A TWO-STAGE FRAMEWORK FOR DESIGNING VISUAL ANALYTICS SYSTEMS TO AUGMENT ORGANIZATIONAL ANALYTICAL PROCESSES

by

Xiaoyu Wang

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2011

Approved by:

____

Dr. William Ribarsky

____

Dr. Robert Kosara

____

Dr. Zachary Wartell

____

Dr. Richard Souvenir

____

Dr. Shen-En Chen

ABSTRACT

XIAOYU WANG. A two-stage framework for designing visual analytics systems to
augment organizational analytical processes. (Under the direction of DR.
WILLIAM RIBARSKY)

A perennially interesting research topic in the field of visual analytics is how to effectively develop systems that support organizational knowledge worker's decision-making and reasoning processes. The primary objective of a visual analytic system is to facilitate analytical reasoning and discovery of insights through interactive visual interfaces. It also enables the transfer of capability and expertise from where it resides to where it is needed–across individuals, and organizations as necessary.

The problem is, however, most domain analytical practices generally vary from organizations to organizations. This leads to the diversified design of visual analytics systems in incorporating domain analytical processes, making it difficult to generalize the success from one domain to another. Exacerbating this problem is the dearth of general models of analytical workflows available to enable such timely and effective designs.

To alleviate these problems, this dissertation presents a two-stage framework for informing the design of a visual analytics system. This two-stage design framework builds upon and extends current practices pertaining to analytical workflow and focuses, in particular, on investigating its effect on the design of visual analytics systems for organizational environments. It aims to empower organizations with more systematic and purposeful information analyses through modeling the domain users' reasoning processes.

The first stage in this framework is an ***Observation and Designing stage***, in which a visual analytic system is designed and implemented to abstract and encapsulate general organizational analytical processes, through extensive collaboration with domain users. The second stage is the ***User-centric Refinement stage***, which aims at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their task behavior. To implement this framework in the process of designing a visual analytics system, this dissertation proposes four general design recommendations that, when followed, empower such systems to bring the users closer to the center of their analytical processes.

This dissertation makes three primary contributions: first, it presents a general characterization of the analytical workflow in organizational environments. This characterization fills in the blank of the current lack of such an analytical model and further represents a set of domain analytical tasks that are commonly applicable to various organizations. Secondly, this dissertation describes a two-stage framework for facilitating the domain users' workflows through integrating their analytical models into interactive visual analytics systems. Finally, this dissertation presents recommendations and suggestions on enriching and refining domain analysis through capturing and analyzing knowledge workers' analysis processes.

To exemplify the generalizability of these design recommendations, this dissertation presents three visual analytics systems that are developed following the proposed recommendations, including Taste for Xerox Corporation, OpsVis for Microsoft, and IRSV for the U.S. Department of Transportation. All of these systems are deployed to domain knowledge workers and are adopted for their analytical practices. Extensive empirical evaluations are further conducted to demonstrate efficacy of these systems in facilitating domain analytical processes.

ACKNOWLEDGMENTS

This wonderful journey would not have been possible without the help and support of many individuals. I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

To begin with, I would like to express my deepest gratitude to my advisor, Dr. William Ribarsky, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. Bill sustained me with much needed encourage, and advised and funded me throughout my years in the Ph.D. program. I am deeply indebted to him, who challenged me while remaining encouraging all the way through, especially in the final months, when I needed his support the most.

I am grateful to Dr. Eric Bier and Dr. William Janssen for inviting me to pursue my research in Palo Alto Research Center, both as an intern and as a visiting research scientist. They have been such wonderful remote collaborators and discussants throughout my research. Their supports have helped me to develop my understanding about organizational environments, and led to the establishment of the proposed the design framework in this dissertation.

I am also grateful to the members of my committee who have taken the time to thoroughly and professionally guide me. Their efforts in giving me advice and technical critique have made me rethink and rewrite portions of this dissertation for the better. I would like to thank Dr. Souvenir, Dr. Kosara, and Dr. Wartell for guiding my research for the past several years and helping me to develop my background in machine learning, information visualization, and geo-spatial visualizations. Special thanks to Dr.Shen-En Chen, who let me experience and lead the research with the U.S. Department of Transportation, and financially supported my research.

I would like to thank Dr. Thomas Butkiewicz , who as a good friend, was always willing to help and give his best suggestions. It would have been a lonely lab without him. Many thanks to Dr. Remco Chang, Dr. Evan Suma, Dr. Dong Jeong, Alex Godwin, Li Yu, and other workers in the laboratory of Dr. Ribarsky's for their great friendship throughout the years.

I owe a big hug and kiss to my parents, Renxuan Liu and Zhihong Wang, who have always been remarkably supportive of my interests, despite the fact that those interests have not always been understandable or interesting to them. I would also like to thank my brother, Chen Hu, for being such a good support.

My greatest thanks, however, goes lovingly to my best friend, and fiance, Wenwen Dou, who has inspired me to grow in ways I could never have imagined. Despite my tendency to act as though visual analytics is the most important thing in life, she was able to see through to my true self, remaining passionate, patient, giving, and forgiving.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The primary objective of a visual analytic system is to facilitate analytical reasoning and the discovery of insights through interactive visual interfaces. It also enables the transfer of capability and expertise from where it resides to where it is needed–across individuals, and organizations as necessary. As the field strides forward, visual analytics research has been applied to various domains and thus lead to the development of diversified systems that are tailored to individual domain. Many of current systems have demonstrated the utility in facilitating domains analysis. For example, Wang et al. [168] incorporated the investigative journalism methodologies into an interactive visual analytics system to facilitate policy makers' investigation of global terrorism activities; Xiao et al. [174] presented a traffic analysis system to help network traffic analyst analyze cyber-attack patterns through the use of domain knowledge representation.

A perennially interesting research topic in the field of visual analytics is *how to* effectively design and develop systems that augment organizational knowledge worker's decision-making and reasoning processes. The topic becomes significant because, as visual analytics is applied to more analysis domains, the field needs to identify a general design framework that can instrument an effective system design and development, and provide researchers a basis for making assessment about visual analytics use patterns, and evaluate its impacts.

The problem is, however, most of domain analytical practices generally vary from organizations to organizations. This leads to the diversified design of visual analytics systems in incorporating domain analytical processes. With few exceptions [127, 164], the process for analysis-integration utilized in most of current visual analytics systems is often specific to the targeted domain and its analytical tasks. The lack of consensus, from knowledge workers in different domains, on general analytical tasks and workflows makes it difficult to generalize the success of one visual analytics systems to new problem domains.

Exacerbating this problem is the dearth of general recommendations to articulate the boundaries within which particular design assumptions apply. While a few models [29, 115, 161] have been created to inform the conceptual design processes for visualization, these models have not yet been developed to a point where they can provide tangible system design recommendations, nor making the design process more tractable for visual analytics developers. Thereby, they are still limited on instructing and improving visual analytics development outcomes.

To address these problems, the needed design framework must comply with two fundamental requirements: 1) it must reveal the generalizability of visual analytics as a science in encapsulating and facilitating domain analysis processes, and more importantly 2) it must clearly inform a systematical development process that guarantees the efficacy and validity of a customizable visual analytics system.

To achieve such framework, in the past three years, this research conceptualized and followed a series of research processes: it began by categorizing the design experiences gained from collaborations with various organizations into a general organizational analysis workflow. Then, validated by domain users, this research encapsulated the general workflow into a two-stage design, and listed the necessary design considerations for each stage. It further followed these considerations and developed actual visual analytics system through iterative prototyping with domain users. Through extensive empirical evaluations of the two design stages, this research finally encapsulated both stages into a general design framework, and outlined its four essential design recommendations.

This dissertation started by extending current practices pertaining to analytical workflow and focused, in particular, on investigating its dynamics to the design of visual analytics systems for organizational environments. Specifically, three extensive collaborations were conducted with organizations and groups of knowledge workers to gain insights about the general analytical tasks and workflows. The results in this dissertation are grounded on actual system developments with three groups of professionals in different organizational settings: bridge-asset managers in The U.S. Department of Transportation, who propose and execute strategic bridge maintenance plans; business analysts from Xerox, who retrieve and analyze documents for information essential to the operation of the business; and network operational manager from Microsoft, who monitors the status of physical servers and network health. The developments of visual analytics systems are carried out through close examinations of these domain users' analytic workflows, and interviews with them in learning their actions required for achieving each analytical task. As suggested in previous empirical studies [18, 20][15], the observed analysis tasks in these large organizations are representative across the similar domains and, thereby of great value for generalizing the needed analytical workflows.

Based on these extensive collaborations, a two-stage design framework is proposed in this dissertation for designing visual analytics systems. The goal for this framework is to inform the design of a visual analytics system through disseminating and incorporating the general analytical workflows into the process. In particular, as shown in Figure 1, the first stage in this framework is an ***Observation and Designing*** stage, in which a visual analytic system is designed and implemented to abstract and encapsulate general organizational analytical processes. The second stage is the ***User-centric Refinement*** stage, which aims at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their analysis processes. Details of each design components and their related design processes are described in Chapter 4.

To implement this framework in the process of designing a visual analytics system,

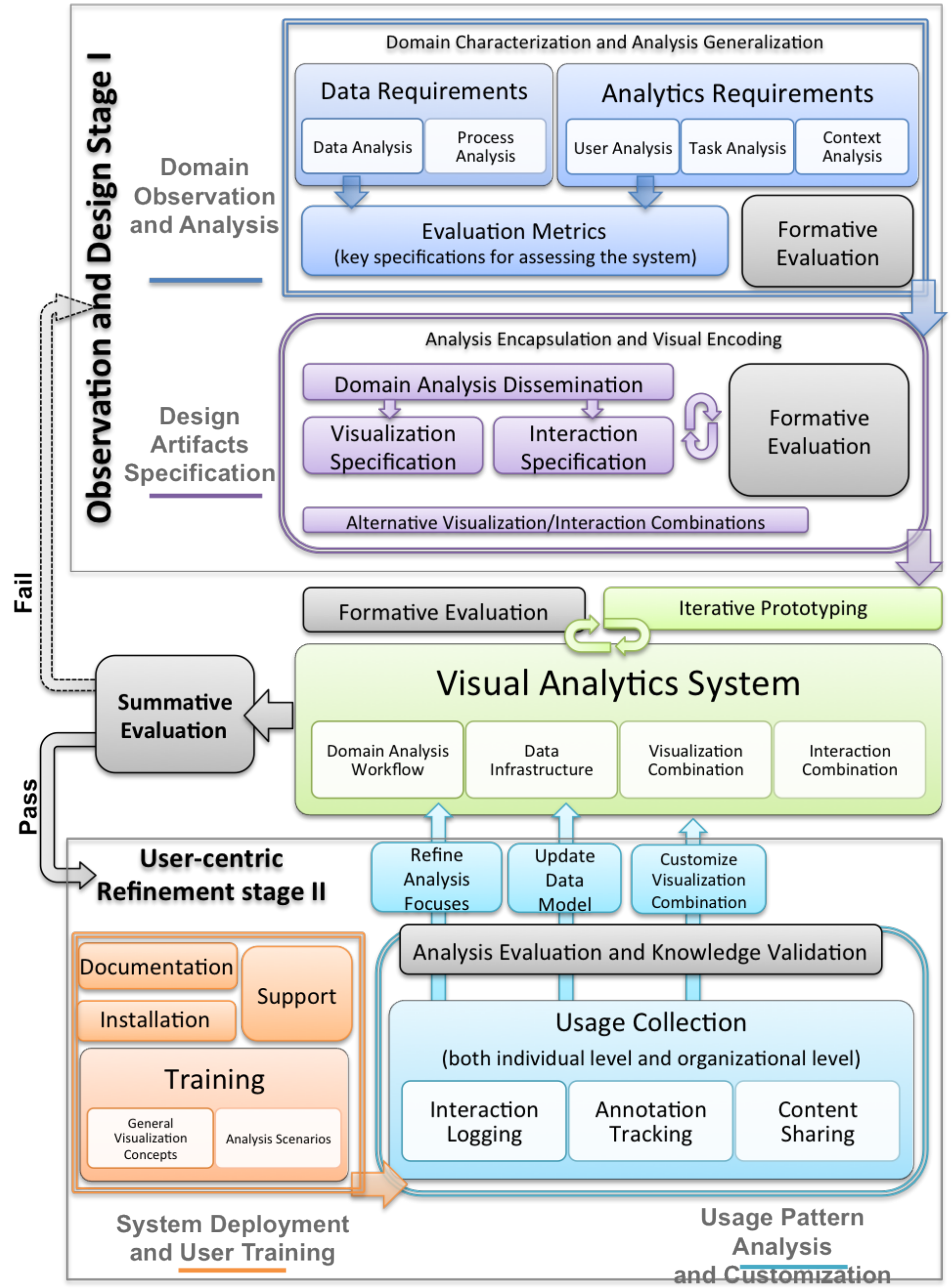


Figure 1: An overview for the two-stage design framework.

a set of four general design recommendations is suggested in this dissertation. As shown in Table 1, these recommendations are presented as a natural progression for designing a visual analytic system. From the initial communication with targeted domain users and to the prototyping and iteration of visual analytics system, these recommendations illustrate the necessary actions and recommendations to design a visual system that augments organizational analytics processes.

Table 1: The four recommendations for the two-stage design framework.

Recommendation 1	Characterize Organizational Analytics Processes Through Interactions with Domain Users
Recommendation 2	Disseminate Analytics Workflows to Key Actionable Knowledge
Recommendation 3	Design for Actionable Knowledge Transformation Through Software Prototyping
Recommendation 4	Design for Integrating individual's Analysis Practices with General Analytical Workflow

The primary contributions of this dissertation are therefore threefold: first, this dissertation presents a two-stage framework to incorporate both the general domain analytical workflow and individual analysis processes into the design of a visual analytics system. It illustrates general design considerations that, when followed, empowers a visual analytics system to bring the users closer to the center of their analytical processes.

By placing the analytical models into the center of the visual analytics design, this framework enables the domain users to directly interact with the data in real time and makes analytical decisions in an customized reasoning environment. This framework presents a general characterization of the analytical workflows in organizational environments. This characterization fills in the blank of the current lacking of such analytical model and further presents a set of domain analytical tasks that are commonly applicable to various organizations. Specifically, this work has identified six task activities essential for these professionals' decision-making workflows.

As shown in Figure 2, these six tasks are recurrent and central in jobs involving foraging and organizing relevant information, and enable these workers to update status and coordinate progress with other individuals and groups. Currently, these tasks are handled dispersedly in an individual's workflow with little support for systematically aggregating, organizing, or analyzing the desired information. This high-level semantic workflow is further disseminated into key knowledge actions, more tangible artifacts that represent the fine-grained design elements of each analytical task.

Secondly, this dissertation provides a general ground to bridge research and industry on design and development. It connects the academic research on visual analytics to industrial organizations, and showcases the utility of organizational visual analytics systems. The use of the proposed framework would not only provide the industrial collaborators concrete ideas about the utility of a visual analytics system, but also suggest practical design processes and considerations for implementing visual analytics system.

To illustrate the generalizability and effectiveness of the design considerations, three organizational visual analytics systems are introduced and evaluated in this dissertation. These systems are designed using the proposed considerations as a basis. All of the three systems are deployed to domain knowledge workers and were adopted for their analytical practices. Extensive empirical evaluations are further presented to demonstrate efficacy of these systems in facilitating domain analytical processes.

In addition, by bridging the gap between high-level design concepts and fine-grain implementation of such concepts, this dissertation provides a pragmatic view of implementing an organizational visual analytics system that can help augment organizational information analyses through modeling domain users reasoning approaches

Finally, this dissertation presents design considerations on enriching and refining domain analysis through capturing and analyzing knowledge workers' analysis processes.

The considerations are used to achieve the *user-centric analysis refinement* stage. This work presents two possible techniques to achieve in this stage—namely interaction capturing, and annotation sharing—and further discusses their utility in understanding users' analytical preferences in order to customize their analysis processes. Both the techniques have been utilized in the design of two visual analytics systems to exemplify their utility. Empirical evaluations with domain analysts has been conducted to demonstrate the efficacy of these techniques in supporting customized analytical processes.

This dissertation is structured as follows. Chapter 2 outlines the previous work which is relevant to acquiring and incorporating knowledge with visual analytics. Chapter 4 presents the two-stage design framework. Chapter 3 introduces the four design recommendations that are used in instrumenting the designs of a visual analytics system. For each recommendation, detailed examples are presented to show the means to follow the recommendations. Chapter 5 evaluates the framework and each recommendation in detail for individual collaboration. Together with the conclusions and some future research directions as described in Chapter 6, this dissertation intends to serve as a step forward in fully developing a theoretical foundation for visual analytics designs.

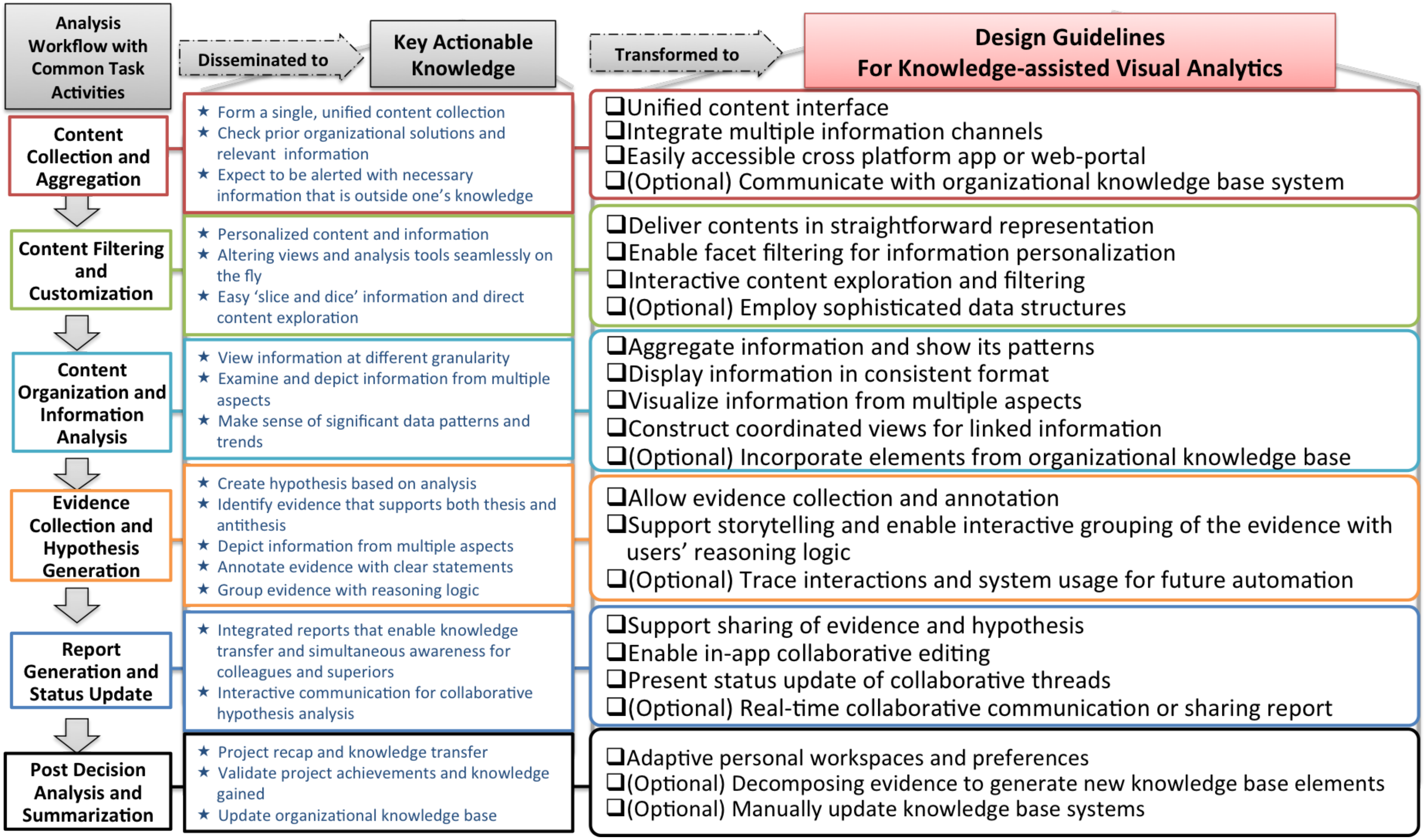


Figure 2: An overview for the tasks activities that are encapsulated in the two-stage design framework and their related design considerations. These design considerations are consolidated by transforming this actionable knowledge into practical functions.

