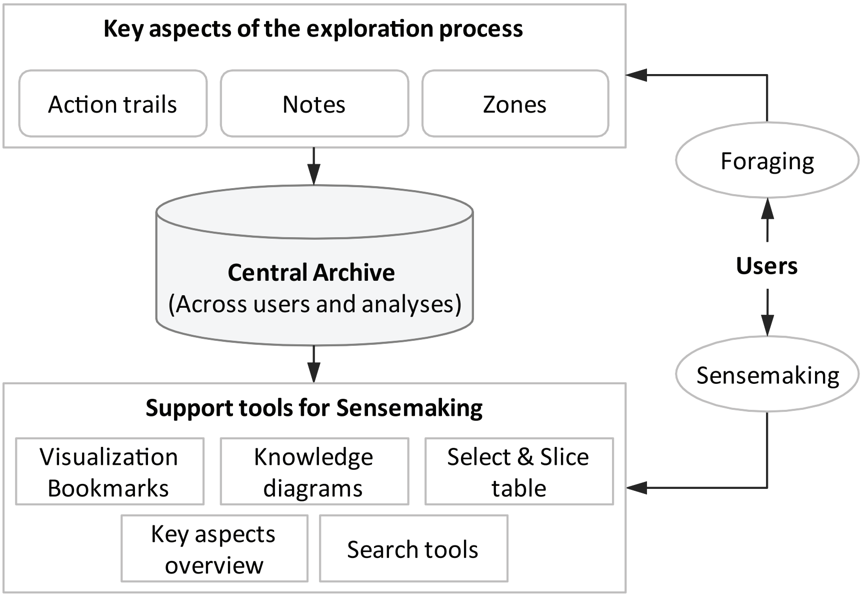


Figure 7.2: A centralized archive to support the sensemaking process in visual analytics.

With the recent advances in multimedia techniques, there are better opportunities to present the analysis results than just sharing a knowledge diagram. For instance, Wohlfart and Hauser [137] have used story telling techniques to present results of volume visualization, which improve both the comprehensibility and credibility of the intended visualization message. Using the visualization stories, users may just watch the presentation passively; in addition, they can reinvestigate the visualization independently from story guidance, offering the ability to verify, confirm, or even disapprove the presented visualization message.

Evaluation

We adopted the short-term and long-term case studies approach to study support offered by these tools for the sensemaking process. In these case studies, we observed analysis processes of the participants, and conducted informal interviews to understand the implications of the models and tools in their analysis process. Based on our observation and participants' feedback, we separately evaluated the sensemaking framework (Chapter 3), the Select & Slice table (Chapter 4), the exploration awareness tools (Chapter 5) and the connection discovery tools (Chapter 6) in this dissertation.

Except the context based retrieval system, all models and tools were implemented in Aruvi. We conducted a longitudinal case study for understanding the sensemaking behavior of the users over time. Three software quality analysts from LaQuSo, the Laboratory for Quality Software in the Netherlands, participated in the case study. We encouraged the analysts to use Aruvi during daily analytical activities. The notes view and the Select

& Slice table were the most popular sensemaking tool among them. However, analysts dropped out of the case study due to issues such as lack of visualization tools in the data view, and lack of support for capturing analytical activities from external tools. We have discussed these issues earlier in this section. These issues have to be addressed for evaluating the support for the sensemaking process based on user experience over time.

Finally, we want to conduct some focused user studies to evaluate and improve the design of the user interfaces such as the metadata view, the key aspects overview, the history tree representation, the notes view and the Select & Slice table presented in this dissertation. An important requirement here is that the subjects are again data analysts focusing on real analytical tasks.

Bibliography

[1] General dynamics c4 systems. *http://www.gdc4s.com/products/*, Accessed May 2010.

[2] Google gapminder. *http://www.gapminder.org/*, Accessed May 2010.

[3] i2 analyst's notebook. *http://www.i2group.com/*, Accessed May 2010.

[4] Iris explorer. *http://www.nag.co.uk/Welcome_IEC.asp*, Accessed May 2010.

[5] Magnaview. *http://www.magnaview.nl/*, Accessed May 2010.