CHAPTER 2: RELATED WORK

2.1 Objectives

The aim of this chapter is to identify a comprehensive set of theoretical information about visual analytics design. A broad analysis of relevant existing frameworks must be accomplished to serve as a foundation for the objective of the research, that is to synthesize some useful theoretical concepts for the construction of the two-stage framework for designing organizational visual analytics.

The current literature does not yet present a general framework for design visual analytics systems; and building such design framework is the main scope of this dissertation. It is postulated that there do exist in the current literature sufficient practical implementations about visual analytics design to form a solid foundation from which to synthesize useful framework. The object of this chapter is to form such foundation .

After reading this chapter, readers should understand the several existing visualization and HCI design frameworks that relevant to the construction of the visual analytics design framework. Readers should further understand the needs for designing visual analytics system for organizations.

2.2 Background

Visual Analytics is the discipline emphasizes on the facilitation of human's reasoning processes. The premise of visual analytics is concerned with the domain analysis, design, evaluation, and the implementation of interactive visualization systems. In particular, the core science of visual analytics examines the phenomena that surround the domains' analysis processes and more importantly, portraits and enhances those phenomena through encapsulating them into interactive visual interfaces. Since its inception five years ago, the field of visual analytics has been a emerged discipline that has undergone tremendous growth and recognition. It has been applied to many domains to address the need of managing the ever-growing mountain of data, and helps the domain users to make sense of data, information, and knowledge through the use of computation and visualization.

The pioneers of visual analytics—researchers joined from data analysis, information visualization and HCI—has brought with them invaluable application designs, algorithmic thinkings, and engineering traditions. Quite early on, the influence from intelligence community stressed the application of analysis model and theories of sense-making processes when designing the visual analytics systems [23][95][71]. As show in Figure 3, the sense-making loop presented by Card and Pirolli [131] provides a theoretical basis for understanding and portraying the analytical discourse that the analyst performs. Later on, parts of the cognitive science showed interests in this emerging field and introduced the field with a science approach, that is a strong belief in the value of empirical observations of analysis processes and users' performances [61].

Recently, influences from machine learning, human-computer interaction and knowledge management [167] came to establish the methodological grounds in the field of visual analytics, broadening its scope to not only design visual interfaces for individual users, but also to reveal the needs of visual analytics designs for the broader analytical processes that are taken places across individuals, groups, and organizations. Therefore, the contemporary visual analytics field is multidisciplinary to its nature, and it has been fast maturing thanks to the collective contributions from these diversified research areas.

2.3 Motivation

The core of contemporary visual analytics lies on the capturing and encapsulating of domain analysis processes in to a human-centered software design. This fundamental

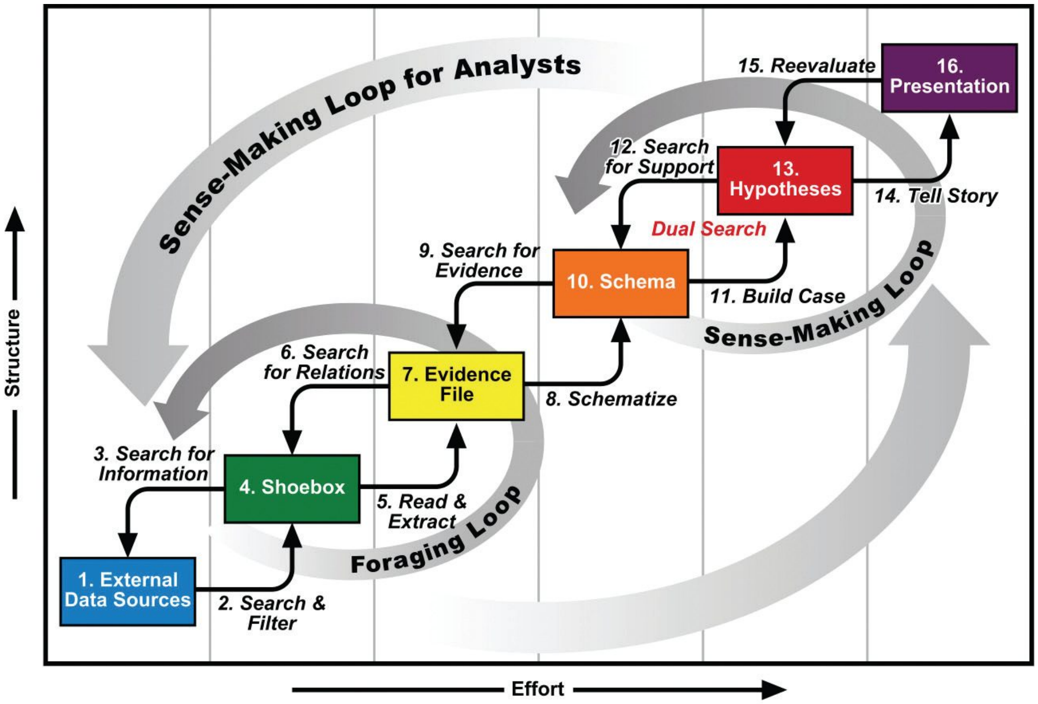


Figure 3: The sense-making loop proposed by Card and Pirolli [131].

drives the maturation of the field, and converts the researchers ideas materialize and take on concrete forms.

However, a comprehensive general model of designing a visual analytics systems for organizations is not published in the current literature. This dissertation performs a comprehensive analysis to identify the existing relevant practical implementations and design frameworks. It utilizes these set of identified research to form a solid foundation from which to synthesize useful framework (see Chapter 4).

In pursuing the successful design of a visual analytics system, this dissertation started by performing a thorough scan of the existing models.

2.4 Design Framework in Visualization and HCI

Many researchers have studied the design framework in related areas to visual analytics. These research efforts can be categorized into three groups, data-centered design model, process driven design mode and system capability-driven approach. All these categories have resulted in successful designs of visual analytics, and are effective in characterizing visualization operations.

2.4.1 Data Driven Visualization Design

In the data drive visualization design, the researchers focuses on accommodating the nature of the data. Their emphasizes the utilization of mathematical and statistical methods in deducting the informations embedded in a dataset. One of the earliest practitioner, Jacques Bertin had noted that on the understanding of deduction of relationships is a matter of permutation [12]. Bertin further proposed a synoptic that differentiated between ordered, reorderable, and topographic data, established retinal qualities of marks that would allow viewers to differentiate between marks, and provided recommendations for representing data as arrays of marks, histograms, and curves based on its dimensionality.

John Tukey, in addition, developed several methods known collectively as exploratory data analysis [157]. Tukey was interested in using statistics to extract potentially useful hypotheses from data, as opposed to confirming existing proposed hypotheses. To accomplish these goals, he introduced quantitative methods to reduce the effect of outliers, such as resistant lines and median polish analysis, and visual techniques such as box plots, rootograms, and Pareto charts that emphasize summary statistics and enumerate potential root causes of phenomena.

2.4.2 Process-based design

Van Wijk [161] had tried to answer the question how the value of visualization can be assessed using a process . He considered visualization purely from a technological point of view, aiming for effectiveness and efficiency. This requires that costs and benefits are assessed. The simple model proposed enables us to get insight in various aspects of visualization, and also to understand why certain classes of methods have success and others not. Another view is to consider visualization as an art, i.e., something that is interesting enough for its own sake, and finally a view on visualization as an empiric science was dis- cussed. Finally, He considers that each view that is adopted does imply playing a different game, and if we want to win, we should play those games according their own rules: aim for provable effectiveness and efficiency, aim for elegance and beauty, and aim at generic laws with predictive power.

Ming et al. [30] in their position paper on knowledge assisted visual analytics system had proposed an high-level design pipeline. This pipeline focuses on utilizing visualization to help application users to transfer data in the computational space to information and knowledge in the perceptual and cognitive space. As a discipline, they suggests the need for visual analytics infrastructures to support the development of about visualization, and to transfer such data to information and knowledge, which helps further our understanding as well as enhance the visualization technology. While this pipeline provided a clear conceptual design direction for visual analytics systems, it has been too general and high-level to be informative for actual system development.

Munzner [115] proposed a nested model in designing visualization system. In this model, Munzner focused on the use of validation to guide the visualization designers to navigate through the design processes. She presented the nested model that classifies validation methodologies for use at only one of four separate levels, in a unified approach to visualization design and evaluation. While it is not directly targeted at addressing challenges in visual analytics, this model has quite influence to the development of this dissertation.

2.4.3 System Capability-driven Design

Card, Pirolli and colleagues have done work in understanding analyst sensemaking techniques using cognitive task analysis techniques [131]. This work posits interlocked, bidirectional information foraging and sensemaking loops, and describes high-level tasks done in going both from theory to data as well as data to theory.

Chi et al. [32] extended the Card reference model into a Data State Reference

Model [24] in order to isolate common operational steps within each visualization type. Chi had the explicit aims of assisting implementers with choosing and deploying visualization techniques and broadening the visualization design space. In addtion, Chuah et al. [33] present frameworks for organizing the different types of interactions within a visualization. Both Chi and Chuah et al. organized the interactions by their end effects (e.g., whether the value (data), view (graphics), or some combination is affected). Chi discussesed implementation only briefly, pointing out that where the operator would be optimally placed (within the visualization, the database, or in a specialized tool) depends on where in the visualization pipeline the interaction falls. They further taxonomizes existing visualization techniques into several data categories (scientific visualization, geographic visaulizations, multi-dimensional, information landscape).

Shneiderman [143] posits a task-by-data-type taxonomy that crosses information-seeking visualization tasks with different types of data and discusses both examples and missed opportunities for supporting the given tasks. The taxonomy assumes an implicit mapping between user goals and these visualization tasks. Snap-Together Visualizations [120] focus on how to layer user interaction on top of visualizations for coordination. While they focus on interaction rather than visualization (leaving that to the individual visualization tools snapped together using their interface), they have a very different model in which every visualization that is snapped together is done so via the equivalent of a database join. This model leads to easily achieving some powerful interaction capabilities such as brushing and linking.

As a result of these interdisciplinary research collaborations, the science and techniques of design visualization have reached a matured status. There is a significant amount of ongoing development currently in information-assisted visualization, such as UrbanVis [27] and Terrain Analysis [22]. With a large amount of information collected locally and globally, it is inevitable that there will be a transition to knowledge- assisted visualization. have been applied and well-received by many analytical domains.

In summary, while all these approaches had demonstrated influences in visual analytics designs, they are often too general to be informative to actual system designs. More over, the core foundation of visual analytics lies on the incorporation and customization for analysis processes, which are not explicitly supported by these frameworks. Therefore, the primary goal for this dissertations is to develop a comprehensive general model of designing a visual analytics systems for organizations.

CHAPTER 3: DESIGN RECOMMENDATIONS AND CASE STUDIES

3.1 Objectives

This chapter presents four recommendations for designing a visual analytics system for an organizational environment. These four recommendations, as presented in Table 2, follow a natural progression for designing a visual analytic system. From the initial communication with targeted domain users and to the prototyping and iteration of visual analytics system, these recommendations illustrate the necessary actions to design a visual system that augments organizational analytics processes. This chapter further presents three successful visual analytics systems to demonstrate the utility of these recommendations. Each of the four recommendations is validated by the corresponding actions taken in the design of actual visual analytics systems.

This chapter:

- Illustrates four recommendations in designing a visual analytics system for organizational environment.
- Validates all four recommendations through actual design practices.
- Presents three actual visual analytics designs and their correlations to the recommendations.

3.2 Overview

In this chapter, the dissertation describes a set of four design and implementation recommendations for developing organizational visual analytics systems. These recommendations are concluded based on several long-term collaborations with multiple organizations, including Microsoft, Xerox PARC and the US Department of Transportation.

All four recommendations are used to present the progression from the design of individual visual analytics system to the conclusion of a general design framework for the visual analytics field. By describing the evolution of these recommendations in the context of the actual design practice, the research intends to capture and present the richness of the design framework in a manner that mere verbal descriptions cannot achieve. In addition, details about how the development of a visual analytics system are provided to exemplify the design recommendations. While these design recommendations focus on the organizational environment, nonetheless, the recommendations presented in this dissertation presents a general design strategy for a visual analytics system to follow the two-stage design framework (see Chapter 4).

Table 2: The overview for the design recommendations.

Recommendations	**Actions**
Recommendation 1	Characterize Organizational Analytics Processes Through Interactions with Domain Users
Recommendation 2	Disseminate Analytics Workflows to Key Actionable Knowledge
Recommendation 3	Design for Actionable Knowledge Transformation Through Software Prototyping
Recommendation 4	Design for Integrating individual's Analysis Practices with General Analytical Workflow

3.3 Recommendation 1: Characterize Organizational Analysis Processes through Interactions with Domain Users

The analysis process adopted by individual knowledge workers in an organizational environment is quite representative to inform their domain knowledge. Thus, identifying a detailed portrait of domain users and their analytical workflows are the most important design recommendation.

However, as presented in previous research [118] [36], an organizational analytical task is a process of handling multiple channels of information through the utilization of trained knowledge and current resources. Characterizing the analytical process in an organizational setting, such as a company or a governmental agency, is a complex process:

- The characterized analytical process must support the needs and practice of a known user group. This requires a long term collaboration and communication between the visual analytics system designer and the end-users. Sometimes, identifying such analysis processes could be difficult due to the lack of general recommendations.
- The characterized analytical process must incorporate the domain experts' knowledge about their analytical workflows. This is challenging since these experts may not lay clear workflows and do not follow prescription for action. Domain analysts are used to perform analytical reasoning in their own way, making the externalizing their workflow difficult.
- The characterized analytical process must represent the prescribed process and restrict ad hoc analytical processes. This requires a long-term close collaboration and commitment from both the visual analytics designer and participating organizations.

All these challenges are exacerbated by the lack of generalized models that can indicate the representative analytical workflows in an organization. Much of the current research on designing visual analytics systems for organizations are principally by trial and error, making it difficult to reuse and generalize the design approaches applied in different domains.

Therefore, the first recommendation in this design framework focuses on the generalization of the analytical workflows in organizational environments. This recommendation emphasizes the use of interactions to engage the end-users in the process of the development of a visual analytics system. By placing domain analysts in the center of the design process, this recommendation aims to address the users' lack of incentives to use a new system, and focuses on motivating them to adopt advanced analytical tools and practices.

In addition, this recommendation presents a baseline for visual analytics designers to systematically examine each organizational environment, and provides consistent methods to approach the domain knowledge workers. It provides information for several design considerations, such as what does the domain knowledge work user generally know; what do they need to know; and what they probably do not know yet, but want to know? How do they normally perform domain analysis? And what could be the implications for their requirements of a visual analytics systems?

Drawing from previous information system design theories [109], this recommendation is centered on domain process analysis. It regards the understanding of domain analytical process as a two-step approach. The first step involves the actions that conduct extensive studies with domain users to learn the answers to the above design questions. This provides guidance for charactering the general analytical process. This step follows the process-redesign analysis that is customary in reengineering engagements, as can be seen in the theory proposed by Hammer et al. [64]. The involved actions represent the high-level task activities in each individual analytical domains; and they are summarized and presented in the horizontal analytical workflows, as a direct diagram illustrated in Figure 7.

The second step involves generalizing the high-level task activities of the analytical process. Such generalization focuses on cross-process analytical tasks that are commonly applicable to multiple domains. The latter process representations are also generally presented as flow diagram. But the corresponding process activities run vertically down in Figure 7, across the kinds of horizontal analytical-process examined for each organizational environment.

The remaining of this section introduces the analytical characterizations for the targeted organizations. It details the interview process and analysis methodologies used to specify the task activities for each domain. It further concludes a cross-domain general analytical workflows through correlating and comparing these individual task activities.

3.3.1 Action: Characterizing General Organizational Analytics Processes

To produce appropriate design recommendations for an effective visual analytics system, this research is conducted through extensive collaboration with three large organizations and groups of knowledge workers. The targeted domain users are knowledge workers from various analytical domains, including bridge-asset managers in the U.S. Department of Transportation, who propose and execute strategic bridge maintenance plans; business analysts from Xerox, who retrieve and analyze documents for information essential to the operation of the business; and network operational manager from Microsoft, who monitors the status of physical servers and network health.

These domain users' devotion and generosity are of great contribution to the establishment of this design recommendation. All professionals from the above three organizations granted the opportunity for close, in-depth interactions with their knowledge workers and to conduct surveys and interviews, which were crucial in studying their analytic processes. With the collected inputs, deduct invaluable resources and insights were deducted to create schematics detailing their workflows, and identify the general domain analytical processes used across all the organizations.

The characterization of the general domain analytics process is carried out through close examination of these users' analytic workflows, and interviews with them in learning their knowledge work required for achieving each analytical task. Interview with representatives from the three organizations revealed the analytical needs for the potential users, including their focuses on fusing multiple streams of data, retrieving information for context-dependent tasks, analyzing and sharing their findings, and finally collaborating with others to reach business decisions. In addition, members from all three groups are required to generate shared products effectively (e.g., a maintenance proposal or analytical report). Subsequently, these professionals need to coordinate with multiple colleagues in different locations to agree on strategic decisions.

This recommendation is concluded based on three separate investigations with knowledge workers from each of the three organizations. Participants varied in number, depending on the availability of these busy professionals at each time. During each investigation, data was collected using online questionnaires and/or semi-structured interviews. The data collected was used to characterize these workers' task activities within analytical processes, and further used to develop the design requirements for a visual analytics system.

In the following sections, the procedures and results are described for each investigation. This work will further summarize the general analytical workflow and its six significant analytics tasks in Section 3.3.2.

3.3.1.1 Depicting Tasks in Bridge Maintenance Process

Background and Domain Analysis

Bridge maintenance workflow is a process of deciding the severity, trending, relevance, and benefits of maintenance work on a specific bridge as well as a network of bridges. According to AASHTO's asset management recommendations [2], the first step in this process is to gather relevant data about a particular bridge, including its known damages, previous maintenance histories, and typical deterioration patterns. Bridge managers will then start analyzing the obtained information, identifying the needs for maintenance and coming up with proper maintenance plans.

According to bridge managers from NCDOT, it is common for a bridge manager to be responsible for hundreds of bridges. Since the federal recommendation dictates that bridges are inspected on a biennial basis, approximately 50% of the bridges are inspected in a given year. However, in that same year, only a portion of the bridges, approximately 20% - 25%, would require any maintenance attention. Even fewer bridges (around 10%) may actually receive maintenance work. Given the complexity of these inspection results, compounded with external constraints on budget and resources, a bridge manager needs to have complete understanding of all bridges under his/her jurisdiction when making maintenance decisions.

It is therefore necessary to have a bridge management system (BMS) that monitors and analyzes the conditions of bridges in a way that allows a bridge manager to maintain an overview of all bridges and yet retain the capability to inspect detailed information of a particular bridge. Currently, there are a few available BMSs such as Pontis[163] and BRIDGIT[66] that promise analytical capabilities. However, there exist many limitations and issues with these BMSs (some of which will be described in detail in the following section), many bridge managers, including a few from NCDOT, still rely on using simple spreadsheets such as Microsoft Excel to perform their analyses.

Identifying Domain Analytical Workflow and Limits

While these strategies have largely balanced the limited resources with the upkeep of bridges across the country, the collapse of the I-35 bridge in Minneapolis during August 2007 serves as a devastating reminder that the complexity of bridge management still demands novel techniques and proper tools to interpret and understand bridge data.

Starting in January 2008, a research partnership with the USDOT and The North Carolina State Department of Transportation (NCDOT) to investigate novel approaches in assisting the bridge management process. One of the first actions under this research partnership was to conduct a nation-wide survey [165] regarding professional profiles, tool usage, and tool preferences. The surveys were designed to provide a baseline and statistics for comparisons between normal tools used in bridge management, and to identify potential areas for improvement.

This survey focused on collecting information about the utilization of BMS in each state, and asked for feedback on the utilities of the existing systems. As listed in the following table 3, our survey was centered around these three substantive questions:

Table 3: The three substantive questions to all the state DOTs in the U.S.

Questions	**Details**
Question 1	What do you see as the most important next step in the further development of your agency's BMS?
Question 2	What do you see as the necessity of expanding current BMSs?
Question 3	What are the biggest barriers in your department in implementing innovations that may strengthen your BMS?

Thirty-five out of the fifty state DOTs responded to this survey. The results clearly indicated that current bridge management systems are often insufficient in supporting effective bridge analysis. Almost all the responding states expressed the need to have a management system that would enable them to be more effective at analyzing their bridges, and that such a system needs to be customizable to assist their individual workflows.

Based on the response of these state DOTs, the major drawbacks of existing BMSs in supporting domain analytical processes can be categorized in the following three areas. In general, these areas includes challenges on: the insufficient support for analytical processes, the restrictions in personalizing analysis routines, and the difficulties in integrating heterogenous data.

- **BMSs have not provided effective support for bridge managers' decision-making processes.** Many states have reported that they mainly utilize BMSs as data storage software. Although some BMSs have certain automatic decision making support capability [163], their analysis tools are not appropriate or adequate to be incorporated in to a bridge manager's analysis process.

- **BMSs are rigid in structure and cannot be easily adapted to support individual bridge manager's task routines.** Many states have reported that it is difficult for them to customize BMSs to suite their own analytical approaches. These states have also indicated that it is very difficult for them to implement additional features within these BMSs.

- **BMSs have not provided abilities to incorporate local inspection technologies.** Many states have their own inspections results that are complementary to the national standard inspections. However, as reported by state DOTs, it is often difficult to import such information into the data structure that these BMSs provide.

Based on this categorization, a series of semi-structured interviews were further conducted with bridge managers on a regular basis (every two weeks), in order to iteratively identify and propose features that can better support their analyses. Through these interviews, it becomes clear that bridge maintenance workflow is a process of deciding the severity, trending, relevance, and benefits of maintenance work on specific bridges, as well across as entire networks of bridges. Bridge managers hold the role of knowledge manager and are attuned to information analysis and sharing practices.

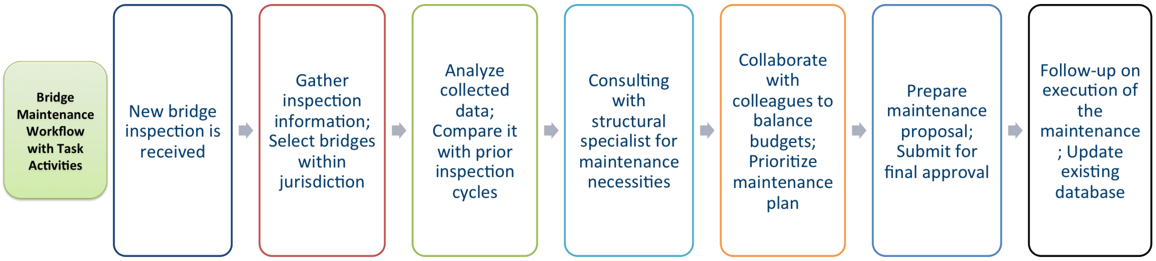


Figure 4: A typical analytical workflow for bridge maintenance planning.

As shown in Figure 4, the first essential analytical task in the bridge analysis process is to gather all the relevant data about a particular bridge, including any known damage, previous maintenance history, and typical deterioration patterns of the materials involved. Bridge managers then analyze the obtained information, identify any need for maintenance, and write up proper maintenance plans. It is also noted that bridge managers often need to develop their own custom analysis routines. Depending on available resources, a bridge manager's strategy can be very different from their peers', requiring a different combination of the above analysis processes. In addition, sometimes even a single manager needs to utilize multiple alternative analytical approaches due to changes in priorities. At the heart of these individual routines are different combinations and sequences of the above analytical processes. Therefore, it is important for a visual analytics system to provide bridge managers with the flexibility to combine and sequence these analytical processes to fit their own, customized workflows.

Several followup interviews had been conducted with state DOTs to understand the limitations caused by these shortcomings. These interviews were also used to further identify possible analysis focuses that are essential to support the domain analytical workflow. In general, there are three analyses that are often crucial to the bridge maintenance planning: structural analysis, temporal analysis, and geospatial analysis. As shown in the following paragraphs, these analysis focuses are of great importance in facilitating bridge managers to assess bridge conditions from multiple perspectives, and therefore are integral to their daily workflows:

- **Dynamic Geospatial Analysis**: Bridges exist in a dynamic environment with changing surroundings. Therefore, rather than using a static map, bridge managers often need to adapt to new situations and analyze bridges with additional information such as traffic patterns, flooding regions, and population densities. According to bridge managers, supporting dynamic geo-exploration is a primary area for bridge analysis.

- **High Dimensional Structural Analysis:** Typically, the data representing bridge structures are high in dimensionality. Federal regulation requires bridge inspection to record nearly 130 structural variables biennially. Given the complexity of the data, a tool that could assist bridge managers' comprehension of these variables would be essential. Specifically, on a high level, bridge managers need to detect and identify causal relationships and trends in these variables so that they could identify phenomena that are affecting all bridges. On a detailed level when inspecting a single bridge, bridge managers need to examine the overall structure integrity of a bridge across multiple variables and to focus on particular structural components inside that bridge.

- **Scalable Temporal Analysis:** Through analyzing the temporal changes of a bridge's condition, bridge managers can compute the deterioration rate of the bridge. In addition, bridge managers can adjust the future maintenance plans by assessing the outcomes from previous work. Therefore, the ability to capture the temporal information is of great value to bridge managers when planning for future maintenances. However, temporal analysis in most existing BMSs is limited to analysis on a per bridge basis. Having an overview that could help the bridge managers spot bridges with abnormal temporal behaviors would be very beneficial.

In summary, the detailed characterization of the bridge maintenance workflow (see Figure 4) is critical in understanding the domain analytical processes and engaging the incentives from domain users. The primary goal of a visual analytics system is therefore to address these challenges in accordance with the needs of the bridge managers at these state DOTs.

3.3.1.2 Identifying Analytical Processes in Cloud Service Management

Along with the investigation on analytical processes in bridge-asset management, collaboration with Microsoft Cloud Service department introduces another rich organizational environment for this research to enrich the analysis of general domain analytical workflow. In monitoring cloud services, a central problem is to identify anomalies and problems, in the face of the high degree of replication and the high degree of natural variability in the workloads. In talking to service developers and operators, a huge hunger for perspective in identifying anomalies and problems across distributed systems, and found that perspective comes largely from correlating across the highly replicated structure of these services. Averages were deemed meaningless, variations were valued. Therefore, these busy professionals are seeking tools that provide effective data analyses and help them rapidly gain insights from the deluge of monitored network data.

Background and Domain Analysis

At Microsoft, several hundred thousand of servers live in data centers around the world, making up several hundred different cloud services, ranging in size from very large systems with tens of thousands of servers (such as Live Search and Hotmail) to far smaller systems with a few dozen servers. Microsoft online services operations have been running web services since the mid-1990s, and have had to incorporate many different design philosophies, architectures, and even operating systems over that time. While the many systems share their physical data centers, and some first-responder support personnel, each system has its own topology, design, and dependencies, and as a result requires considerable expertise to maintain.

Expectations for consistent performance and availability of cloud services are high and getting higher. Yet, within a data center, even under normal operating conditions, given the scale and complexity of the hardware and software, hundreds of hardware or software components may be in various degraded states: failing, undergoing upgrade, or failed. Typically, these problems do not impact performance or availability, as seen from a user's perspective: the services build in replication and resilience to cope with these conditions. That said, inevitably, things can and do go wrong; for example, an unanticipated dependency that leads to a significant service outage, or a situation where too many critical components fail. Operators and developers need tools to proactively identify looming problems, to localize and diagnose problems that arise in the field, and to assure unanticipated failures are not triggered during service upgrade (during which time the system is particularly vulnerable). Today, the operators of these systems have ready access to enormous lists or tree-views of individual components, with a blizzard of configuration and usage data available for viewing behind each component. In addition, new service features, and corresponding new sets of logging features are born every day. Operators are not lacking in data about their cloud services. However, and unfortunately, they are often lacking in the ability to rapidly gain insight from the deluge of available data.

In monitoring cloud services, a central problem is to identify anomalies and problems, in the face of the high degree of replication and the high degree of natural variability in the workloads. It may be that one server in a cluster is running slow: perhaps its disk is failing, and disk seeks are being retried. Perhaps a set of databases are overloaded: specific content may have suddenly become extremely popular. Perhaps, owing to aberrations in the workload, computational loads within a cluster are shooting up and response times are creeping up: it could be that the complexity of answering the individual requests has increased. Perhaps all machines in one specific cluster are dropping network packets: it might be a result of workload shift or something totally unrelated, for example, an ongoing update of network switches.

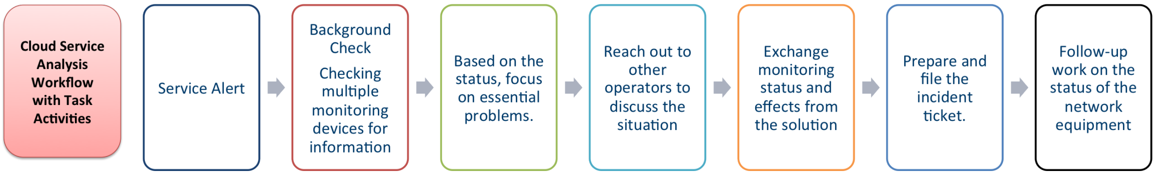

Identifying Domain Analytical Workflow and Limits

Figure 5: A typical analytical workflow for cloud service management.

To better portray the complex analytical processes, several on-site interviews and discussions were conducted to observe the day-to-day operations performed by the back-end cloud service teams. During this collaboration, the main contact team is the Address Book Clearing House, or ABCH, a medium-sized cloud service, which stores users' address books and presence information. It maintains several hundred back-end databases, and around one hundred front-end servers, that service requests for users' address books from web-based email, instant messaging tools, and other sources. ABCH, like other services at Microsoft, uses its own set of tools for tracking its status; an additional set of very general tools give information about the data center as a whole.

The results from these interviews concluded that, while these busy professionals thought of the system in terms of connections and clusters, their analytical tools were more limited. They would examine one custom tool to check one machine's details, switch to another for its connections, and query a database for its status. Difficult as it was to track services on one machine, it was even harder to move from one role (e.g., front-end) to another (e.g., load balancer), or to separate out clusters from each other. This caused the team to talk about their systems mainly independently: a discussion of the front-end would have little discussion of the back-end, for instance. Moreover, high-value cloud services are often built by combining together other cloud services. This means that the most important services do not have a single tool from which the status of the entire service can be viewed. Instead, the separate tools for each of the components must be sequentially viewed and the information from them manually integrated and interpreted by the operator. Details of their workflow is listed in Figure 5.

For each on-site visits, group members from ABCH demonstrated one "over-the-shoulder" debugging sessions as they worked through recent crisis. In doing so, these knowledge workers explain details about their analysis process, including what the alert would be, how they respond to each alert, and when to shift from one maintenance stage to another. They further provide details schematics for their system designs and explained the organizations of their data and analysis needs. Emails and trouble tickets from system failure were further collected to understand details about their problem-solving process. Based on these scenarios, the most important problems with this ecosystem of tools are:

- **Scalability**

  Challenge: Many of the tools built for enterprise applications use mechanisms such as tree-views for listing the servers, and these mechanisms become unusable with the thousands of servers in cloud services.

  Area of Improvement: There must be accommodation for a large number of servers and databases. Users should have the ability to find problem spots and zoom in rapidly, or pull all the way out for an overview.

- **Single Perspective:**

  Challenge: The "tool per component" model forces operators to coalesce and correlate information, resulting in extra steps for the operator as they switch between tools and reducing their ability to achieve situation awareness.

  Area of Improvement: A support of unified representation is required. All the information need to be coordinated to support a single and comprehensive data analysis.

- **Rigidity**

  Challenge: Current tools typically have the architecture of the service deeply embedded into them, such that a change to the architecture of the service requires massive changes to the tool. Similarly, choosing to expose new types of information in the tools can require a substantial rewriting.

Area of Improvement: Flexibility must be accommodated to different users' needs, and different services' requirements. A system should accommodate both of these variances, and be generally agnostic to the network structure. Modular components are also necessary to support diversified configuration for individual data.

- **Topology**

  Challenge: Most tools represent nodes in lists, and have minimal or no way of representing or separating by network topology or other clusters.

  Area of Improvement: A service's topology will change over time. Rather than forcing the user to specify details about the topology, the user should instead provide information on where the data can be found, and the system should present the results.

- **Monitoring overhead**

  Challenge: Many tools come with their own monitoring infrastructure. If a single server is to be monitored by several tools, it will often require several monitoring applications installed on it, at substantial cost in processor cycles and network traffic.

  Area of Improvement: Rather than implementing individual monitoring itself, the system may take advantage of the logs collected by other existing applications. These data sources may produce data at different intervals and in different ways.

- **Inconsistent data**

  Challenge: Each monitoring tool operates in its own way and has its own idiosyncrasies. Existing tools do not accommodate occasional out of range or non-compliant data. Further, experience shows that sources of meta-data may be inconsistent with each other (e.g., assigning the same server to two different roles or clusters). Today's tools do not help operators cope with, identify, or resolve these discrepancies.

Area of Improvement: Each monitoring tool operates in its own way and has its own idiosyncrasies. The system should accommodate occasional out of range or non-compliant data. The system needs to handle the inconsistency of certain database.

- **Context loss**

  Challenge: Users are forced to switch among multiple tools, and must maintain the context of the jobs they are working on (e.g., the name of the server they are modifying) using ad hoc methods (e.g., using the copy/paste clipboard).

  Area of Improvement: The system needs to handle the variety of data in a contextually consistent manner. The data comes in a variety of types, including value data, categorical data, and free text. These need to be handled at low levels, and summarized at a high level

In summary, working with ABCH cloud service team provided insights about how a single cloud service team creates, maintains their network assets. It indicated the mental model and analytical workflow used in such maintenance process. This characterization helped to establish guiding requirements for the design of a visual analytics system, detailing the challenges and area of improvements. The designed system should therefore emphasize on addressing these domain challenges and on incorporating the necessary areas of improvements. Details for the design of this system can be seen in Section 3.5.2.2.

3.3.1.3 Understanding Business Information Analysis

To further investigate analytical workflows in organizational environment, a collaborative project was established with Xerox Corporation in the summer of 2009. In an organizational environment, such as Xerox, employees' document-centric activities result in the creation of many diverse information streams, including email threads, calendar entries, web browsing histories, and versions of office documents. Many of these documents contain information essential to the operation of the business, such as project proposals and emails capturing product discussions. Thus, the goal in this project is to investigate the general analysis methodologies used in such a large organization, compare that analytical workflow with the ones that are previously observed in DOT and MSR, and finally design a system that is effective to assist corporate employees in both managing these information streams, and extracting desired business information from them.

Background and Domain Analysis

In the enterprise environment, employees' document activities result in the creation of many information streams, including email threads, calendar entries, Web browsing histories, and versions of office documents. Many of these documents contain information essential to the operation of the business, such as project proposals and emails capturing product discussions. Thus, it is crucial for enterprise employees to have an effective means to manage these information streams and retrieve desired business information from them.

However, due to the dynamic and diverse nature of document activities, finding the desired information can be an exhausting task. Recent reports from Interactive Data Corp. (IDC) show that employees typically spend 3.7 hours per week searching but not finding information, and 2.5 hours per week recreating content that couldn't be found [37].

One challenge in finding such information is coping with its diversity. Finding desired information may require using different tools or looking in different places as a result of the different behaviors and conventions of the many desktop applications, such as individual office document suites and email clients. This challenge becomes even more pronounced in projects that take place over long periods of time or that involve many people; both of these factors tend to increase the amount of information to be managed and the number of places in which the information is stored. It is observed that the complexity of finding such essential information can result in reduced productivity in an enterprise environment [37]. Therefore, there appears to be an urgent need for an information management system that can facilitate information search and retrieval in the enterprise environment.

Some commercial products, such as Google Desktop [57] and Apple Spotlight [9] have built-in document indexing that enable users to search for information with keywords. However, keyword search can be difficult and is often insufficient in an enterprise environment [151]. For example, if one can't remember the name of a document or can't think of any distinctive words or phrases in a document, a keyword-based search can be doomed to failure. In addition, if one only remembers a vague time frame of the occurrence of certain document, then searching through temporally-sorted results can be unacceptably slow.

Alternatively, as described in Thomson et al.'s [43] theory, people remember and recall things through associations with other clues. A recent observatory study by Teevan et al. [151] supports the theory that, instead of trying to directly locate the targeted information by keywords, people usually follow a chain of clues in finding the desired information. For example, users may not remember the particular title of a proposal or any text in it, but can find that proposal through some retrieval cues, such as, the person they have communicated with, the application they used, and even the rough time frame. In practice, the Feldspar system [28], has demonstrated the effectiveness of utilizing such retrieval strategy in managing document activities. Yet, research in applying such retrieval strategies in an enterprise environment is still in a preliminary stage.

Identifying Domain Analytical Workflow and Limits

The collaborative work with Xerox PARC is an investigation of some next steps in designing system that facilitate the document analysis. This project focus on identifying the specific analytical workflows that are typically used in employees' document retrieval and analysis activities.

To understand this particular information analysis process, 30 semi-structured interviews were conducted with Xerox employees. The interviewees held a broad range of positions, including product researchers who needed to write proposals and research papers, managers who were in charge of business planning and marketing, and administrative staff members who oversee hiring. These interviews were designed to provide us with baseline statistics about the general information analysis methods that were being used in managing business information.

As shown in Table 4, these interviews focused on examining three substantive questions:

Table 4: The three substantive questions for interviewees about their information analysis tasks.

Question 1	How often do you need to search for document activity information?
Question 2	What are the typical approaches you use in searching for desired information? Including both clues and applications.
Question 3	How well do the applications that you use support your retrieval tasks?

The results of these interviews showed that the most challenging problems for the corporate employees was handling large amounts of content and, more importantly, managing information from multiple channels simultaneously. In addition, the results add support to previous research (see Section 3.3.1.3) that suggests that, while the some keyword-based approaches are used in workspaces, these applications are less tailored to support information seeking processes in enterprise environments. Many of the interviewees mentioned that they usually find it hard to describe things they want to find with keywords; some of them also described that often the keywords they used were not precise enough for such search tools to have useful results. For example, one interviewee asked “How can I tell the search bar that, even though I don’t remember the title of a document I am trying to find, I do remember I have copied it to Alice in an email two weeks ago?” During these interviews, many employees expressed similar needs: they want tools that can best support their reasoning process and facilitate them to efficiently find useful information.

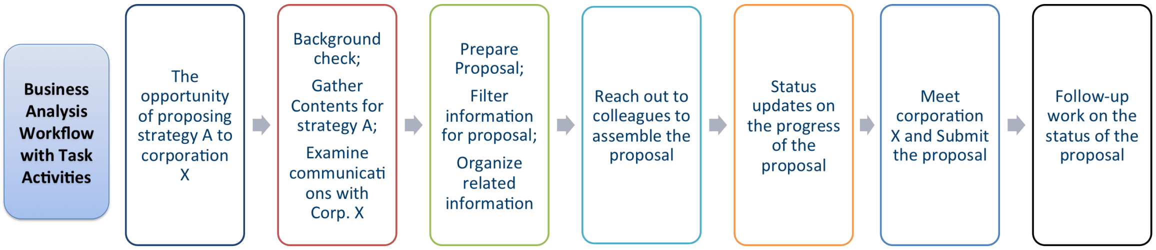


Figure 6: A typical analytical workflow for a business information analyst.

As shown in Figure 6, the analytical tasks of finding business information often include content aggregation, information organization and correlation, and sharing and collaboration. To analyze certain business information, an employee often starts with aggregating content, such as possibly relevant documents, into a single location. They will then filter this large collection of data, and attempt to organize it in a clear and consistent manner to support the awareness and sense-making process. It is also noticed that sharing their analysis findings and providing status updates are crucial activities in these employees’ workflows. Because most current tools lack support for these critical functions, employees will often resort to paper formats or email to communicate with other colleagues about the business information which they have found or their need for help finding it.

During the interviews, many employees expressed similar frustrations: they put in the effort but could not find any useful information. From these interviews, we concluded that there is need for a tool that could enable users to be more effective at analyzing their activities and retrieving desired information.

Identifying three Essential Retrieval Cues

Many interviewees further provided information about how they recall and search for document activity information in their daily analytical workflows. Combining their practical experiences and a review of the prior research literature [28, 43, 151], this research categorized the retrieval cues into three major categories.

Temporal hints contain significant information relating to document activities. Specifically, the interviewees mentioned that both the exact date and time when events occurred can be very helpful (e.g., when documents were received, read, created, or modified), and also the relative sequences of events.