[6] Oculus. *http://www.oculusinfo.com*, Accessed May 2010.

[7] Palantir. *http://www.palantirtech.com/*, Accessed May 2010.

[8] Qlikview. *http://www.qlikview.com/*, Accessed May 2010.

[9] Tableau software. *http://www.tableausoftware.com/*, Accessed May 2010.

[10] Tibco spotfire. *http://spotfire.tibco.com/Products/Default.aspx*, Accessed May 2010.

[11] Visual analytics inc. *http://www.visualanalytics.com/*, Accessed May 2010.

[12] ABOWD, G. D., AND DIX, A. J. Giving undo attention. *Interacting with Computers 4*, 3 (1992), 317–342.

[13] AHLBERG, C., AND SHNEIDERMAN, B. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proc. ACM CHI '94* (1994), pp. 313–317.

[14] ALBERS, M. J. Goal-driven task analysis: improving situation awareness for complex problem-solving. In *SIGDOC '98: Proceedings of the 16th annual international conference on Computer documentation* (New York, NY, USA, 1998), ACM, pp. 234–242.

[15] AMAR, R. A., AND STASKO, J. T. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics 11*, 4 (July-Aug. 2005), 432 –442.

[16] ANDERSON, J. R., AND PIROLLI, P. L. Spread of activation. *Journal of Experimental Psychology: Learning, Memory and Cognition 10* (1984), 791–798.

[17] BATES, M. J. Design of browsing and berrypicking techniques for the online search interface. *Online Review 13* (1989), 407–424.

[18] BAUER, M. I., AND LAIRD, J. J. How diagrams can improve reasoning. *Psychological Science 4*, 6 (1993), 372–378.

[19] BAVOIL, L., CALLAHAN, S., CROSSNO, P., FREIRE, J., SCHEIDEGGER, C., SILVA, C., AND VO, H. Vistrails: Enabling interactive, multiple-view visualizations. In *Proc. IEEE Visualization '05* (2005), IEEE Computer Society Press, pp. 135–142.

[20] BECKER, R. A., AND CLEVELAND, W. S. Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142.

[21] BEECH, J. R., AND COLLEY, A. M., Eds. *Cognitive Approaches to Reading*. John Wiley & Sons, Limited, 1987, ch. Reading and working memory, pp. 57–86.

[22] BERTIN, J. *Semiology of graphics*. University of Wisconsin Press, 1983.

[23] BHATT, G. D. Organizing knowledge in the knowledge development cycle. *Journal of Knowledge Management 4*, 1 (2000), 15–26.

[24] BIER, E., CARD, S., AND BODNAR, J. Entity-based collaboration tools for intelligence analysis. *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), 99–106.

[25] BRODLIE, K., BRANKIN, L., BANECKI, G., GAY, A., POON, A., AND WRIGHT, H. GRASPARC - a problem solving environment integrating computation and visualization. In *Proc. IEEE Visualization '93* (1993), IEEE Computer Society Press, pp. 102–109.

[26] BURKHARD, R. A. *Knowledge and Information Visualization*, vol. 3426/2005 of *Lecture Notes in Computer Science*. Springer, 2005, ch. Towards a Framework and a Model for Knowledge Visualization: Synergies Between Information and Knowledge Visualization, pp. 238–255.

[27] BUZAN, T., AND BUZAN, B. *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*. Penguin Books, 1993.

[28] CALLAHAN, S. P., FREIRE, J., SANTOS, E., SCHEIDEGGER, C. E., SILVA, C. T., AND VO, H. T. Vistrails: visualization meets data management. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (2006), ACM Press, pp. 745–747.

[29] CARD, S. K., AND MACKINLAY, J. The structure of the information visualization design space. In *Proceedings of the IEEE InfoVis '97* (1997), IEEE Computer Society, pp. 92–99.

[30] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B., Eds. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[31] CHEN, C. *Mapping Scientific Frontiers: The Quest for Knowledge Visualization.* Springer-Verlag, 2003.

[32] CHEN, H. Compound brushing. In *Proc. IEEE InfoVis '03* (Oct 2003), pp. 181–188.