Content keywords (e.g. the title of a document or the name of a person or company) are often used in filtering down to the areas of interests during the initial stage of the retrieval process. In addition, it is observed that certain keywords can also help employees to associate and connect document activities together [28], and therefore led them to recall events more precisely.

Document Types or Particular Applications are also considered important clues for employees to locate desired information [123]. Many employees report that the first thing they think of in searching for document activities is often the applications they used. In addition, many interviewees mentioned that these three retrieval cues generally coexist during their search for the desired information. Therefore, it is advantageous to integrate them into one system and present the cues to users cohesively.

All of these retrieval cues are essential in conducting the analysis tasks, and therefore they are incorporated in the identification of these employees analytical workflows (See Figure 6).

In summary, the characterization of the business analysts' workflow indicated the needs for an integrated, efficient, information retrieval tool tuned to the enterprise environment. In response to such needs for, the designed interactive visual analytics system should aim to enhance corporate employee capabilities for finding and sharing the business information that is embedded in their daily document activities. Details of the design are described in a later section (see Section 3.5.2.1).

3.3.2 Six Common Task Activities in Organizational Analytics Processes

As mentioned at the beginning of this chapter (see Chapter 3.3), this first design recommendation provides suggestions on two main areas: it first emphasizes presenting a baseline for visual analytics designs to perform systematical examination of each organizational environments, and provides consistent methods to approach the domain knowledge workers. Second, it points up the importance of interactions to engage the end-users in the process of the development of a visual analytics system.

The core of this recommendation is the characterization and generalization of a set of general domain task activities and workflow. This set of task activities are cross-domain, and are consolidated and characterized based on the Think Loop model [131]. These tasks represents the common task activities that are are applicable to a wider selection of organizational environments, and provides the design of visual analytics in a theory of information flows through the users' analysis processes.

Based on a careful analysis across all three organizations, it is clear that, while different organizations shared diverse tasks, each's analytical processes constituted a series of similar, loosely defined, and collaborative task activities. Knowledge workers accomplished analytical goals via subtasks, had focused targets, and accessed a range of services and resources [55].

As shown in Figure 7, this recommendation identified six task activities common to organizational analysis processes. On one hand, these common task activities serve as means to achieve user-friendly system designs, as shown in Watson et al.'s prior research. On the other hand, they are crucial components in representing the general domain analytical workflows, and further illustrate methods to engage the domain

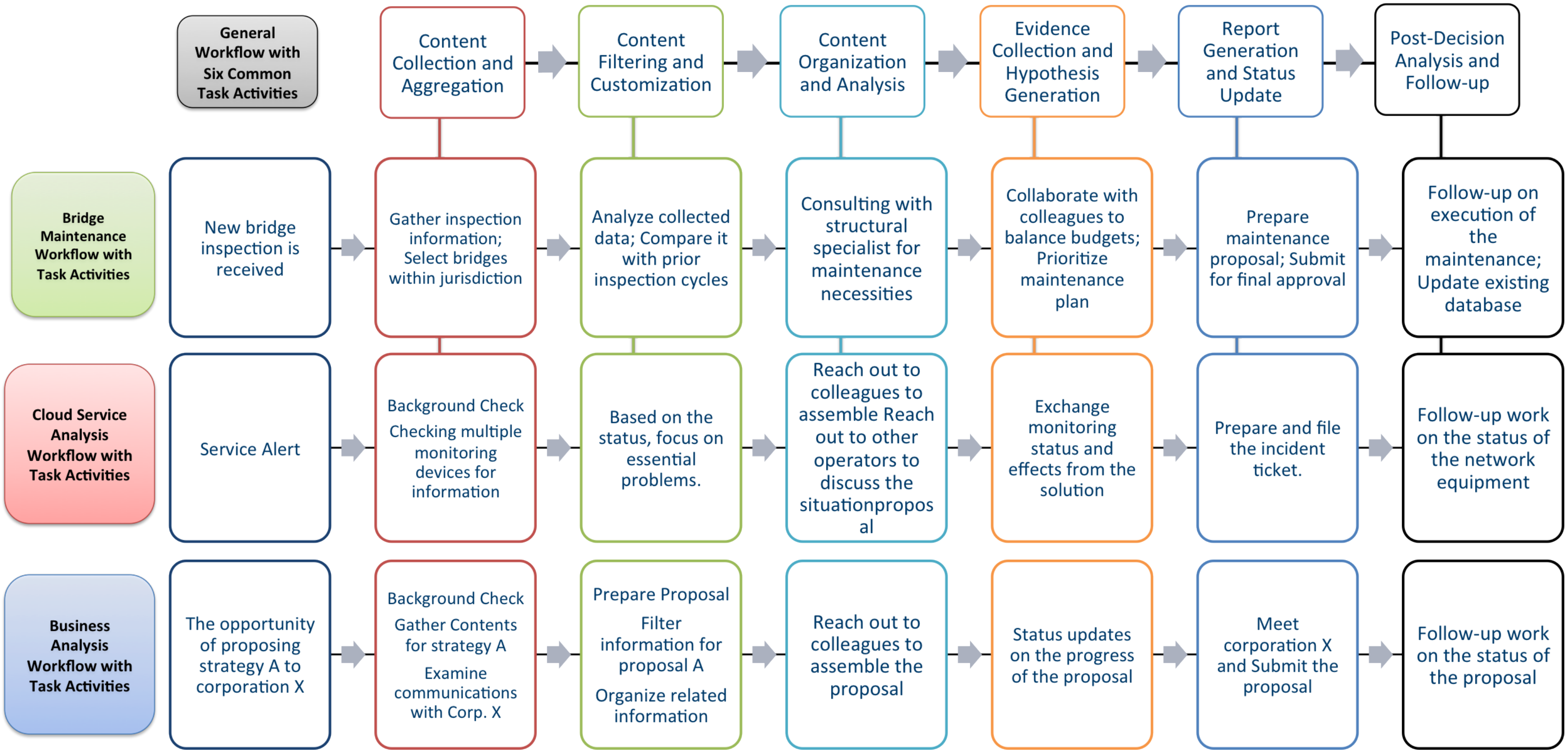


Figure 7: This chart describes the workflow in each organization. (Top) The six task activities common to both organizational analysis processes. (Middle) A typical analytical workflow for bridge maintenance planning. (Bottom) A typical analytical workflow for a business information analyst.

analysts in the center of the visual analytics design practice.

Utilizing these common task activities, this recommendation lays out the concrete items that future visual analytics designs could use to approach the end-users. It aims to address the end-users' lack of incentives to use a new system, and focuses on motivating them to adopt advanced analytical tools and practices.

- ***Content Gathering and Aggregation:*** Knowledge workers identify appropriately-scoped content to form basic analytical contributions. They seek and extract information from multiple channels relevant to the analytical tasks.

- ***Content Filtering and Customization:*** Knowledge workers use filtering to familiarize themselves with content they have collected. They also personalize the analysis environment in which this content is filtered.

- ***Content Organization and Information Analysis:*** Knowledge workers organize the collected content and examine it from multiple perspectives to look for data patterns and desired information.

- ***Evidence Collection and Hypodissertation Generation:*** Knowledge workers create hypotheses regarding their analyses, and collect related supporting evidence.

- ***Report Generation and Status Update:*** Knowledge workers increase visibility to others regarding analysis status, by providing notification and updates on the progress of their analyses,

- ***Post-Analysis and Summarization:*** Knowledge workers focus on validating project achievements and introspecting workflows, after accomplishing an analytics process.

3.3.3 Summary: Recommendation 1

This chapter describes the first recommendation in the design of a organizational visual analytics system, which is “Characterize Organizational Analytics Processes Through Interactions with Domain Users ”. This recommendation illustrates the importance of domain characterization in designing visual analytics, and demonstrated three successful research collaborations following this direction.

To support integrated analytical workflow and visual analytics system design, this recommendation extend the individual workflow diagram from above three organizations to reflect a general analytical workflow that is applicable to various domains. This extended workflow representation augments the reusability of each domain-specific analytical task to a broader, more general analytical scope. And it further concludes a set of six analytical task activities that commonly exist in various analytical domains.

This cross-domain perspective facilitates system design in terms of understanding the high-level sense-making process in general analytical workflows. It provides a consistent and efficient instantiations to start visual analytics system designs, and serves as a good initial approach to engage and communicate with domain users.

3.4 Recommendation 2: Disseminate Analytics Workflows into Key Actionable Knowledge

The six common task activities described in Recommendation 1 shows the general analytical relationship between the knowledge worker, the information source, and the workflows between the worker and the information source. This tasks activities outline the analytical processes that are used in a general analytical process, and illustrate the high-level utility of each process.

While these activities are useful in describing a general analytic process, they are often too general to provide any specific recommendations in actual system designs. In order to designing a visual analytics system requires support for the analytical workflows of the knowledge workers, it is therefore needed for the identification of tangible design artifacts that can connect the design of a system with the general domain analytical workflows.

The second recommendation, therefore, focuses on disseminating high-level analytical workflows to tangible design artifacts. This recommendation is established based on the determinants of the tangible artifacts in terms of characteristics of the knowledge worker's analysis tasks, information source, and the relationship between both the task and the needed information source. It is typically conceptualized into a two-step action to identify the nature and criteria for the tangible design artifacts, and to describe the general analytical workflow using these artifacts.

In the process of searching for the tangible design artifacts, this dissertation has performed an extensive studying of related research fields, such as knowledge management, business intelligence, intelligence analysis. The intent of the study is to relate to existing theories in determining the criteria of the artifacts; and it presents insights of the associations of analytical needs and task-relevant expertise. This investigation would further be used in the second step to illustrates the task relationship within the general analytical workflow, and further disseminate the common task activities into concrete visual analytics design elements. In the following sections, both investigation steps are described in details.

3.4.1 Terminology: Design Artifacts

Following Simon [147] statement on "solving a problem simply means representing it so as to make the solution transparent.", this dissertation considers the fine-grain "representation", which solves an organizations analytical problems, an "*design artifact*". The design artifacts by nature hold sufficient amount of information to represent the domain problem-solving process.

The design artifacts can lead to the depiction of the rich phenomena that emerge from the interaction of people, organizations, and technology [96]. More importantly, they can be qualitatively assessed to yield an understanding of the phenomena adequate for theory development or problem solving for an organization [96]. As field studies enable behavioral-science researchers to understand organizational phenomena in context, Nunamaker et al. [121] pointed out that such artifact can be represented in a structured form and are useful for design-related researchers to understand the atomic-level task activities to organizations' analytical solutions

As suggested by Hevener et al. [72], the proper way to construct and evaluate the design artifacts is through empirical observations and studies of an organization. Particularly, constructs, models, methods, and instantiations needs to be exercised within appropriate environments to obtain the desired design artifacts.

Therefore, given the imperative value of such artifacts, this dissertation emphasizes the use of them in informing the appropriate specification for both visualizations and interactions that are used in a visual analytic system. Specific to the dissertation, it focuses on capturing the design artifacts that are concrete enough for practical visual analytics system designs; and more importantly, such artifacts must be consumable for the knowledge workers, who need to decide how to make use of them, without introducing a considerable cognitive overhead.

Several empirical observations with multiple organizations have been conducted in this dissertation to search for the tangible design artifacts. Details about these observations are described in the following sections:

3.4.2 Action: In Search of Tangible Design Artifacts

The search for tangible knowledge and design artifacts has been conducted in many research fields, for instance, knowledge management field, organizational learning, information retrieval, and intelligence community. For identifying the proper tangible design artifacts for visual analytics systems, an extensive literature study has been conducted to establish basis of this design research.

On one hand, many approaches have been used to denote such artifacts in the field of knowledge management (KM). Prior research in knowledge management focuses on the capture and sharing of codified experiences and re-apply them to products [42, 41]. This research has focused on the key role of knowledge and its management in the analytical processes. A significant amount of the research efforts have emphasized on building a tangible knowledge-flow, which is then used to associate the knowledge process (e.g. knowledge creation, consumption, and transfer) with the analytical process. Among this research, Nonaka et al. [119] has proposed an influential organizational knowledge management theory. This theory treats the artifacts in an organizational workflow as tacit (internal, in users' minds) and explicit (written down, and transmittable) knowledge. As shown in Figure 8, a group of four general knowledge conversion processes is designed to describe to support the transmission and communication of these artifacts. The KM researchers have further attempted to deal with the management control regarding leveraging the knowledge structure at the intersection of human and computer [139]. For instance, Nissen [118] has extended Nonaka's theory and developed a model of the knowledge-flow phenomenon. They advanced this theory by incorporating the consideration of time and life cycles of each conversion process. As show in Figure 9, Nissen et al. described in depth about the development of knowledge management in seeking of the organizational knowledge artifacts designs.

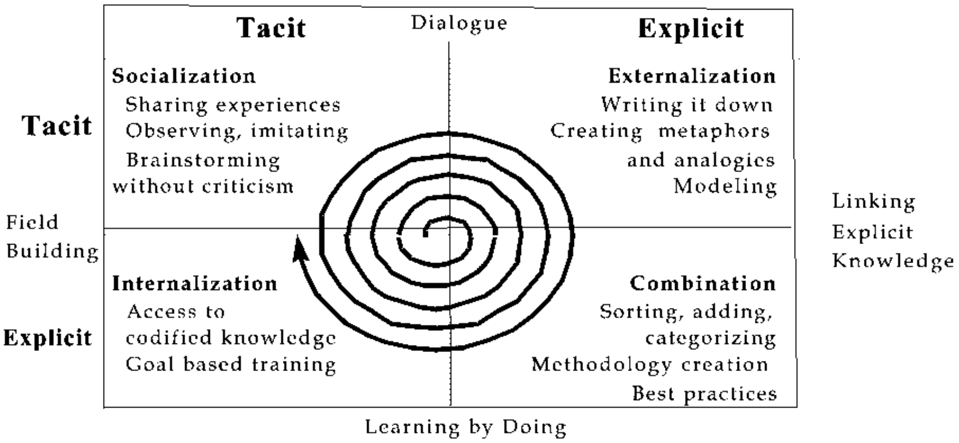


Figure 8: The four knowledge conversion processes illustrated by Nonaka et al. [119]

Model	Phasel 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6
Nissen	Capture	Organize	Formalize	Distribute	Apply	
Despres and Chauvel	Create	Map/bundle	Store	Share/transfer	Reuse	Evolve
Gartner Group	Create	Organize	Capture	Access	Use	
Davenport and Prusak	Generate		Codify	Transfer		
Amalgamated	Create	Organize	Formalize	Distribute	Apply	Evolve

Figure 9: The Knowledge management life cycle models (adapted from Nissen et al. [118])

On the other hand, the intelligence community has also focused on identifying the suitable structures that can be used in representing the design artifacts. Towards computational approaches, the postulated central role of acquired knowledge has encouraged efforts to computationally externalize experts' knowledge into structures that can be utilized by computers and users [74]. Much of the research has turned into expert systems, which matches the structure of a knowledge-base to the knowledge representation of domain experts. Much organization design knowledge is represented as "if-else" heuristics for suggesting the organizational decisions. Several techniques have been developed to help systematically construct such knowledge structures. Specifically, Koppen [97] developed an interactive questioning routine to extract the personal knowledge of domain experts. In addition, Langley et al. [100] demonstrated the possibilities of using computers to algorithmically compute such knowledge structures and facilitate users to search for patterns in data repositories.

Although knowledge management theory and expert systems may cover a wide range of organizational knowledge basis, their view of knowledge process are not suitable to represent the design artifacts for a visual analytics system. First, the conceptualizations of knowledge being static and hence can be modeled into certain fixed structures limits the design utility of the designed systems. Many of current analytical practices in organizations are dynamic, requiring the fast adaptation of emerging knowledge. The knowledge-base abstracted from each organization may be lagging behind the actual development of actual analytical process. For instance, in an analytical domains that shares a fast changing pace (i.e. financial domain and cyber security), the abstracted domain practice, for example cyber attack patterns or wire fraudulence patterns, could have already changed by the time the end-users utilizes the information system. The less flexible structure of knowledge-base would restrict the expansion and utilization of new domain analytical practices.

Second, many of current knowledge management systems lack consensus and knowledge integrity due to the diversity of knowledge worker's practices. Domain users do not follow a finite set of rules to achieve analysis, but rather balance many tasks simultaneously and search for an optimal solution [126]. As Bucher et al. [20] pointed out that, information analysis is generally isolated from the knowledge workers analytical workflow, leaving a significant amount of data and information detached from an interpretation context. Since information and domain-knowledge used in an analytical workflow is not evenly distributed through the organization, individuals knowledge workers may have individual task routines when solving analytical problem and their analysis may share differences. Therefore, while the rule-based presentation may work for experts, observations show that it doesn't match non-expert's expectations of an analytical system [109].

Finally, yet importantly, it is traditionally a challenge to assess and maintaining the validity and integrity of a knowledge base. In general, there are four potential issues that can introduce informal or even ill-structured domain knowledge into an existing knowledge base: duplicated, partial-overlapped, imprecise and conflicting knowledge [167]. Without careful validations of the association between the system and actual domain analytical practice, these issues may potentially degrade the value of the incorporated knowledge and may lead to inaccurate analytical processes. Nonetheless, it is still quite cumbersome in applying precise metrics to validate the integrity of a knowledge base, due to the quantify of knowledge elements and complexity of the internal relationships.

3.4.3 Action: Identify Design Requirements through Implicit Dialectical Process

Even though the knowledge management modeling approaches are limited, they do point out the importance of the existence of domain expertise and ways to extract them from the domain users. It is recognized that the autonomy of knowledge workers makes explicit knowledge-modeling process guidance risky and failure-prone. Based on the examination of previous literature [109, 74, 10, 178], there hasn't been a effective way to ensure that the knowledge workers would conduct complete analyses or engage their co-workers in deliberations about the meanings of terms, interpretations of findings, and evaluations of alternative actions. Therefore, instead of asking them information explicitly, this research guided the domain users (including expert representatives and general end-users) implicitly through the task analysis, and emphasized on revealing the importance of knowledge actions to support the transitions between different analytical task stages,

The implicit task analysis were done through dialectical process [109]. Through communicating with these professionals, this dialectical process placed the domain users in their most familiar environment, and encouraging them to perform as much analysis as they normally would do. In this process, both consensus and contradictions were collected to form the mechanism by which effective actionable knowledge is identified and captured. Both processes were further integrated within the analytical process as a whole.

These task analysis further led to the determinants of the target design artifacts for two basic requirements: (1) they need to be concrete enough for practical visual analytics system designs, and more importantly (2) they must be consumable for the knowledge workers, who need to decide how to make use of them, without introducing a considerable cognitive overhead.

3.4.4 Represente Organizational Analytics Processes using Actionable Knowledge

3.4.4.1 The Utilization of Theory of Action

Similar to Heuer's [71] perspective, this research considers knowledge as a dynamic expectation of information. It regards the requirements for the tangible artifacts as the product resulted from characterizing of the relationship between the analysis tasks and information source.

Enlightened by the Theory of Action [10], this recommendation followed Anrigyri et al.'s definition, and described the target artifacts as a series of ***Actionable Knowledge***. Actionable knowledge is explicit symbolic knowledge, typically presented in the form of *tradeoffs for action* or *action rules* [109], which allows the decision maker to recognize some important relations and perform an action. Information search and problem solving in these organizational analytical processes are directed toward utilizing actionable knowledge, which in turn leads to immediate progress on a current assignment or project.

The concept, actionable knowledge, represents a pragmatic view of knowledge utilization and application toward specific analytical ends [25]. Such examples can be seen as targeting a direct marketing campaign in a Bank's operations, or planning infrastructure maintenance aimed at repairing those assets with lowest health in asset managements. The initial knowledge exchanges between the researchers and the organizational members is through the shared trust of using interactive dialogues. It emphasizes on working with groups of organizational employees as co-researchers, and developing the self-reflexive critical awareness that triggers action based on the knowledge that is created [38].

Compare to the traditional knowledge management process, the creation of actionable knowledge is typically coupled with situational characterization in an organizational environment [10, 170]. It is distinguishable by the following five key processes: the emergent task process, the inquiry process, the integration process, the experimental process and the diffusion process [142]. These processes focuses on utilizing an emergent collaborative inquiry process in which behavioral and social science knowledge is integrated with existing organizational knowledge for the purpose of generating simultaneously scientific and actionable knowledge [38, 170]. During these processes, the organizational members are fully involved in the inquiry process and share the responsibility for the effort [125].