[33] CHENG, W.-H., AND GOTZ, D. Context-based page unit recommendation for web-based sensemaking tasks. In *IUI '09: Proc. international conference on Intelligent user interfaces* (New York, NY, USA, 2008), ACM, pp. 107–116.

[34] CHI, E. H.-H., AND RIEDL, J. An operator interaction framework for visualization systems. In *Proceedings of the IEEE INFOVIS '98* (1998), IEEE Computer Society, pp. 63–70.

[35] CHUAH, M. C., ROTH, S. F., MATTIS, J., AND KOLOJEJCHICK, J. Sdm: selective dynamic manipulation of visualizations. In *UIST '95: Proc. ACM symposium on User interface and software technology* (1995), ACM Press, pp. 61–70.

[36] CLARK, H. H., AND BRENNAN, S. E. *Perspectives on socially shared cognition.* American Psychological Association, 1991, ch. Grounding in Communication, pp. 127–149.

[37] COLLINS, A. M., AND LOFTUS, E. F. A spreading-activation theory of semantic processing. *Psychological Review 82*, 6 (November 1975), 407–428.

[38] COWAN, N. *Attention and memory: An integrated framework*, vol. 26 of *Oxford Psychology Series.* Oxford University Press, 1998.

[39] CRAFT, B., AND CAIRNS, P. Beyond guidelines: what can we learn from the visual information seeking mantra? In *Proc. Information visualization IV'05.* (July 2005), pp. 110 – 118.

[40] DADZIE, A.-S., IRIA, J., PETRELLI, D., AND XIA, L. The xmediabox: Sensemaking through the use of knowledge lenses. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web* (2009), Springer-Verlag, pp. 811–815.

[41] DADZIE, A.-S., LANFRANCHI, V., AND PETRELLI, D. Seeing is believing: Linking data with knowledge. *Information Visualization 8*, 3 (2009), 197–211.

[42] DENISOVICH, I. Software support for annotation of visualized data using hand-drawn marks. In *IV '05: Proc. IEEE Information Visualisation (IV'05)* (2005), IEEE Computer Society Press, pp. 807–813.

[43] DERTHICK, M., KOLOJEJCHICK, J., AND ROTH, S. F. An interactive visual query environment for exploring data. In *Proc. ACM UIST '97* (New York, NY, USA, 1997), ACM, pp. 189–198.

[44] DERTHICK, M., AND ROTH, S. F. Enhancing data exploration with a branching history of user operations. *Knowledge-Based Systems 14*, 1-2 (Mar. 2001), 65–74.

[45] DERVIN, B. An overview of sense-making research: Concepts, methods, and results to date. In *International Communication Association* (Dallas, TX, USA., 1983).

[46] DOLEISCH, H., GASSER, M., AND HAUSER, H. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003* (2003), pp. 239–248.

[47] DYKES, J., MACEACHREN, A. M., AND KRAAK, M.-J., Eds. *Exploring Geovisualization*. Elsevier Ltd., 2005.

[48] ELLIS, D., COX, D., AND HALL, K. A comparison of the information seeking patterns of researchers in the physical and social sciences. *Journal of Documentation* (1993), 356–369.

[49] ELLIS, S. E., AND GROTH, D. P. A collaborative annotation system for data visualization. In *AVI '04: Proc. ACM SIGCHI Advanced visual interfaces* (2004), ACM Press, pp. 411–414.

[50] ELLKVIST, T., KOOP, D., ANDERSON, E. W., FREIRE, J., AND SILVA, C. Using provenance to support real-time collaborative design of workflows. 266–279.

[51] ELMQVIST, N., DRAGICEVIC, P., AND FEKETE, J.-D. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1141–1148.

[52] ENDSLEY, M. R. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society 37* (1995), 32–64.

[53] FERRUCCI, D., AND LALLY, A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering 10*, 3-4 (2004), 327–348.

[54] FESTINGER, L. A theory of social comparison processes. *Human Relations* (1954), 114–140.

[55] FEW, S. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2004.

[56] FEW, S. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, 2006.

[57] FURNAS, G. W. Generalized fisheye views. *ACM SIGCHI Bulletin 17*, 4 (1986), 16–23.

[58] GOODELL, H., CHIANG, C.-H., KELLEHER, C., BAUMANN, A., AND GRINSTEIN, G. Collecting and harnessing rich session histories. In *IV '06: Proc. IEEE Information Visualization* (2006), IEEE Computer Society, pp. 117–123.

[59] Gotz, D., Wen, Z., Lu, J., Kissa, P., Zhou, M. X., Cao, N., Qian, W. H., and Lui, S. X. Harvest - visualization and analysis for the masses. *Proc. IEEE InfoVis '08 Poster* (2008).

[60] Gotz, D., and Zhou, M. X. Characterizing users visual analytic activity for insight provenance. *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), 123–130.

[61] Gotz, D., Zhou, M. X., and Aggarwal, V. Interactive visual synthesis of analytic knowledge. In *Proc. IEEE Symposium on Visual Analytics Science and Technology* (October 2006), pp. 51–58.

[62] Haeberli, P. E. Conman: a visual programming language for interactive graphics. *SIGGRAPH Computer Graphics 22*, 4 (1988), 103–111.

[63] Hearst, M. A., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K.-P. Finding the flow in web site search. *Communications of the ACM 45*, 9 (2002), 42–49.

[64] Heer, J., and Agrawala, M. Design considerations for collaborative visual analytics. *IEEE Symposium on Visual Analytics Science and Technology, 2007. VAST 2007.* (2007), 171–178.

[65] Heer, J., Agrawala, M., and Willett, W. Generalized selection via interactive query relaxation. In *Proc. ACM CHI '08* (2008), pp. 959–968.

[66] Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. In *Proc. IEEE InfoVis '08* (2008).

[67] Heer, J., Viégas, F. B., and Wattenberg, M. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM Press, pp. 1029–1038.

[68] Hutchins, E. *Cognition in the Wild.* MIT Press, Cambridge, MA, USA., 1994.

[69] Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B. B., Druin, A., Plaisant, C., Beaudouin-Lafon, M., Conversy, S., Evans, H., Hansen, H., Roussel, N., and Eiderbäck, B. Technology probes: inspiring design for and with families. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (2003), pp. 17–24.

[70] i2 Analyst's Notebook. http://www.i2inc.com/, 2009.

[71] Jankun-Kelly, T. J., Ma, K.-L., and Gertz, M. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 357–369.

[72] JELEN, B., AND ALEXANDER, M. *Pivot Table Data Crunching*. Que Corp., Indianapolis, IN, USA, 2005.

[73] JOHNSON-LAIRD, P. N. *Mental models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press, 1983.

[74] JOHNSON-LAIRD, P. N., AND BYRNE, R., Eds. *Deduction*. Lawrence Erlbaum Associates., Hillsdale, NJ, 1991.

[75] JONKER, D., WRIGHT, W., SCHROH, D., PROULX, P., AND CORT, B. Information triage with TRIST. In *International Conference on Intelligence Analysis* (May 2–4 2005).

[76] JR., R. H. B. *International Waterfall Classification System*. Outskirts Press, 2006.

[77] KEIM, D., ANDRIENKO, G., FEKETE, J.-D., GÖRG, C., KOHLHAMMER, J., AND MELANÇON, G. Visual analytics: Definition, process, and challenges. 154–175.

[78] KEIM, D. A., MANSMANN, F., SCHNEIDEWIND, J., THOMAS, J., AND ZIEGLER, H. *Visual Data Mining*. Springer, 2008, ch. Visual Analytics: Scope and Challenges, pp. 76–90.

[79] KEIM, D. A., MANSMANN, F., AND THOMAS, J. Visual analytics: How much visualization and how much analytics? *SIGKDD Explorations 12*, 2 (December 2009), 5–8.

[80] KLEIN, G. *Decision Making in Action: Model and Methods*. Ablex, 1993, ch. A recognition-primed decision model of rapid decision making, pp. 138–147.

[81] KLEMMER, S. R., NEWMAN, M. W., FARRELL, R., BILEZIKJIAN, M., AND LANDAY, J. A. The designers' outpost: a tangible interface for collaborative web site. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), pp. 1–10.