All these characterization processes guaranteed the nature of actionable knowledge would fit well with the above two requirements in that: (1) it represents the fine-grained elements of each analytical task, and thus is quite instructive for the design of a visual analytics system; (2) it is extracted from domain users' knowledge actions, and therefore can be consumed without additional cognitive overhead.

As illustrated in [10] [20] [40], there are many approaches to acquire and model actionable knowledge. Given the advantage of the existing close working relationships with actual domain users, this research adopted the domain-driven modeling process, and grounded the search for actionable knowledge on the interviews and surveys with the above three interviewee groups.

During the interviews, all the participants were asked to envision the hypothetical process of carrying out their usual tasks with their regular tools and working environments.

They were encouraged to also think about additional functions that might be useful but not yet available in any of the tools they typically used.

Specifically, these participants were asked about the fine-grained knowledge actions they used in their daily practices, the essential tools they have, and how they utilized these tools to execute each action. The interviews were semi-structured with the ambition to encourage the respondents to give a narration more than just answer questions. In doing so, this research was able to identify key actionable knowledge that a tool should support to improve productivity and reduce workload.

In their responses, all the interviewees expressed the importance of actionable knowledge to the organizational decision making process. In their analytical process, actionable knowledge is followed to respond to different situations, and illuminates potential action paths for overcoming obstacles. The use of actionable knowledge further directs these professionals to discover certain information or data patterns, and helps them to react to the advantages of a specific task.

For example, for the content aggregation task, a bridge manager often needs to check multiple sources of information (e.g. structural, financial, and historical) prior to their response for a new bridge maintenance request. During this process, actionable knowledge regarding where to look for information, and how to examine the information, plays a significant role in addressing this task.

Tools, in this context, are considered as means to transform the knowledge into desired task actions. Knowledge workers primarily use tools such as email/documents/local folders, to produce and communicate task related contents and information. In the process, their domain knowledge (i.e. the expertise) is employed, and further results in context-dependent actions that are used in their analytical process. These professionals currently posses and use a number of different tools; however, we found that both groups were severely lacking tools that were actually designed to support to their analysis workflows. This finding pointed to the need for a tool that encapsulates the users' actionable knowledge and helps them effectively perform necessary actions.

Based on the feedback from the interviews, a set of selected actionable knowledge was summarized to describes the six common organizational task activities. As shown in Figure 10, every task in an analytical process is decomposed into a set of fine-grained actionable knowledge. Note that, this list contains only a subset of all the collected actionable knowledge; some of the stated actionable knowledge is unclear, ambiguous, or contradictory, and is therefore excluded from this list. Also as seen in Figure 10, a clear mapping have been constructed between high-level tasks and their fine-grained tangible artifacts. This mapping provides clear insights into the organizational workflow. More importantly, it is further transformed into a range of important design requirements for creating an effective visual analytics system.

3.4.5 Summary: Recommendation 2

This section introduces the second recommendation in the design of an organizational visual analytics system. This recommendation emphasizes the importance of disseminating the common domain task activities into the tangible design artifacts for visual analytics systems. Specifically, this research focuses on deducting essential actionable knowledge that are used in typical analytical workflows, and proposes the use of this knowledge as key design artifacts for a visual analytics system.

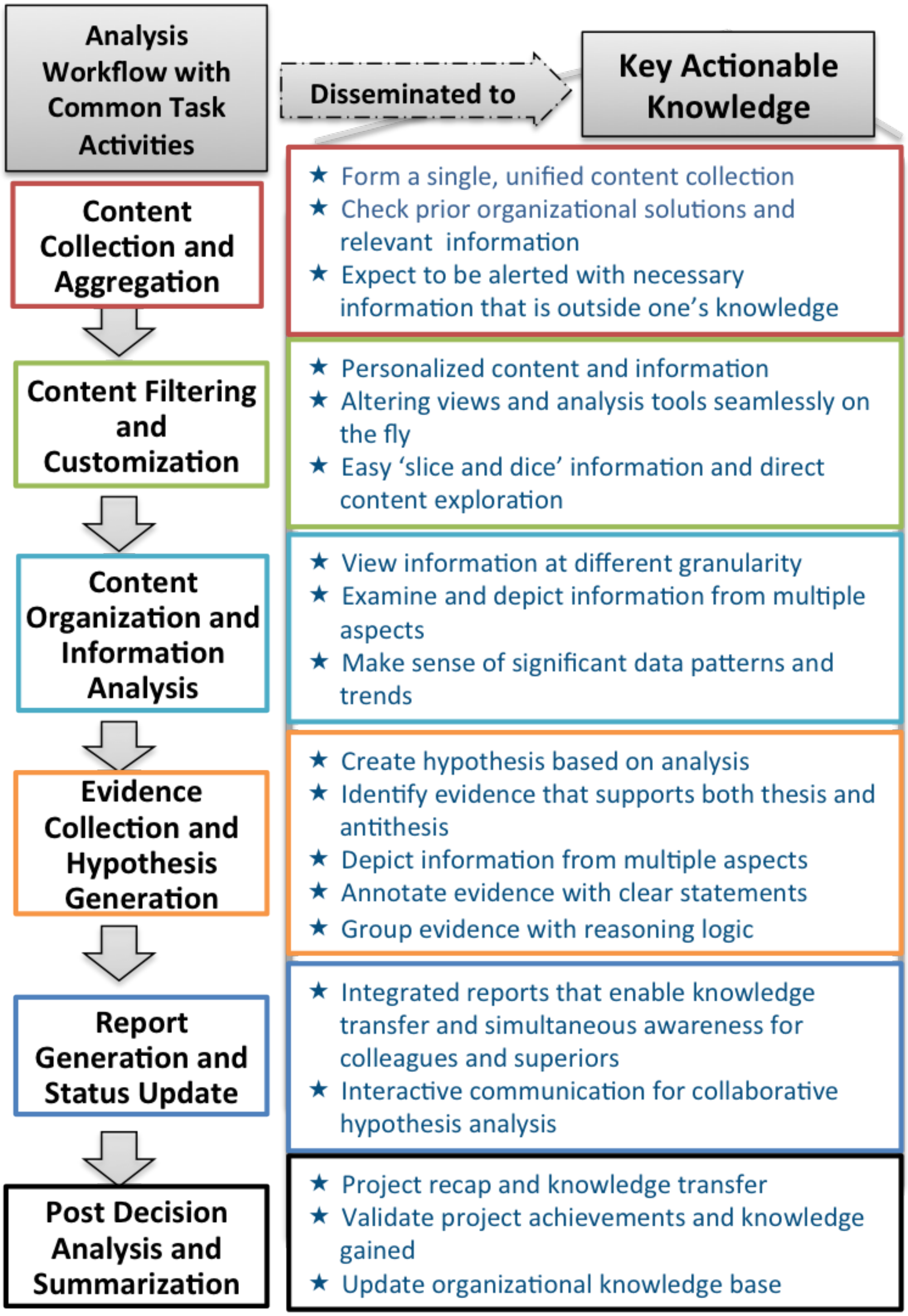


Figure 10: The identified actionable knowledge (design artifacts) in checklist form.

3.5 Recommendation 3: Design for Actionable Knowledge Transformation Through Software Prototyping

3.5.1 Design Considerations to Transform Actionable Knowledge into Visual Analytics Systems

To determine the likelihood of domain users' accepting a visual analytics system's functionalities, the initial design specifications for visual interfaces and interactions need to be concluded right after the identification of the above actionable knowledge and design requirements. Based on the two longitudinal field experiments on the affect of user acceptance testing to the development of a user-centered system, Davis and Venkatesh [44] pointed out that errors in requirements specifications have been identified as a major contributor to costly software project failures. It would be highly beneficial if information systems developers could verify requirements by predicting workplace acceptance of a new system based on user evaluations of its specifications measured during the earliest stages of the development project, ideally before building a working prototype [44].

Therefore, this recommendation provides a list of possible design considerations (see Figure 11) to encapsulate the general domain task activities into the design of a visual analytics system. It emphasizes on deducting proper specifications to instantiate the design of a visual analytics system, and encapsulates the general analytical workflow into users' accepting visual analytics functions.

These considerations are generalized based on the transformation of the above-specified actionable knowledge into actual visual analytic systems, in all of the aforementioned organizational environments. Following commonly established design theories [143, 109, 178], these considerations are identified through several iterations of prototyping with the targeted domain end-users.

As shown in Figure 11, they describe in details about the needs for transforming each general analytical tasks into system's functionalities. As exemplified in three successful visual analytics systems, these design considerations are demonstrated to have sufficient information to inform successfully implementations for visual analytics systems. They are further evaluated by domain users through extensive empirical evaluations, as reported in chapter 5

Iterative prototyping and formative evaluations presents an invaluable role in identifying these design considerations. They essentially helped to encapsulate the domain users' actionable knowledge into functions, and led to the acquisition of critical functionalities required to build a visual analytics system. Although all three organizations shared similar common analytical tasks, the implementations of the prototyping methods for them were quite different (considering their diverse workspaces and time constrains). Specifically, the evolutionary prototyping [122] method was adopted for the collaboration with Xerox. Given the requirement of a deployable product to the enterprise, the evolutionary prototyping guaranteed more design iterations and, more importantly, allowed the build of a robust system in a structured manner. It also enabled the designers to constantly refine of the system and to explore broader options for transforming the actionable knowledge into visual analytics designs.

As indicated based on the iterative prototyping, the business analysts had a clear preference for a unified, intuitive, and less intrusive system that can help effectively retrieve and manage desired information. Therefore, Taste was finalized and implemented to support such preference [166]. Taste is an interactive visual analytics system that enhances employees' capabilities to search and share business information. As shown in Figure 12, Taste is structured to embed information retrieval cues into a coordinated multi-level visualization system. At a high level, Taste encodes these cues with a set of three visualizations, a Facet view (A), a Temporal view(B), and a Entity Tag view (C). Each view presents a particular aspect of document activity information across entire collections. In lower-level views, Taste presents visualizations that integrate related activity information for single documents(D). Using this multi-level structure, Taste helps users to cohesively depict document activity from different

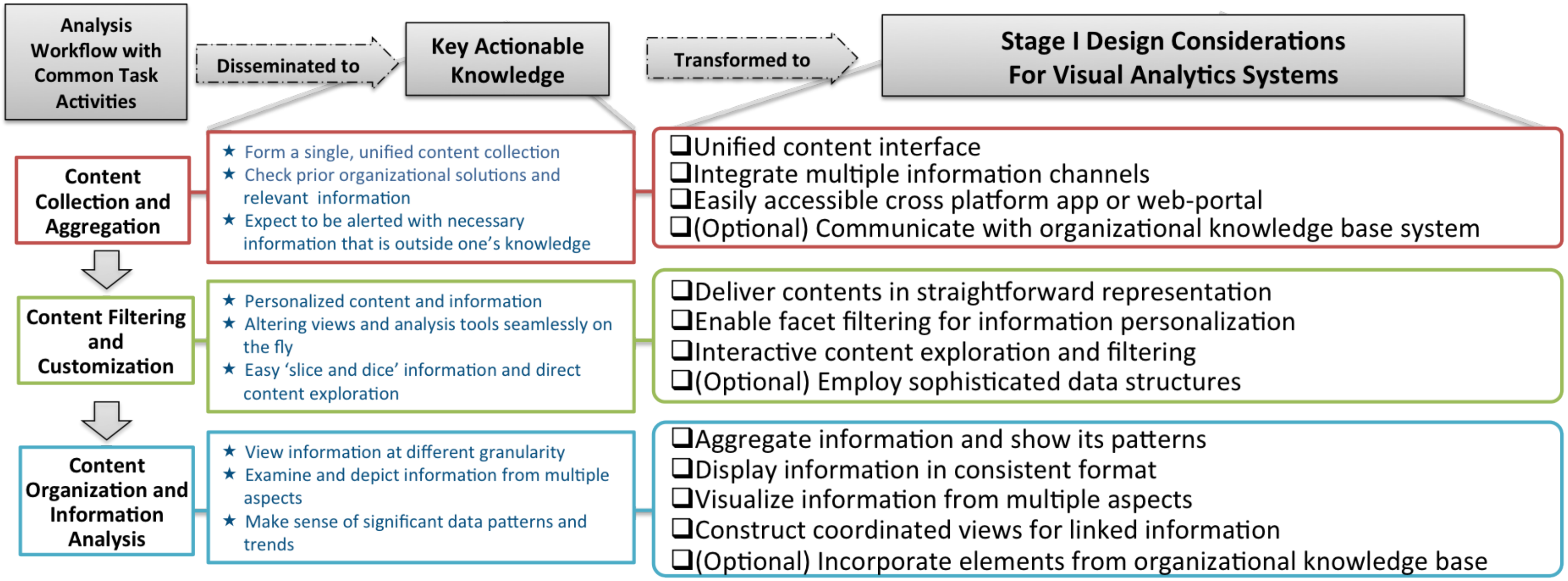


Figure 11: The design considerations for Stage-I, the observation and design stage.

points of view, and effectively find the desired information. Details of the implementation of Taste can be found in section 3.5.2.1;

Given the shorter design cycle (three months), a more frequent, rapid prototyping method [7] was used for the project with Microsoft. This rapid prototyping provided efficiency in modeling domain requirements, and helps to create a working model of various parts of the system at a very early stage, after a relatively short investigation. The model then becomes the starting point from which users can re-examine their expectations and clarify their requirements. When the basic model has been established, the system is formally then developed based on the identified requirements [7].

During the three month collaboration, over 7 prototypes were generated. Among all these prototypes, what the cloud service managers emphasize most is the ability to have a single tool to monitor the status of the entire cloud service. To accommodate their individual task requirements, these managers also require the ability to customized their analyses views. (e.g different combination of status, data sources and servers). As a collaborative results, an interactive visual analytics system, OpsVis, is therefore implemented to support all these requirements. As shown in Figure 19 OpsVis provides a single view of the entire cloud service, as a network. It shows high-level objects, such as clusters of servers, as a single unit. These units are interconnected with network edges, allowing the user to visualize the network configuration and dependencies. Details of the design of OpsVis is introduced in section 3.5.2.2;

Thanks to a long-term collaboration plan, a longer-cycle, more functionality-based prototyping process was carried out by this dissertation with the bridge managers at both USDOT and NCDOT. This iterative functional prototyping [60] simulates application behavior and helps to ensure that more of our design system is understood at each step of the collaboration. In each iteration, the bridge managers were invited to test and evaluate our prototypes by working with the system to perform actual bridge analysis. Based on their suggestions and requests, this research then refined, re-designed, and re-implemented the prototype system to increase its effectiveness to support the bridge analysis process.

During a nine-month period, over ten functional prototypes were generated, including various changes to the visualization and interface designs. Over the course of past two years, the prototyping has resulted in a final set of variations of the system. These all focus on providing support for bridge management using integrated remote sensing and visualization, so and they are generally referred as IRSV. While each of the systems is designed to accommodate requirements for different use cases, all follow a similar set of underlining actionable knowledge, and were designed to achieve the same goal: to provide examination of heterogeneous data sources and facilitate effective bridge maintenance planning.

At the heart of IRSV is a set of visualizations to help bridge managers organize and analyze their assets from the multiple perspectives essential to their decision-making process. As seen in Figure 21, these visualizations were designed to perform the three high-level analyses: structural analysis (G), temporal analysis (H), and geospatial analysis (I). For lower-level tasks, we designed a structural detail view (F) to automatically link information between each bridge component, and provided bridge managers with an intuitive visualization to interactively analyze specific corresponding information. All of these visualizations are tightly coordinated together in such a way that an action performed in one view affects all other views. Implementation details can be seen in section 3.5.2.3

3.5.2 Actions: Iterative Prototyping for Organizational Visual Analytics Systems

Systems like Taste, Opsvis and IRSV were designed following the above listed design considerations. These visual analytics systems are implemented to support the analytic processes encountered in organizational environments. Through iterative prototyping processes, each was tailored to the analytical workflow of its target domain. As shown in the following above examples (see section 3.5), the design considerations actually incorporated within each system are illustrated separately.

In the following section, this dissertation will introduce each design practices in details, including the utilized design considerations (including the total numbers used in designing each system), the prototyping process, and the finalized visual analytics systems.

3.5.2.1 Case: Taste for Xerox Corporation

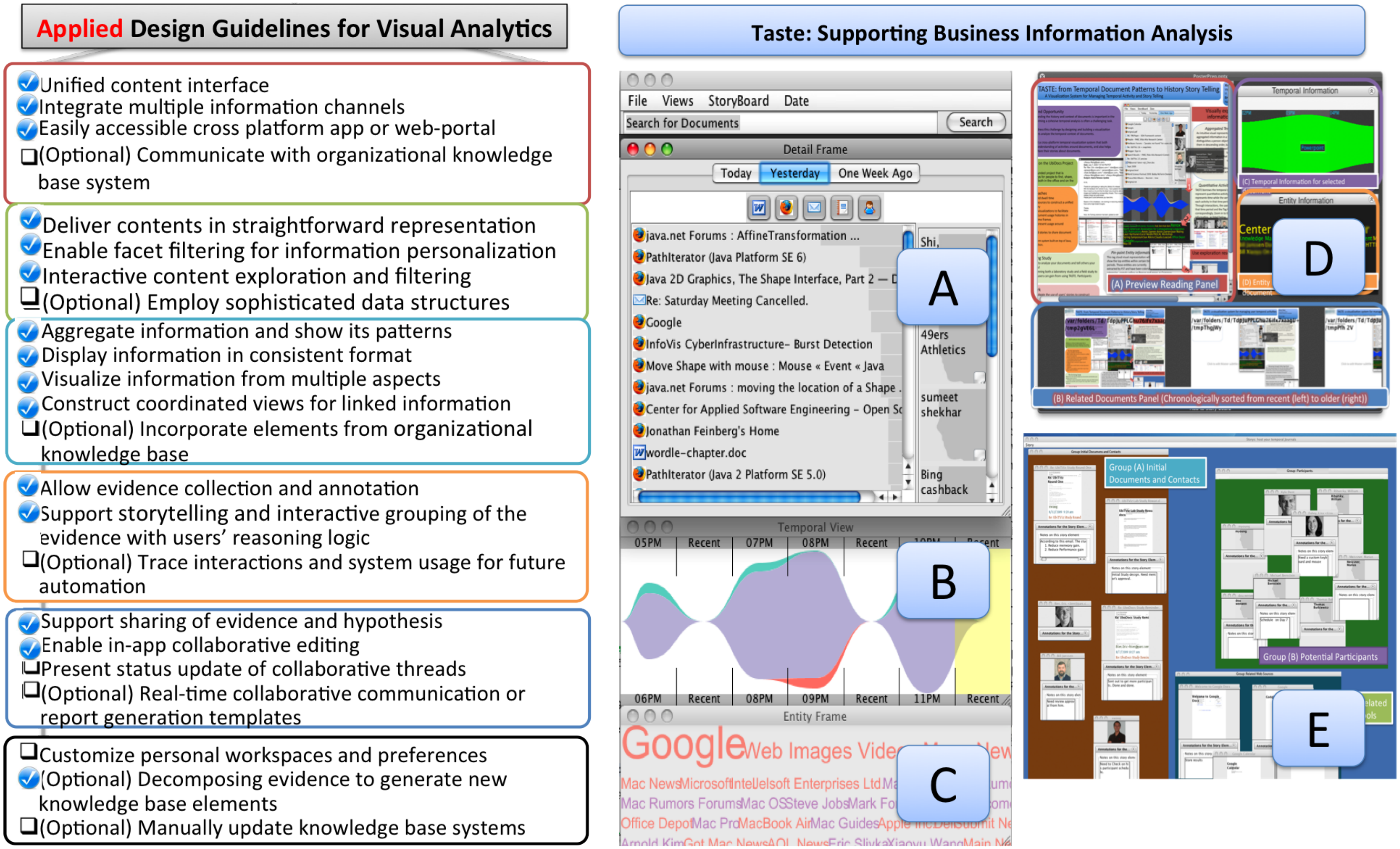
In response to the identified needs for an integrated, efficient, information retrieval tool tuned to the enterprise environment, Taste was designed and deployed to facilitate the analysis needs for employees in Xerox corporation. Taste is an interactive visual analytics system that aims to enhance corporate employee capabilities for finding and sharing the business information that is embedded in their daily document activities.