[82] KONYHA, Z., MATKOVIC, K., GRACANIN, D., JELOVIC, M., AND HAUSER, H. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1373–1385.

[83] KUHLTHAU, C. C. Inside the search process: Information seeking from the users perspective. *Journal of the American Society for Information Science 42* (1991), 361–371.

[84] LARKIN, J. H., AND SIMON, H. A. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science 11*, 1 (1987), 65–100.

[85] LEE, J. P., AND GRINSTEIN, G. An architecture for retaining and analyzing visual explorations of databases. In *Proceedings of IEEE Visualization* (1995), pp. 101–108.

[86] LIVNY, M., RAMAKRISHNAN, R., BEYER, K., CHEN, G., DONJERKOVIC, D., LAWANDE, S., MYLLYMAKI, J., AND WENGER, K. Devise: integrated querying and visual exploration of large datasets. In *Proc. ACM SIGMOD '97,* (May 1997), pp. 301–312.

[87] MA, K.-L. Image graphs - a novel approach to visual data exploration. In *Proceedings of IEEE Visualization 1999 conference* (1999), IEEE Computer Society Press, pp. 81–88.

[88] MACEACHREN, A. M. *How Maps Work: Representation, Visualization and Design.* The Guilford Press, 2004.

[89] MACKAY, W. *Users and Customizable Software: A Co-Adaptive Phenomenon.* PhD thesis, Massachusetts Institute of Technology, 1990.

[90] MACKINLAY, J. D., ROBERTSON, G. G., AND CARD, S. K. The perspective wall: detail and context smoothly integrated. In *Proceedings of the ACM SIGCHI conference on Human factors in computing systems* (1991), ACM, pp. 173–176.

[91] MARTIN, A. R., AND WARD, M. O. High dimensional brushing for interactive exploration of multivariate data. In *Proc. IEEE Visualization '95* (Nov 1995), pp. 271–278.

[92] MARTIN, R. C. Design principles and design patterns. *http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf*, Accessed on Sept. 14 2009.

[93] MCCORMICK, B. H., DEFANTI, T. A., AND BROWN, M. D., Eds. *Visualization in scientific computing, 10(1).* ACM Press, 1987.

[94] MOSCOVICI, S., AND NEMETH, C. *Social psychology: Classic and contemporary integrations.* Chicago, IL: Rand McNally, 1974, ch. Minority influence, pp. 217–249.

[95] MUNZNER, T. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 921–928.

[96] MYATT, G. J. *Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining.* John Wiley & Sons, Limited, 2007.

[97] NABI, M., BLAGOJEVIC, M., BASTEN, T., GEILEN, M., AND HENDRIKS, T. Configuring multi-objective evolutionary algorithms for design-space exploration of wireless sensor networks. In *To appear in ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks, '09* (2009).

[98] NORMAN, D. A. *Learning and Memory.* W. H. Freeman & Co., New York, NY, USA, 1982.

[99] OAKHILL, J., AND GARNHAM, A., Eds. *Mental Models in Cognitive Science.* Psychology Press, 1996, ch. Models, Arguments, and Decisions, pp. 95–118.

[100] OLSTON, C., STONEBRAKER, M., AIKEN, A., AND HELLERSTEIN, J. M. Viqing: Visual interactive querying. In *Proc. IEEE Visual Languages '98* (Washington, DC, USA, 1998), IEEE Computer Society, p. 162.

[101] PIKE, W., BRUCE, J., BADDELEY, B., BEST, D., FRANKLIN, L., MAY, R., RICE, D., RIENSCHE, R., AND YOUNKIN, K. The scalable reasoning system: Lightweight visualization for distributed analytics. *Information Visualization 8*, 1 (2009), 71–84.

[102] PIROLLI, P., AND CARD, S. Information foraging. *Psychological Review 106 (4)* (1999), 643–675.

[103] PIROLLI, P., AND CARD, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *International Conference on Intelligence Analysis* (May 2–4 2005).

[104] PIROLLI, P., AND RAO, R. Table lens as a tool for making sense of data. In *Proceedings of the workshop on Advanced visual interfaces* (1996), ACM, pp. 67–80.

[105] PLAISANT, C. The challenge of information visualization evaluation. In *AVI '04: Proc. Advanced visual interfaces* (2004), pp. 109–116.

[106] PLAISANT, C., MILASH, B., ROSE, A., WIDOFF, S., AND SHNEIDERMAN, B. Lifelines: visualizing personal histories. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems* (1996), ACM Press, pp. 221–227.

[107] RUSSELL, D. M., STEFIK, M. J., PIROLLI, P., AND CARD, S. K. The cost structure of sensemaking. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems* (1993), ACM Press, pp. 269–276.

[108] SALZBRUNN, T., AND SCHEUERMANN, G. Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1601–1612.

[109] SCHEIDEGGER, C., VO, H., KOOP, D., FREIRE, J., AND SILVA, C. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov.-Dec. 2007), 1560–1567.

[110] SCHROEDER, W., MARTIN, K., AND LORENSEN, B. *The Visualization Toolkit: An Object Oriented Approach to 3D Graphics.* Kitware, 2006.

[111] SEREBRENIK, A., ROUBTSOV, S., AND VAN DEN BRAND, M. D_n-based architecture assessment of java open source software systems. In *Proc. International Conference on Program Comprehension '09* (2009).

[112] SHARP, H., ROGERS, Y., AND PREECE, J. *Interaction Design: Beyond Human-Computer Interaction*, 2 ed. Wiley, 2007.

[113] SHNEIDERMAN, B. Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (1994), 70–77.

[114] SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. *Proceeding of the IEEE Symposium on Visual Languages,* (1996), 336–343.

[115] SHRINIVASAN, Y., GOTZ, D., AND LU, J. Connecting the dots in visual analysis. pp. 123 –130.

[116] SHRINIVASAN, Y. B., AND VAN WIJK, J. Supporting exploration awareness in information visualization. *IEEE Computer Graphics and Applications 29*, 5 (2009), 34–43.

[117] SHRINIVASAN, Y. B., AND VAN WIJK, J. J. Supporting the analytical reasoning process in information visualization. In *Proc. ACM CHI '08* (2008), pp. 1237–1246.

[118] STASKO, J., GÖRG, C., AND LIU, Z. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (2008), 118–132.

[119] STASKO, J., GORG, C., LIU, Z., AND SINGHAL, K. Jigsaw: Supporting investigative analysis through interactive visualization. *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on* (2007), 131–138.

[120] STOLTE, C., TANG, D., AND HANRAHAN, P. Polaris: a system for query, analysis, and visualization of multidimensional databases. vol. 51, pp. 75–84.

[121] THEUS, M. Interactive data visualization using mondrian. *Journal of Statistical Software 7*, 11 (11 2002), 1–9.

[122] THOMAS, J. J., AND COOK, K. A., Eds. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005.

[123] TOLLIS, I. G., BATTISTA, G. D., EADES, P., AND TAMASSIA, R. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[124] TUFTE, E. *Envisioning information*. Graphics Press, 1990.

[125] TUFTE, E. R. *The visual display of quantitative information*. Graphics Press, 1986.

[126] TUFTE, E. R. *Visual explanations: images and quantities, evidence and narrative*. Graphics Press, 1997.

[127] UPSON, C., FAULHABER, JR., T., KAMINS, D., LAIDLAW, D. H., SCHLEGEL, D., VROOM, J., GURWITZ, R., AND VAN DAM, A. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics Applications 9*, 4 (1989), 30–42.

[128] VAN WIJK, J. J. The value of visualization. In *Proceedings of IEEE Visualization 2005 conference* (2005), IEEE Computer Society Press, pp. 79–86.

[129] VENKATESH, V., MORRIS, M. G., DAVIS, G. B., AND DAVIS, F. D. User acceptance of information technology: Toward a unified view. *MIS Quarterly 27*, 3 (2003), 425–478.

[130] VIÉGAS, F. B., AND WATTENBERG, M. Communication-minded visualization: A call to action. Tech. Rep. 4, IBM Systems Journal, 2006.

[131] VIEGAS, F. B., WATTENBERG, M., VAN HAM, F., KRISS, J., AND MCKEON, M. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1121–1128.