As shown in Figure 12, Taste is designed following the general design recommendations that are indicated by the marked checklist. Specifically, the design of Taste is essentially centered on incorporation of the aforementioned retrieval cues 3.3.1.3, namely temporal hints, content keywords, and document types or particular applications. It integrate all these retrieval cues with the general design considerations to support the users' search for the desired information. Figure 13 further illustrated the detailed design pipeline for Taste, including considerations from data and information processing to interactive visual interface.

In the following paragraphs, we explain the in detail on how each general design consideration and domain information retrieval cue is transformed into the design of Taste:

Data Capture and Storage

At the heart of Taste is an automatic and transparently real-time contextual data capturer. As shown in Table 5, this data capturer is implemented to provide a unified content interface as well as to integrate multiple information channels. As part of the UbiDocs project [81], Taste runs on a single user's computer, and captures user



Applied Design Guidelines for Visual Analytics

- ✓ Unified content interface
- ✓ Integrate multiple information channels
- ✓ Easily accessible cross platform app or web-portal
- ☐ (Optional) Communicate with organizational knowledge base system

- ✓ Deliver contents in straightforward representation
- ✓ Enable facet filtering for information personalization
- ✓ Interactive content exploration and filtering
- ☐ (Optional) Employ sophisticated data structures

- ✓ Aggregate information and show its patterns
- ✓ Display information in consistent format
- ✓ Visualize information from multiple aspects
- ✓ Construct coordinated views for linked information
- ☐ (Optional) Incorporate elements from organizational knowledge base

- ✓ Allow evidence collection and annotation
- ✓ Support storytelling and interactive grouping of the evidence with users' reasoning logic
- ☐ (Optional) Trace interactions and system usage for future automation

- ✓ Support sharing of evidence and hypothesis
- ✓ Enable in-app collaborative editing
- ☐ Present status update of collaborative threads
- ☐ (Optional) Real-time collaborative communication or report generation templates

- ☐ Customize personal workspaces and preferences
- ✓ (Optional) Decomposing evidence to generate new knowledge base elements
- ☐ (Optional) Manually update knowledge base systems

Taste: Supporting Business Information Analysis

Figure 12: The design considerations for Taste. Taste consists of (A) Facet view, (B) Temporal view, (C) Entity Tag view, (D) Detail view, and (E) Storytelling view.

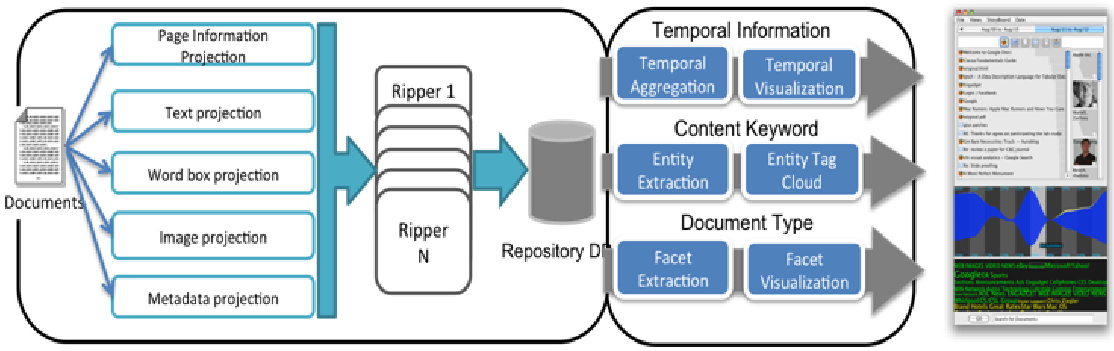


Figure 13: The Taste pipeline.

Table 5: Design considerations implemented in the data capture and storage component

1	Unified content interface
2	Integrated multiple information channels

activities around office documents, calendars, emails, Web pages, etc.

Like a desktop search engine [57], Taste creates an index of documents on a business analyst's machine. As shown in Figure 13, Taste utilizes UpLib [81] to extract information from and about each document, including its title and authors, its text, the people and other entities that it mentions, its paragraphs and its images. All of the captured information is indexed and grouped with its related documents. This provides Taste with a secure long-term storage for a wide variety of business related documents such as proposals, papers, web-knowledge bases, presentations, and email. Taste can support collections comprising tens of thousands of documents, and provided for ease of document entry and access as well as high levels of security and privacy. A cross-document similarity matrix is also created in real-time to enable search for related documents that are essential for business information.

However, more than a desktop search engine, Taste also logs information about a user's activities with documents. In particular, it collects information about which documents, emails, and web pages were read, and how long they were open. It also collects document metadata, such as information about the senders and recipients of email messages. Taste stores this activities information, along with copies of the documents, into the above unified document repository, UpLib.

Utilizing this rich data reposition, Taste is designed to integrate multiple channels of document data into a single content space. It further collects the document activities that contain rich information representing user's analysis behaviors. In doing so, Taste prepares a unified data repository that can help effectively manage relevant analytical resources; and based on this data repository, Taste further implement an interactive visual analytics systems that facilitate the users' to retrieve and analyze business information.

Visualization Interface

Table 6: Design considerations implemented in the visualization interface

3	Easily accessible across platform application or web-portal
4	Deliver contents in straightforward representation
5	Enable facet filtering for information personalization
6	Interactive content exploration and filtering
7	Aggregate information and show its patters
8	Display information in consistent format
9	Visualize information from multiple aspects
10	Construct coordinated views for linked information

Instead of presenting the diverse document activities through a keyword search interface, Taste embeds the retrieval cues into a coordinated visualization system, through utilizing the design considerations listed in Table 6.

At a high level, Taste encodes the three cues with a set of three visualizations, each of which presents a particular aspect of the document activity information. To provide a lower level detailed view, Taste also presents a visualization that integrates related activity information for a single document. Using this multi-level structure, Taste helps users to cohesively depict document activities from different points of view and to effectively find desired information.

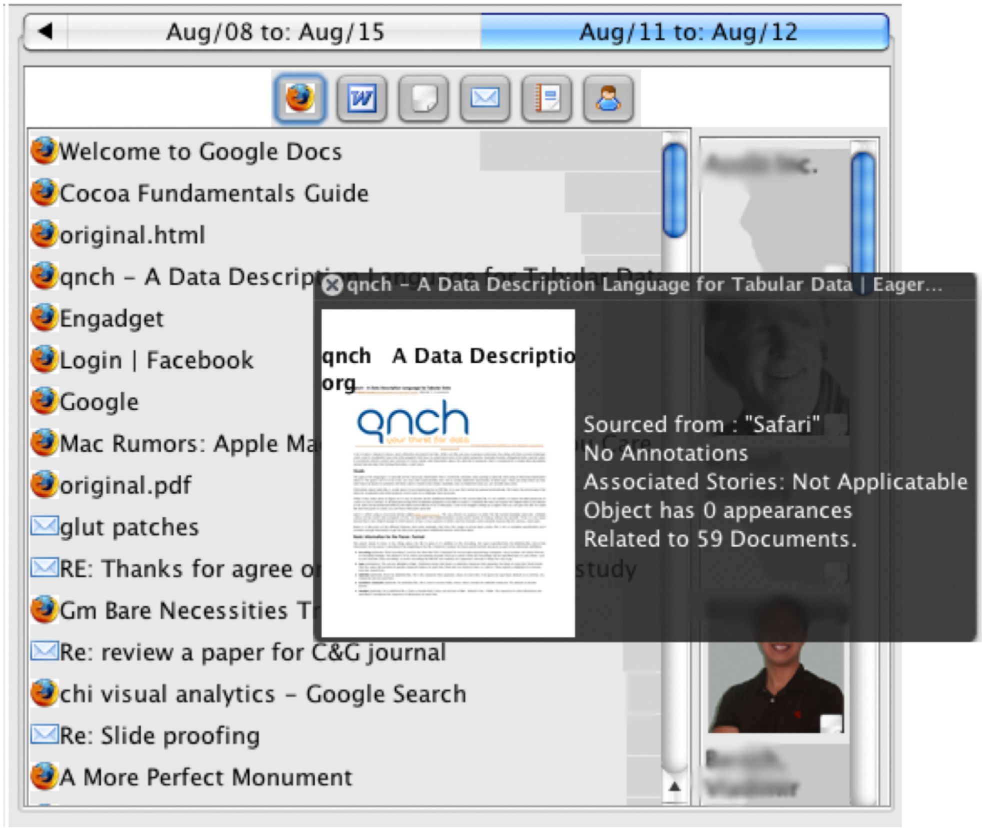
High-level Visualization Overviews:

Figure 14: The zoomed in temporal view in Taste.

The Temporal view presents temporal hints:

The Temporal View shows how a user's activities unfold over time. It presents both the number of documents a user interacted with in different time periods, and the types of those documents. This view is created as an interactive ThemeRiver [85], and it shows the temporal trends and patterns of a user's document activities. In this view, each vertical axis represents a period of time, while horizontal ribbons indicate both the format of documents (i.e., email, Microsoft Word, etc.), and the time spent on each. For example, Figure 12(B) depicts the history of a day in which a user spent a significant time browsing Web pages and took a quick break around 3:00 pm.

Besides showing general trends and patterns, the temporal view also allows the user to drill down into time periods. When the user selects a time period on the horizontal axis in the center of the view, Taste zooms into that period of time. At the same time, a time period summary window appears on the desktop (see Figure 14) that presents the highest ranked-N documents for that time period and provides the user with a quick way to return to the original time scale.



Welcome to Google Docs
Cocoa Fundamentals Guide
original.html
qnch – A Data Descrip
Engadget
Login | Facebook
Google
Mac Rumors: Apple Ma
original.pdf
glut patches
RE: Thanks for agree o
Gm Bare Necessities T
Re: review a paper for C&G journal
chi visual analytics – Google Search
Re: Slide proofing
A More Perfect Monument

qnch – A Data Description Language for Tabular Data | Eager...

Sourced from : "Safari"
No Annotations
Associated Stories: Not Applicatable
Object has 0 appearances
Related to 59 Documents.

Figure 15: The Taste facet view. The HUD display shows the information panel.

The Facet view shows document types:

To help corporate employees efficiently retrieve specific documents of interest, Tastes is designed with the Facet View to aggregate both the documents and the people that a user has interacted with during a particular period of time. Like Lee et al [102], this facet view allows the user to filter and sort information based on automatically-extracted data facets, including type (person or document) and format (email, slide presentation, text document, etc.) The user can query this view to filter its output based on a set of facets. For example, the user can choose to see or hide activities with email, with office documents, with Web pages, or with people. Each of these facets can be turned on or off by pressing an associated button.

In order to fit the activity information into a reasonable amount of screen space and in order to draw the user's attention to the most important activities in each time period, Taste sorts document activities by importance, and displays the most important documents at the top and with the most salient presentation by computing the importance value as:

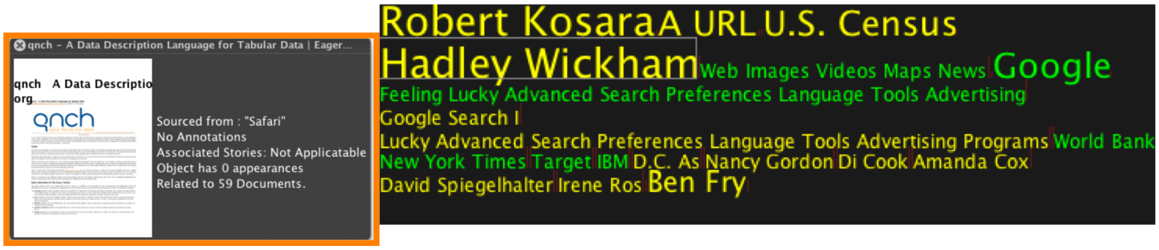
$$D_{importance} = F_{appearance} * \sum_{T_{cur\_start}}^{T_{cur\_end}} T_{dwell}$$

In this equation, the importance of each document ($\mathrm{D_{importance}}$) is set to the number of times each document appears in the repository ($\mathrm{F_{appearance}}$) multiplied by the sum of the amount of time the document was open on the display ($\mathrm{T_{dwell}}$, measured in milliseconds). Document importance is computed in a particular time frame, which is between $\mathrm{T_{cur\_start}}$ and $\mathrm{T_{cur\_end}}$.

Therefore, based on this equation, Taste considers the document that a user spent the most time on to represent the most important activity the user performed during that time period. To give users a sense of the importance of each document, Taste displays scale bars next to each document, where the length of the bar denotes its importance in that time period (shown in light grey as the background of each document in Figure 15).

To enable fast exploration, a summarized information panel (see Figure 15 is shown when the mouse hovers over a visual element representing an activity. Like the Document Card [149], this panel includes a readable thumbnail and aggregated information about that visual element. If the user needs more details, the user can double click on the visual element to bring up a Detail view window (See section

4.2.1.5).



Sourced from : "Safari"
No Annotations
Associated Stories: Not Applicable
Object has 0 appearances
Related to 59 Documents.

Figure 16: The Taste entity tag view.

The Entity Tag view for content keywords:

Since it is observed that content keywords from documents are helpful for information seeking, Taste extracts entities [16], such as company name, contacts, date/time, etc., from all of the documents the user has interacted with and displays them in the Entity Tag view.

To enable fast entity browsing and to emphasize the most frequently encountered entities from a selected time period, this view uses a TagCloud visualization of the entities. The size of each entity in the TagCloud is calculated based on the equation:

$$TagSize = F_{entity\_appearance} * F_{doc\_appearance} * \sum_{T_{cur\_start}}^{T_{cur\_end}} T_{dwell}$$

In this equation, the entity appearance frequency ($\mathrm{F}_{\mathrm{entity\_appearance}}$) is the number of times that the entity is mentioned in a particular document and the document appearance frequency ($\mathrm{F}_{\mathrm{doc\_appearance}}$) is the number of times a document was used in a given time period.

As shown in Figure 16, the Entity Tag view colors the extracted entities based on their categories, i.e. company or person. When the mouse hovers over an entity tag, a summarized information panel appears that shows the top ranked six documents that mention that entity. The user can double-click on an entity name to open a persistent window with information related to that entity, including a larger selection of documents that mention it.

View Coordination:

Since information retrieval can involve the utilization of all three retrieval cues, Taste is structured to encapsulate these visualizations within a coordinated system. All the visualizations in Taste are coordinated so that updates in one view are immediately reflected in the others. For example, if the user zooms in on a particular time period from the temporal view, the facet view responds by creating a new aggregated panel, and the entity tag view updates its displayed entities.

Therefore, a user can start the process of recalling document activities beginning from any retrieval cue that they remember. Updates in a coordinated view will often display information that is more similar to the desired information, allowing the user to follow a path through the visualizations and converge on the desired information quickly.

Low-level Detail View

While the above visualizations focus on presenting overviews of the entire collection, the Detail view depicts a single document from multiple perspectives, showing temporal information, related document information (i.e. how many versions of this document are available), and other information. The Detail view can be invoked from any view in Taste to learn more about a document appearing in that view.

As shown in Figure 17, the detail view contains four panels. Figure 17 A shows the preview panel that allows the user to view the document without having to reopen the corresponding application. By comparing paragraph, image, and page layout similarities [81], the related documents panel (Figure 17 (B)) recommends documents that are similar in content to the selected one. The temporal information panel (Figure 17 (C)) shows how much time the user has spent on that document and when activity with it occurred. Finally, Figure 17 (D) presents the entity information panel in which the user can browse all of the entities that were extracted from the document that is presented in the preview panel.

Figure 17: The Detail view for a selected document.

Summary Taste

In an enterprise environment, documents are of great importance to business operations and information flows. It is therefore essential for corporate employees to have an effective means to retrieve this information and use it in their work. Although current commercial products present efficient methods for keyword-based searches, they are not as effective in an enterprise environment, where information is hard to find by keywords alone.

Taste is therefore designed and implemented to alleviate such challenges. Taste is an interactive visual analytics system that presents an integrated document retrieval interface. As shown in Figure 12, Taste is designed following the general design recommendations that are indicated by the marked checklist. Specifically, the design of Taste is essentially centered on incorporation of the aforementioned retrieval cues 3.3.1.3, namely temporal hints, content keywords, and document types or particular applications. It integrate all these retrieval cues with the general design considerations to support the users' search for the desired information.

3.5.2.2 Case: OpsVis for Microsoft Corporation

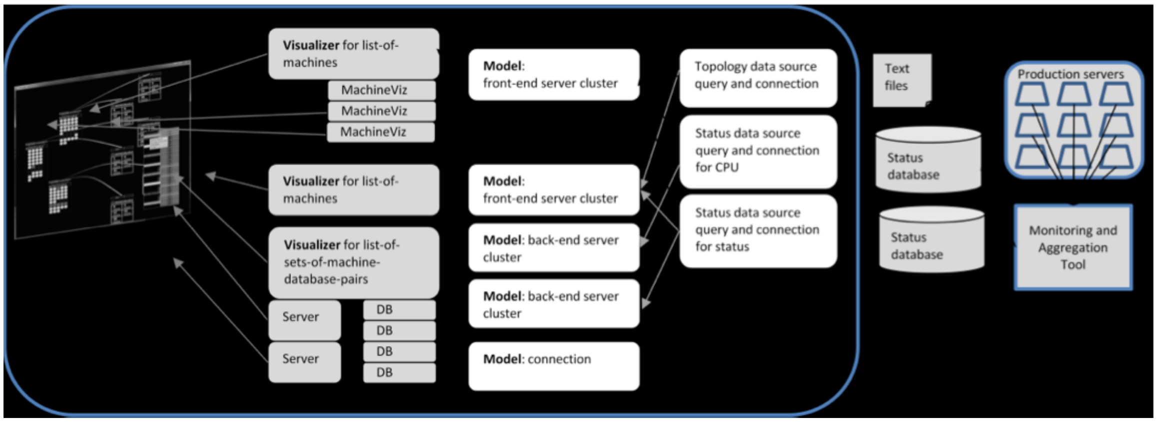


Figure 18: A representation of OpsVis, and the data structures. Areas in light gray are not directly represented in the configuration file, while areas in white are. The configuration file specifies both topology and status information. Note that OpsVis counts on external tools for monitoring and topology information.

In talking to service developers and operators, this research found that data fusion across numerous and diverse sources is a large part of the problem. The goal was not to build a tool customized for one particular service; rather, it was to build a general tool for data center operations applicable to any cloud service (i.e., a data center application that provides services to users across the Internet)

Working with a specific team allowed us to understand how that team creates, maintains, and uses a mental model of their server and network topology. Cloud services are typically constructed from a small number of different types of atomic objects, running service-specific code or configured in a service-specific manner: servers, databases, switches, load balancers. Second, these atomic objects are organized into clusters, with all objects within a cluster performing an identical role (although with different data). Third, within and across clusters, there is a high degree of redundancy in order to be resilient to failure.

Accordingly, a key component of the needed visual analytics system is a flexible back-end to accommodate these many forms of data, in a unified interface. In addition, another key requirement is the ability to customized analyses views that enables individual cloud service managers to perform their individual task requirements (e.g different combination of status, data sources and servers).

To support these requirements, this dissertation designed an interactive visual analytics system, OpsVis (See Figure 20). As shown in its pipeline (See Figure 18), OpsVis facilitate the cloud-service management from data back-end (e.g. collecting and aggregating topology and status data) to visualization front-end (e.g. interactive visualizations, real-time information update). The implementation of OpsVis follows the design considerations listed in the previous section 3.4, and the details are described in the following sections:

The Data Models

Table 7: Design considerations for OpsVis Interface

1	Unified content interface
2	Integrated multiple information channels
3	Aggregate information and show its pattern

Designed following consideration in Table 7, the OpsVis provides cloud-service manager with the flexibility to customize their workflows by editing configuration file. These files are explicitly specifies a set of Data Models. Each Model specifies the information to represent a single top-level object in the visualization. For instance, one Model can represent each front-end cluster, or can represent a single cluster of back-end servers. Figure 20 is represented by 16 Models: 4 front-end clusters, 6 back-end clusters, and 6 connections.

Models have two jobs: they contact data sources to collect topology and status information; and they create Visualizers to reflect this topology and status information. Models act as indices of the Visualizers that make them up; each of them has the ability to assign status information to any named entity. For example, if a status data source indicates that database ABCHA526 has gone down, then the Model can look up that name in its index, and relay that information to the corresponding visualizer. In Figure 19 (left), note that the two Models each specify a topological data source and a parameter.