[132] WARD, M. O. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94* (1994), pp. 326–333.

[133] WARE, C. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2004.

[134] WEAVER, C. Building highly-coordinated visualizations in improvise. In *Proc. IEEE InfoVis '04* (2004), pp. 159–166.

[135] WEAVER, C. Multidimensional visual analysis using cross-filtered views. In *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), pp. 163–170.

[136] WILLIAMSON, C., AND SHNEIDERMAN, B. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (1992), ACM, pp. 338–346.

[137] WOHLFART, M., AND HAUSER, H. Story telling for presentation in volume visualization. In *Eurographics/ IEEE-VGTC Symposium on Visualization* (2007), K. Museth, T. Möller, and A. Ynnerman, Eds., Eurographics Association, pp. 91–98.

[138] WONG, P. C., CHIN, G., FOOTE, H., MACKEY, P., AND THOMAS, J. Have Green - a visual analytics framework for large semantic graphs. *IEEE Symposium on Visual Analytics Science And Technology* (31 2006-Nov. 2 2006), 67–74.

[139] WRIGHT, W., SCHROH, D., PROULX, P., SKABURSKIS, A., AND CORT, B. The sandbox for analysis: concepts and methods. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM Press, pp. 801–810.

[140] YANG, D., RUNDENSTEINER, E. A., AND WARD, M. O. Nugget discovery in visual exploration environments by query consolidation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), pp. 603–612.

List of Publications

The major part of this thesis is based on the following publications:

Peer Reviewed Articles

Shrinivasan, Y.B., and Van Wijk, .J.J, Supporting exploratory data analysis using the Select & Slice table, *Computer Graphics Forum: Eurographics/IEEE Symposium on Visualization* (EuroVis '10), To appear, 2010.

Shrinivasan, Y.B, Gotz .D and Lu .J, Connecting the dots in visual analysis, *IEEE Visual Analytics Science and Technology*, pp. 123–130, October 2009.

Shrinivasan Y.B, and Van Wijk .J.J, Support exploration awareness in information visualization, *IEEE Computer Graphics and Applications*, vol. 29, no. 5, pp. 34–43, Sep./Oct. 2009.

Shrinivasan, Y. B, and Van Wijk, .J.J, Supporting exploration awareness for visual analytics, *IEEE Visual Analytics Science and Technology*, pp. 185–186, October 2008.

Shrinivasan Y.B and Van Wijk .J.J, Support the analytical reasoning process in information visualization, *ACM Human Factors in Computing Systems* (CHI), pp. 1237–1246, April 2008.

Doctoral Colloquium

Shrinivasan, Y.B., Navigation and synthesis in interactive visualization, Doctoral Colloquium, IEEE Visual Analytics Science and Technology, October 2007.

Refereed Short Papers

Shrinivasan, Y.B. and Gotz. D., Connecting the dots with related notes, CHI '09 Extended Abstracts on Human Factors in Computing Systems, April 2009.

Shrinivasan, Y. B. and Van Wijk, .J.J, VisPad: Integrating visualization, navigation and synthesis, IEEE Visual Analytics Science and Technology, 209-210, October 2007. *(Best Poster Award).*

Summary

Supporting the Sensemaking Process in Visual Analytics

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces. It involves interactive exploration of data using visualizations and automated data analysis to gain insight, and to ultimately make better decisions. It aims to support the sensemaking process in which information is collected, organized and analyzed to form new knowledge and inform further action. Interactive visual exploration of the data can lead to many discoveries in terms of relations, patterns, outliers and so on. It is difficult for the human working memory to keep track of all findings during a visual analysis. Also, synthesis of many different findings and relations between those findings increase the information overload and thereby hinders the sensemaking process further. The central theme of this dissertation is

How to support users in their sensemaking process during interactive exploration of data?

To support the sensemaking process in visual analytics, we mainly focus on how to support users to capture, reuse, review, share, and present the key aspects of interest concerning the analysis process and the findings during interactive exploration of data. For this, we have developed generic models and tools that enable users to capture findings with provenance, and construct arguments; and to review, revise and share their visual analysis.