Models must balance a tension between generality and specificity: their design must be general enough to be re-usable, but customized for the sort of data that they will represent. In our current implementation, we have several different Models. One represents a set of uniformly-configured, interchangeable entities; these are used to represent front-end clusters. A second represents one or more sets of back-end servers, which are each coupled with one or more databases. We are continuing to work on generalizing these components. Models populate themselves by querying a topological data source.

Topological Data Sources: Topological Data Sources specify the sources of the information that our tool will use. A topological data source returns a table of data that describes the underlying structure of the data. There is a tight binding between Models and the topological data sources: Models expect their data sources to provide an appropriate table. The goal of the topological data source is to populate the visualization with specific machines, databases, and other bottom- level items; and to cluster them into higher-level clusters.

A topology data source, in the current implementation, refers either to a text file or to an SQL query. In the case of a query, the topology data source specifies both the connection string to the database and the query to be invoked. One such SQL query is described in Figure 18. More complex Models can be populated simply by listing out the full combination of all items. For example, to populate the back ends (on the right side of Figure 19 (left)), the returned table contains rows with a set (e.g., 18), a server name (e.g., SQL04), a database number (e.g., 581), and a partition name (DB14708). The visual connection then assembles the multi-tiered structure on the right side.

Status Data Sources: Like topology data sources, Status Data Sources are based on queries that return tables of data. Models expect status data sources to return tables with columns representing the machine name and its value.

Because many status data sources may be relevant to a given visual component, status data sources contain the list of Models to which they apply. A status data source can be applied to any Model; the data they return can be applied to any named element. Thus, the status data source does not need to specify whether it is returning data that applies to a specific database, to a server, or to a connection; the key that the query returns disambiguates it. For example, receiving a key/value pair that CPU time on SQL04 has increased is enough: each Model then checks its name table to see if it knows about this machine.

The configuration file descriptor for a status data sources also contains information on how it should be rendered. This metadata consists of several cues to the system. It specifies whether the data should be interpreted as categorical or value, and what do to with the data: whether it should be interpreted as a color, as a text string, or in some other way. It also specifies to what part of the Visualizer the data should be applied. In the current implementation, the background color of the shape and the color of the "'glow" around the shapes can both be mapped to colors. While these might someday be automatically detected as in systems like Tableau [1], this is not currently implemented. The data source optionally specifies the minimum and maximum expected values: values outside this range are drawn as black or white. Last, status data sources specify a refresh rate: the frequency with which the database should be checked for new information.

A sample SQL status data source in Figure 19 (left) specifies that the column "CPUTime" should be interpreted as a value, rendered as color brightness based on data values from 25 to 35, applied to the background, and refreshed every ten seconds.

While the status data sources currently all use a "pull" model to retrieve periodic data from SQL servers or text files, the extensibility model designed would allow a developer to add an event-based "push" model, where only specific changes would be propagated through.

Aggregating Status Information: The cloud-service managers are not expected to examine all servers, all the time. Rather, the design of OpsVis aims to provide a set of aggregations that allow a these domain analysts to review a rolled-up set of servers without concern. Status data sources specify whether their value should be rolled up to the next level. Status data sources that expect to be aggregated can also provide instructions for what functions should be used to aggregate them: for value types, the user can specify minimum, maximum, and average; for categorical types, a distribution is drawn. The status data source in Figure 19 (left) expects to be aggregated with the average function.

The OpsVis Configuration Files

Table 8: Design considerations for the OpsVis configuration files

4	Easily accessible cross platform application or web-protal
5	Employ sophisticated data structures
6	Construct coordinated views for linked information
7	Enable in-app collaborative editing
8	Present status update for collaborative threads

One goal of OpsVis is to provide a declarative way of specifying the visualization techniques and data sources that are used in creating a visualization of a cloud service.This goal is achieved using the implementation of the OpsVis configuration file, which follows the considerations listed in Table 8. This declarative specification must describe a multi-layered system that allows for easy aggregation.It further unifies the topological components, data sources, and the color mappings. As such, it allows users to easily modify and customize their visualization, and to share it with others easily. At the lowest level, OpsVis represents individual entities, the smallest items that are parts of the conversation. At ABCH, these are databases and servers. (These are not the same because a single logical database is physically located on two or more servers). At the highest level, there must be ways to represent clusters, the logical groupings which aggregate these entities.

The configuration file is written in XAML, an XML dialect. XAML is a serialization language for Microsoft's .NET family of languages. It represents an object hierarchy conveniently and directly, making it easy for users to interpret the configurations they have generated. A sample XML file that generates two front- ends, with each server colored by CPU load, is in Figure 19 (left).

```
<OpsVisRoot>
  <TopoSQLDatasource Name="LoadFrontEnd"
  Connection="Server=SQLUTIL;Database=OpsVis;" >
    <TopoSQLDatasource.Query>
      SELECT AC AS KeyColumn, FrontEnd
      FROM BackendInfrastructure
      WHERE AC = '{0}'
    </TopoSQLDatasource.Query>

<StatusSQLDatasource Name="MachineCPUTimes"
  ApplyTo="AC1FrontEnd, AC2FrontEnd""
  RefreshFrequency="10" DataType="Value"
  MappingFrom="CPUTIME" MappingTo="Background"
  MapConverter="ValueToBrightness"
  Connection="Server=SAMPLERFARM;Database=Samples;"
  AggregateWith="Avg" MinValue="25" MaxValue="35">
      <StatusSQLDatasource.Query>
          SELECT Computer.Name AS KeyColumn,
           snd.SampleValue AS CPUTime
          FROM SampledNumericData snd
          WHERE SampleType='CPU'
      </StatusSQLDatasource.Query>
    </StatusSQLDatasource>

  <OpsVisRoot.ModelCollection>
      <FrontEndModel Name="AC1FrontEnd"
  TopoDatasource="LoadFrontEnd"  Param="AC1"/>
      <FrontEndModel Name="AC2FrontEnd"
  TopoDatasource="LoadFrontEnd"  Param="AC2"/>
    </OpsVisRoot.ModelCollection>
  </OpsVisRoot>
```

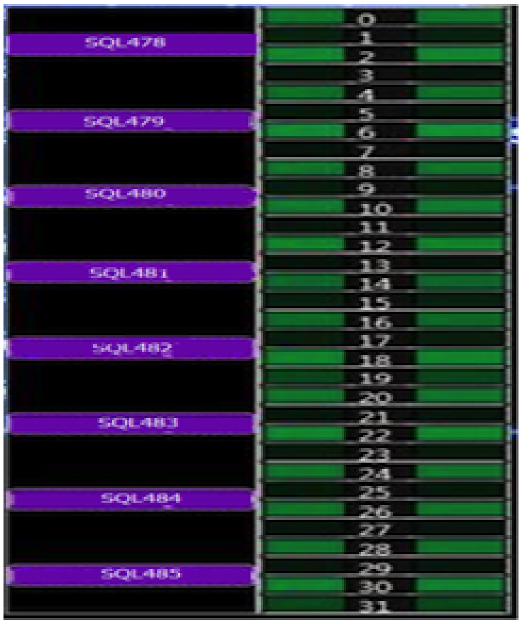


Figure 19: The XML Script (left) for a simple OpsVis configuration file that visualizes two front-ends, and colors them by CPU status information. Detail of OpsVis (right). A set of databases (in green) and servers (in purple). Brighter databases have more rows. The green stripes are a result of a backup cycle as data is moved from one set of databases to another.

Extensibility and customization: Operational problems in cloud services are often found on the backend, particularly in databases. Because the ABCH team was particularly concerned about active database status, they could configured OpsVis to include this information as a primary decoration.

End users could configure the OpsVis's visualizations by tweaking the XAML; while a networking operator might choose to have a configuration that most prominently displays network information, for instance, a database operator might neglect network information entirely for information about their area of specialty.

Because adding a new sort of data to the system as straightforward as writing an SQL query and copying a few parameters, users have found it straightforward to modify the visualization to their needs. This configuration status also means that it should be straightforward for users to share configurations with each other.

The system is generally extensible along several axes. It is most straightforward to connect with different databases and data structures, simply by changing the connection string and the query. The basic Models we have account for a substantial portion of services within Microsoft.

In addition, this research has learned that other network analysis groups desire additional visualization components and status reader types, more precise specification of aggregations, and more precise status visualization. As such, OpsVis is designed an extensibility package for the system: implementing a module that specifies a new aggregation makes it available in the configuration file.

OpsVis Interface

Table 9: Design considerations for the OpsVis Interface

9	Deliver contents in straightforward representation
10	Enable facet filtering for information personalization
11	Interactive content exploration and filtering
12	Aggregate information and show its patterns
13	Display information in consistent format
14	Visualize information from multiple aspects

OpsVis provides a single view of the entire cloud service, as a network. It shows high-level objects, such as clusters of servers, as a single unit. These units are interconnected with network edges, allowing the user to visualize the network configuration and dependencies. Designed following the considerations (see Table 9), each composite object can be expanded to see the next level down: sets of machines, or individual servers and databases. Figure 20 shows OpsVis in action as configured for ABCH. Bubbles and small rectangles represent individual machines and databases, while top-level boxes represent logical clusters of machines.

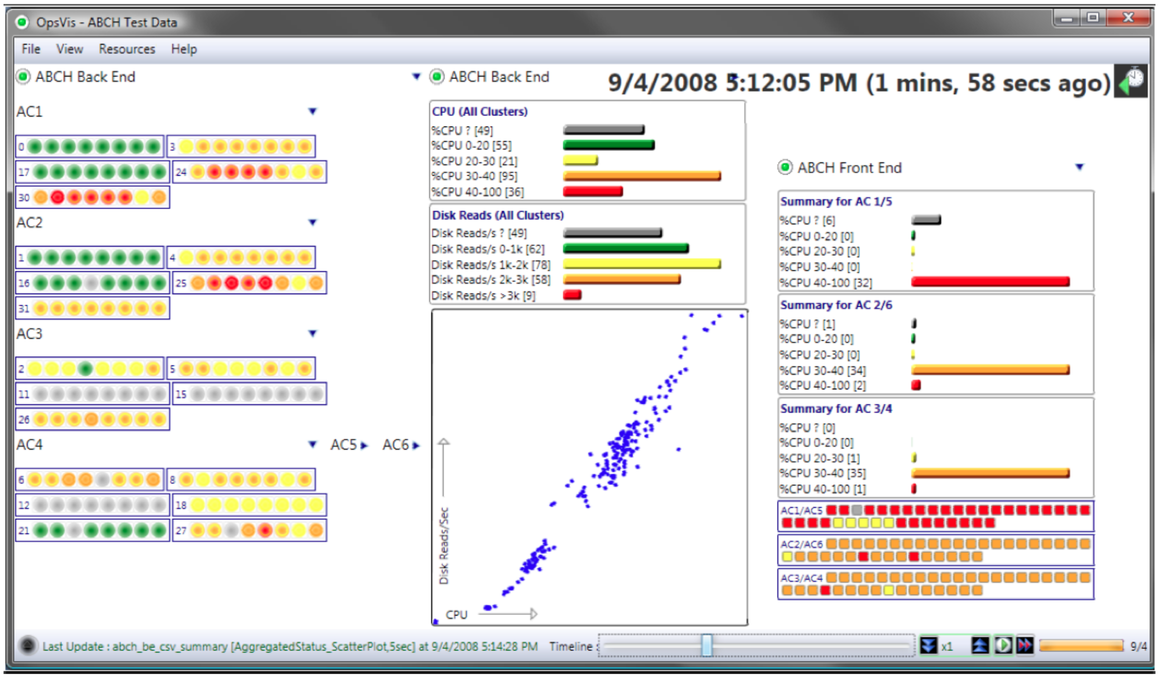


Figure 20: A portion of a display from OpsVis in operation at ABCH. Three different front-end "affinity clusters" each communicate with a pair of back-end database clusters. Each database cluster consists of six databases sets; each set consists of eight servers each hosting four primary and four secondary databases. The visualization represents logical connectivity between servers and clusters . The color of each server in a cluster indicates CPU load. Offline databases are marked red color. Clusters and sets have summaries showing how many databases are online. Tooltips show more information, such as the size of the database.

This layout was developed through several iterations with the ABCH team. On the left, boxes represent clusters of stateless front-end servers. The FrontEnd C cluster is expanded and consists of 34 servers: each server is represented by an oval. In ABCH, the back-end clusters are divided into five or six sets; each set has eight servers which run 32 databases. In Figure 20, one of the back-ends was expanded to show its sets, and Set 18 is expanded to show the servers and databases running on it. Each server, on the left, is a horizontal bar; each database is represented by two boxes showing the “primary” and “secondary”.

Within this framework, the next task is to overlay usage. Here, each server is being monitored for its CPU load. CPU load is drawn on a color scale from black, through saturated color, to white: machines with a higher load are drawn closer to white. The visualization makes immediately clear that four machines are running at below the expected minimum level.

Each database is monitored for status; inactive databases are highlighted in red. A little bit of exploration shows that all eight databases that are highlighted in red come from the same server; however, that server is still running (it is not grayed out, which would signal no “updates”).

The images are animated, with decorations and displays such as CPU load and activity information updating continually as the data changes. OpsVis also provides a time-loop feature so that past behavior can be replayed at high-speed (similar to weather map animations). This helps operators discover recurring and time-dependent behaviors.

Visualizers: The visualization drawn by OpsVis is based around three concepts. The first is a service topology represents the “bones” of the service architecture, the clusters of entities with similar roles, and the relationships among those groups. The second are visualizers, which are responsible for rendering each part of the topology in a manner appropriate for that type and number of machines. Third, data sources provide the raw information used to populate the topology, the relationships between machines in the topology, and the state of the machines that is visualized over the topology.

Each entity that OpsVis knows about is represented by a visualizer. A visualizer is the graphical element that will be rendered, and can carry on it some status information. OpsVis has sought visualizers that allow the eye to correlate across multiple objects in the same role, as well as dependent objects in different roles. For example, in Figure 20, every database and server is represented internally by an Entity Visualizer, which is drawn as a simple object: a circle, or a rectangle. For the collections of elements, a visualizer is drawn as a compound object, including its children.

Every visualizer must be able to be assigned a background color, a string of text, and a series of key/value pairs in its tooltip. Aggregate visualizers aggregate the values of their elements, and thus must also be able to represent the aggregated information of their child entities.

In addition, every visualizer has a unique name, corresponding to the real- world name of the component or grouping it represents. Visualizers are not explicitly represented in the configuration file: instead, they are implicitly created by their internal models. In Figure 18, visualizers are in gray boxes along the left side.

Summary OpsVis

Cloud services are designed and implemented as networks of distributed systems. The underlying distributed systems are characterized by rapid change (in infrastructure, software, and workload), and by use of replication of components (such as servers, data bases, and switches) as the key to scaling out to meet demand in a high performance and reliable manner. Monitoring these systems effectively and economically is a major challenge. Subtle problems inevitably arise in cloud services that impact user perceived performance, and that, unfortunately, are extremely hard to detect, localize, and diagnose.

OpsVis helps to meet these challenges by enabling developers and operators to create visualizations that provide insight at a glance into anomalies and variability across the systems. It provides a perspective that matches the way cloud service developers and operators think about their systems.

3.5.2.3 Case: IRSV for Department of Transportation

Based on the requests of state DOTs as described in section 3.3.1.1, this research has resulted in an interactive visual analytics system (Figure 21) that supports a bridge manager's decision-making process and remains customizable to fit an individual manager's task routine. During a nine-month period, over ten functional prototypes were created, including various changes to the visualization and interface designs. Over the course of the collaboration in the past two years, the prototyping has resulted in a final set of variations of the system. These all focus on providing support for bridge management using integrated remote sensing and visualization, so they are generally referred as IRSV. The following demonstrated is the desktop version of IRSV, which have been widely deployed to multiple state DOTs.

The design of the IRSV system is based on coordinated multiple views (CMV) [133], as well as a modular software architecture that supports customization of the system depending on the bridge manager's preference. It is designed to provide examination of heterogeneous data sources and facilitate effective bridge maintenance planning.

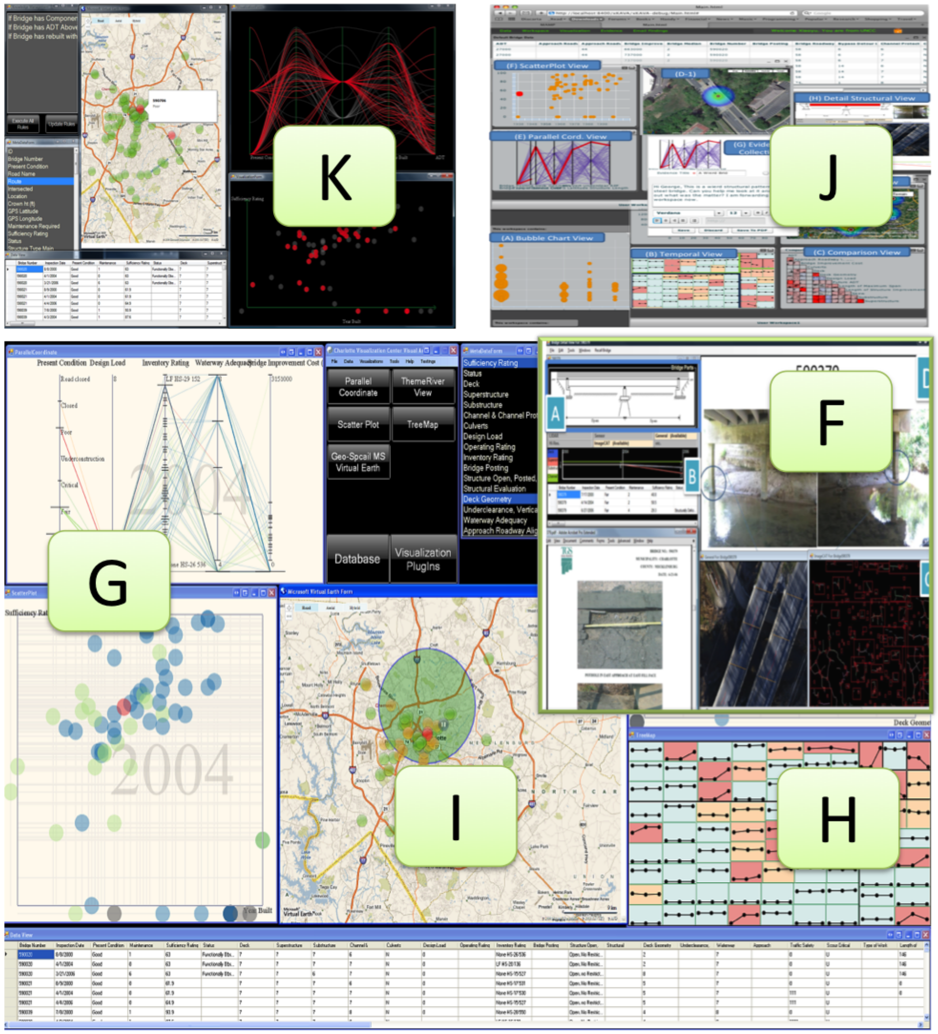
Since previous two projects have led to a more or less complete discovery of the design considerations, the user-engagement and initial system design is much smoother then before. This leads to a comprehensive understanding of the targeted domain and a more efficient and effective system implementation.

Supporting Integration of Heterogenous Inspection Data

Table 10: Design considerations for the IRSV Data Management

1	Unified content interface
2	Integrated multiple information channels

While this is an important issue for the bridge managers, solving it begins with designing new data structures for the BMS. Currently, given the rigid nature of existing BMSs, supporting data integration would require an overhaul of the designs


IRSV: Facilitating Bridge Maintenance Planning

Applied Design Guidelines for Visual Analytics

- ✓ Unified content interface
- ✓ Integrate multiple information channels
- ✓ Easily accessible cross platform app or web-portal
- ✓ (Optional) Communicate with organizational knowledge base system

- ✓ Deliver contents in straightforward representation
- ☐ Enable facet filtering for information personalization
- ✓ Interactive content exploration and filtering
- ☐ (Optional) Employ sophisticated data structures

- ✓ Aggregate information and show its patterns
- ✓ Display information in consistent format
- ✓ Visualize information from multiple aspects
- ✓ Construct coordinated views for linked information
- ✓ (Optional) Incorporate elements from organizational knowledge base

- ✓ Allow evidence collection and annotation
- ☐ Support storytelling and interactive grouping of the evidence with users' reasoning logic
- ✓ (Optional) Trace interactions and system usage for future automation

- ✓ Support sharing of evidence and hypothesis
- ☐ Enable in-app collaborative editing
- ☐ Present status update of collaborative threads
- ✓ (Optional) Real-time collaborative communication or report generation templates

- ✓ Customize personal workspaces and preferences
- ☐ (Optional) Decomposing evidence to generate new knowledge base elements
- ☐ (Optional) Manually update knowledge base systems

Figure 21: The design considerations for IRSV. IRSV contains multiple analysis views, including (F) Detailed structural view, (G) High-level structural view, (I) Geospatial view, and (H) Temporal analysis view. In addition, IRSV provides two variations: (K) a knowledge base integrated system and (J) a web-based system

of these BMSs.