First, we present a sensemaking framework for visual analytics that contains three linked views: a data view, a navigation view and a knowledge view for supporting the sense-making process. The data view offers interactive data visualization tools. The navigation view automatically captures the interaction history using a semantically rich action model and provides an overview of the analysis structure. The knowledge view is a basic graphics editor that helps users to record findings with provenance and to organize findings into claims using diagramming techniques. Users can exploit automatically captured interaction history and manually recorded findings to review and revise their visual analysis. Thus, the analysis process can be archived and shared with others for collaborative visual analysis.

Secondly, we enable analysts to capture data selections as semantic zones during an analysis, and to reuse these zones on different subsets of data. We present a Select & Slice table that helps analysts to capture, manipulate, and reuse these zones more explicitly

during exploratory data analysis. Users can reuse zones, combine zones, and compare and trace items of interest across different semantic zones and data slices.

Finally, exploration overviews and searching techniques based on keywords, content similarity, and context helped analysts to develop awareness over the key aspects of the exploration concerning the analysis process and findings. On one hand, they can proactively search analysis processes and findings for reviewing purposes. On the other hand, they can use the system to discover implicit connections between findings and the current line of inquiry, and recommend these related findings during an interactive data exploration.

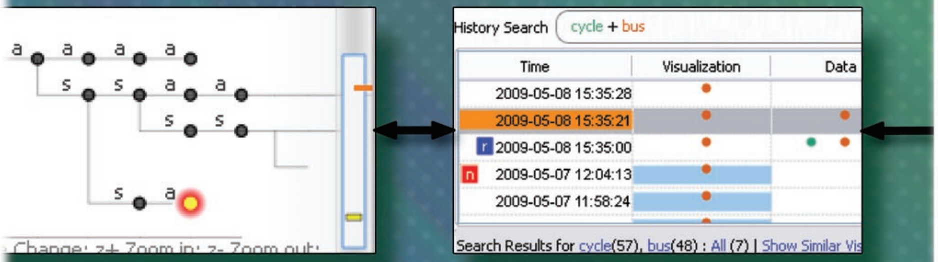
We implemented the models and tools described in this dissertation in Aruvi and HARVEST. Using Aruvi and HARVEST, we studied the implications of these models on a user's sensemaking process. We adopted the short-term and long-term case studies approach to study support offered by these tools for the sensemaking process. The observations of the case studies were used to evaluate the models.

Curriculum Vitae

Yedendra Babu Shrinivasan was born on 25 December 1981 in Tiruvallur, India. He completed his Bachelor of Engineering in Geoinformatics with honors from College of Engineering Guindy, Anna University, India in 2003. He obtained his Master of Science in Geoinformatics with honors from International Institute for Geoinformation Science and Earth Observation, University of Twente, The Netherlands, in 2005. After that he worked as a scientist at National Remote Sensing Center, Indian Space Research Organization, India where he focused on developing decision support systems for disaster management. Since 2006, he has been a PhD Student at Technische Universiteit Eindhoven (TU/e) under the supervision of prof. dr. ir. Jack van Wijk. His research interests include information visualization, human-computer interaction, cognitive science and visual analytics. As of August 2010, he will start working as a research staff member at IBM Research, Bangalore, India.

Interactive visual exploration of the data can lead to many discoveries in terms of relations, patterns, outliers and so on. It is difficult for the human working memory to keep track of all findings during a visual analysis. Also, synthesis of many different findings and relations between those findings increase the information overload and thereby hinders the sensemaking process further. So, there is a need to support users in their sensemaking process during interactive exploration of data.

One approach to support this process is to enable them to capture, reuse, review, share, and present the key aspects of interest concerning the analysis process and the findings during interactive exploration of data.



	Time	Visualization	Data
	2009-05-08 15:35:28	●	
	2009-05-08 15:35:21	●	●
r	2009-05-08 15:35:00	●	● ●
n	2009-05-07 12:04:13	●	
	2009-05-07 11:58:24	●	

Search Results for cycle(57), bus(48) : All (7) | Show Similar Vis

ISBN: 978-90-386-2229-3

TU/e Technische Universiteit Eindhoven University of Technology