As shown in Table 10, the design of IRSV has taken consideration of incorporating heterogenous data sources. It approach enables bridge managers to combine the traditional National Bridge Inspection Standards (NBIS) dataset with their locally collected information. Currently, IRSV is implemented to help NCDOT bridge managers to associate bridge structural information with extensive data collected in the North Carolina region. This extensive information includes, as shown in Figure 23, field inspections imageries, Light Detection And Ranging (LIDAR) scans for each structure, and pavement crack analysis results.

Supporting Decision-Making Process through Multiple Coordinated Visualization

Table 11: Design considerations for the IRSV System

3	Aggregate information and show its pattern
4	Deliver contents in straightforward representation
5	Interactive content exploration and filtering
6	Aggregate information and show its patterns
7	Display information in consistent format
8	Visualize information from multiple aspects

Following the general design considerations (see Table 11), a set of visualizations is implemented to help bridge managers organize and analyze their assets from the multiple perspectives essential to their decision-making process. As seen in Figure 21, these visualizations were designed to perform the three high-level analyses: structural analysis (G), temporal analysis (H), and geospatial analysis (I). For lower-level tasks, a structural detail view (F) is developed to automatically link information between each bridge component, and provided bridge managers with an intuitive visualization to interactively analyze specific corresponding information. All of these visualizations are tightly coordinated together in such a way that an action performed in one view affects all other views.

The following sections describe the details of these implementations:

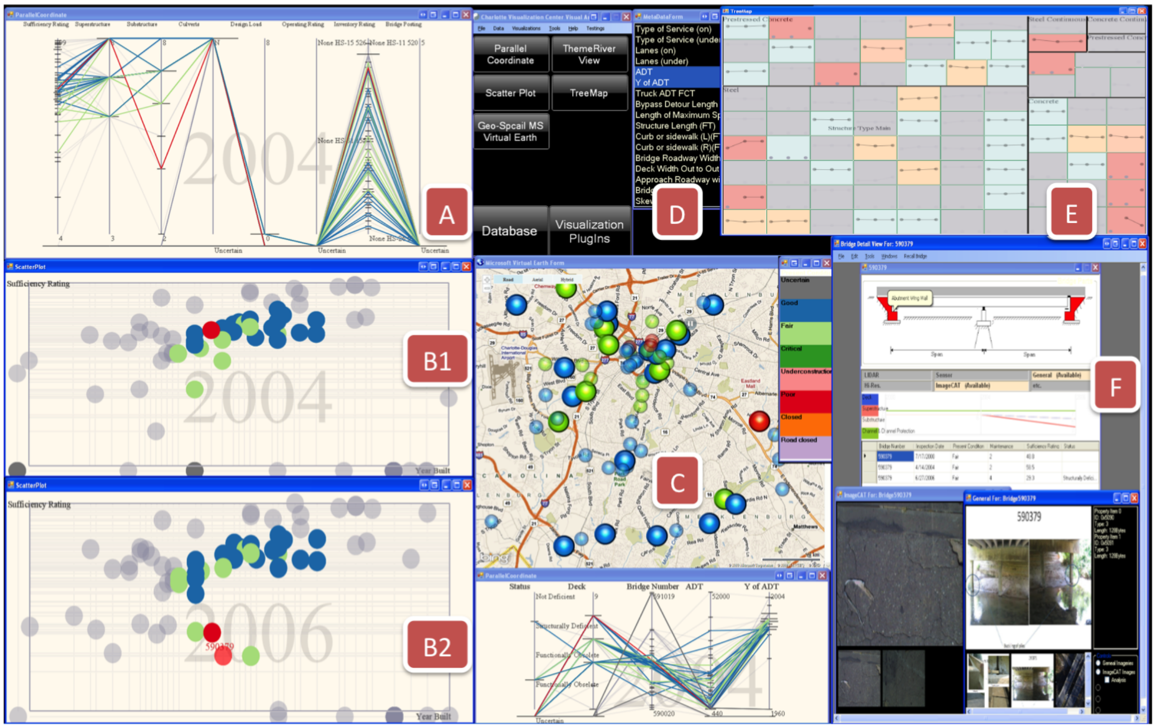
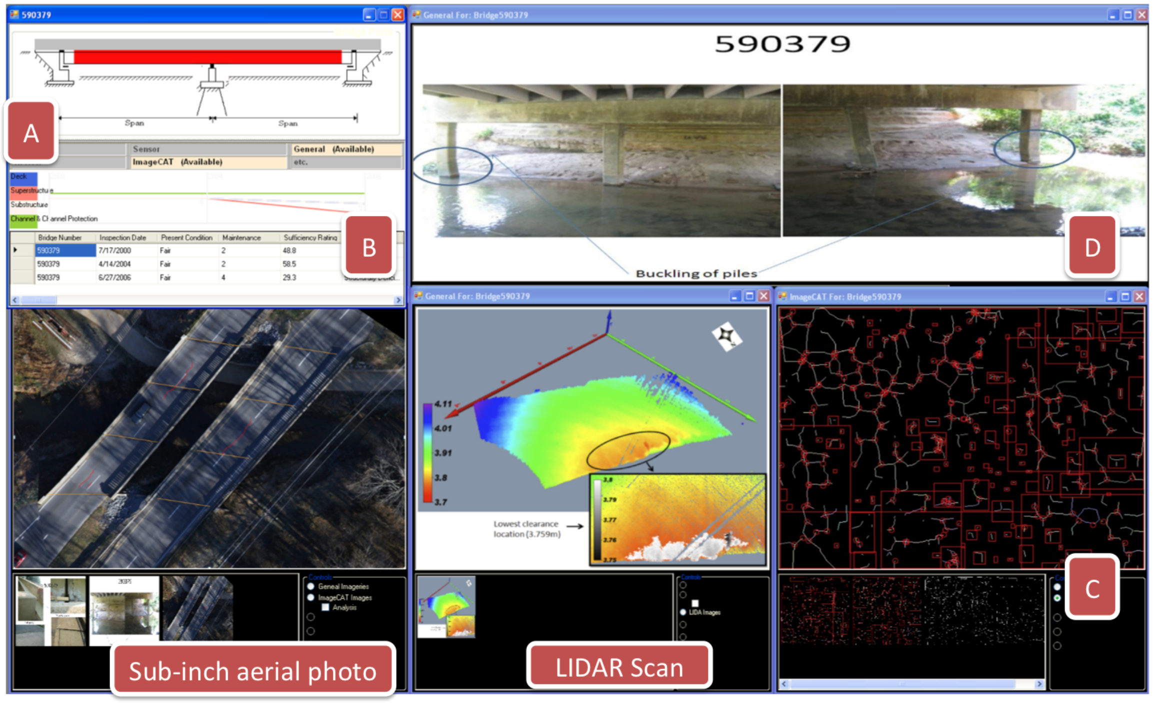


Figure 22: This is the overview of the entire system, including views for Microsoft Virtual Earth(center), Parallel Coordinates (top corners), Scatter Plots (middle left), Temporal Analysis (third row right), and the original data (bottom row). Per-Bridge Detail View(middle right). Several items are highlighted with colors.

High dimensional structural analysis: IRSV system includes three views for helping bridge managers to analyze bridge structures on both a high-level overview and a low-level detail view. On the high level, it utilizes both a parallel coordinate view (PCView, see Figure 22 (A)) [114] and a scatter plot view (SPView see Figure 22 (B)) [156] to help bridge managers detect and identify causal relationships and trends in the data variables. The nature of parallel coordinates limits the number of dimensions that can be effectively displayed at a time. IRSV alleviate this issue by providing control panels to allow the user to select the dimensions of interest (Figure 22 (D)). These dimensions are determined by individual expertise and bridge managers' areas of focus. For example, a structural engineer is more likely to examine bridges based on the structural related dimensions, whereas a planner would focus on analyzing dimensions that represent balances between costs and potential bridge improvements.

Using this view, bridge managers can therefore find correlations in specific sets of the bridge's attributes.

On the other hand, the SP view is designed to depict relationships between bridges across two specific dimensions. The spatial layout of the view allows the user to see clusters and clearly identify outliers, and is a slightly more intuitive interface than the potentially complex PC view. In addition, given the importance of time in bridge analysis, IRSV also extends the ability to see temporal changes in both views, which in turn allows bridge managers to interactively explore and compare information from different inspection cycle. For example, Figure 22 (B1) suggests the sufficiency rating distribution in year 2004, while Figure 22 (B2) shows the distribution of same group of bridges in year 2006. Together, these two visualizations give bridge managers the ability to see high-level trends and patterns in the data's variables.



	Bridge Number	Inspection Date	Present Condition	Maintenance	Sufficiency Rating
▶	590379	7/17/2000	Fair	2	48.8
	590379	4/14/2004	Fair	2	58.5
	590379	6/27/2006	Fair	4	29.3

Figure 23: The Detail View for Bridge (A) An interactive Bridge Schematic Diagram; (B) A Line graph for monitoring temporal changes for major bridge structures; (C) Image Analysis Results for cracks on pavements; (D) Inspection Imageries that suggests the structural damage of the supporting piles of this bridge.

On a detailed level, when inspecting a single bridge, bridge managers need to examine both the overall structural integrity of a bridge across multiple variables, as well as focusing on particular structural components inside that bridge. Therefore, IRSV is designed with a structural detail view to automatically link information between each bridge component and provide bridge managers with an intuitive visualization to interactively analyze the corresponding structural information.

Based on existing bridge design recommendations [46], we model general bridge components into an interactive bridge schematic diagram (see Figure 23 (A)). In this diagram, bridge managers can directly select the major bridge structures, and analyze each component individually. In addition, a line graph enables bridge managers to monitor temporal changes for individual bridge structures. Associated with overall temporal information presented in the small multiples view, this structural temporal component helps bridge managers to gain insight into the effects of structural changes, and to efficiently identify the key factors in the overall deteriorations.

Small multiples for temporal analysis: Bridge managers have expressed the need of having a tool to help them analyze the temporal changes of bridge data. They want to be able to perform analysis over time on a large number of bridges as well as one bridge at a time. Thus, IRSV utilizes a small multiples view [156] to help them achieve temporal analysis of large number of bridges. The design of this temporal view is based on small multiples views in the literature [87], and it shows deterioration changes of each bridge using trend lines.

As shown in Figure 24, each cell in this view represents a single bridge, while the inside line graph represents the bridge's overall rating in all inspection cycles. These cells are further sorted based on the standard deviations of the y axes in the line graph to determine the color of the cells, with warmer color representing sharper changes over time. We note that in this approach, bridges with either downward and upward trending in structural attributes will be colored with warmer colors. Although it is

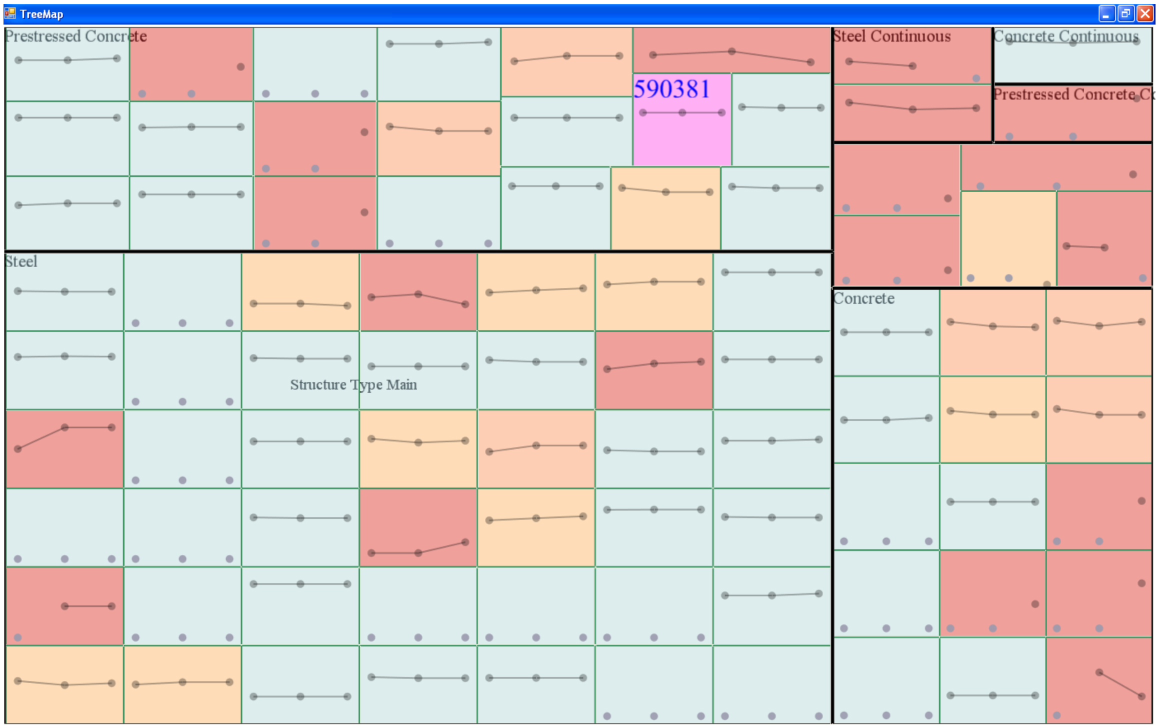


Figure 24: The Small Multiples view with Squarified Treemap layout. Bridges are grouped by their main structure types. For each cell, the x axis represents different inspection cycles and the y axis represents the structural attribute values selected by the user.

not often that bridge managers need to review more than three inspection cycles (6 years), this small multiple views can still efficiently represent all the inspection cycles by changing the line graph to a trail enabled bubble chart, which is similar with the work by Robertson et al [134]. A control panel is also provided to bridge managers to modify the mathematical functions used in highlighting the cells.

Additionally, since it is often necessary for bridge managers to understand the temporal patterns for a certain group of bridges, IRSV adopts a customizable Treemap [19] spatial layout to group the small multiples based on particular structures. There were two main design considerations on utilizing this layout. On one hand, Treemap layout for its inherent advantage in displaying large scale of data, as demonstrated by Bederson et al. [11]. Since bridge managers normally need to monitor hundreds of bridges together, this salability is useful for them to effectively compare difference bridges and see overall trends.

On the other hand, IRSV utilizes the Treemap layout to indicate the size and groupings of different bridges. For example, Figure 24 shows the bridges divided based on their construction material (note the black lines separating regions of the treemap). In this example, the layout enables bridge managers to discover the uncommon temporal pattern where several recently built, known-to-last, concrete structure bridges show significant deterioration. It is therefore mentioned by bridge managers that the capability in finding such insight is not only valuable for their maintenance decisions, but also can help optimize their future construction planning.

While there is still much to be improved for the small multiple view, bridge managers have already seen usefulness in utilizing it to analyze temporal trends and patterns among the bridges.

Geospatial analysis: Extensive research on geospatial visualization [172, 51] have shown the benefits of utilizing online map systems such as Google Maps and Microsoft Virtual Earth. IRSV utilizes Microsoft Virtual Earth (MSVE[113]) to provide bridge managers with dynamic and interactive geospatial analysis (see Figure 22 (C)). By placing the bridges onto the scalable map, detailed geographic relationships and patterns immediately become apparent.

By adopting online map systems such as MSVE, IRSV can have the most up-to-date geospatial information such as road structures and 3D building models. However, IRSV extended MSVE to overlay large amounts of (proprietary) geo-coordinated information over the map, such as traffic distribution patterns and satellite images, and can utilize that information to perform extensive geospatial analysis.

Supporting Domain Knowledge-base

To utilize the existing bridge management technology, IRSV is also designed to incorporate the externalized domain knowledge based into the interactive exploration and analysis process. As shown in Figure 25, a well-designed knowledge database plays an important role in supporting the knowledge internalization, externalization, collaboration, and combination processes. In order to design a useful visual analytics system that incorporates knowledge, a tightly integrated and well-designed knowledge database is considered to be essential in the design process.

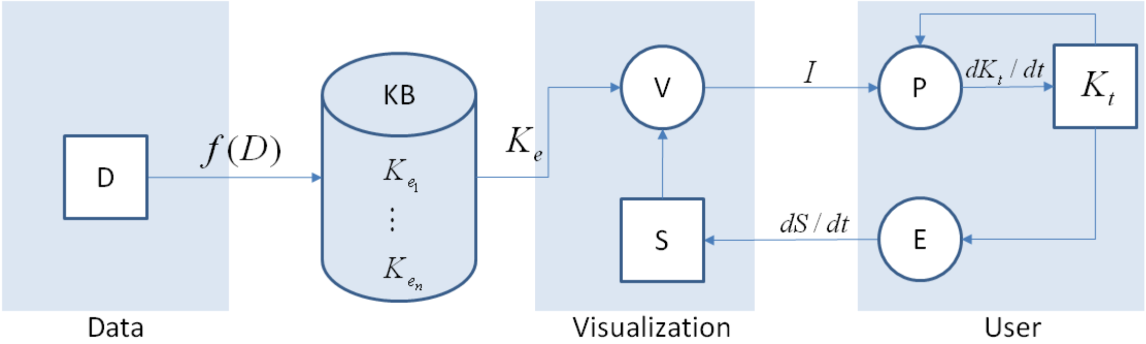


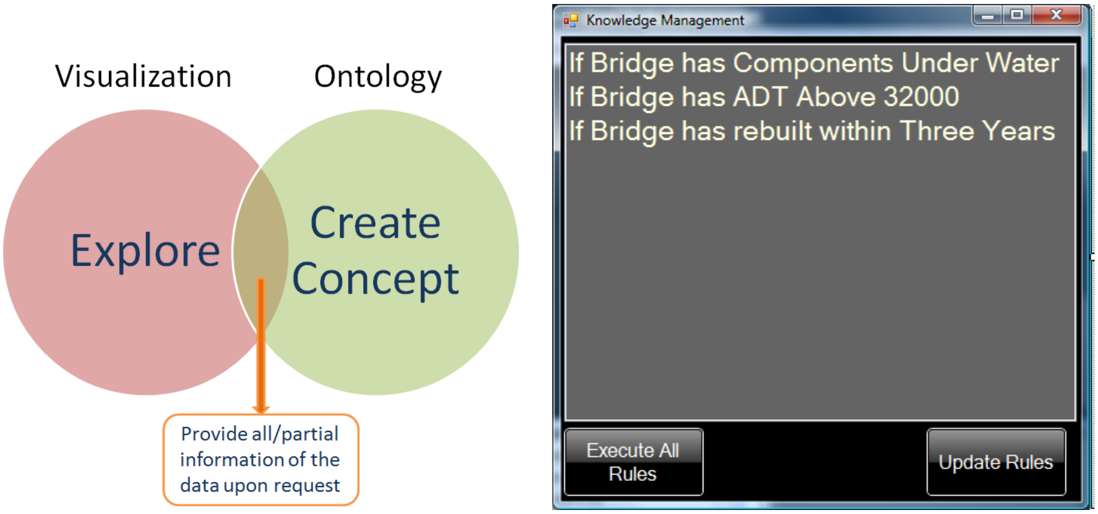
Figure 25: A graphical representation showing four entities: data, knowledge database, visualization, and user. Once explicit knowledge is extracted from the data. It is stored in a knowledge database (KB) and used in visualization to represent it to a user. The user continuously perceives the image and gains tacit knowledge.

There is, however, no definitive way to construct a knowledge database. Much research has focused on designing and developing different forms of such databases that could represent domain knowledge. The differences between these database are not only reflected in their capacities, but also in their structural complexities. As shown in work by Garg et al. [53], a knowledge database could be as simple as a textual structure that contains inductive logic programming equations. On the other hand, it could also be described by extensive decision models, such as Markov decision process (MPD) in the Artificial Intelligent field. IRSV is designed to apply an ontology for storing and retrieving domain specific knowledge.

The ontological knowledge structure is a conceptualization of domain knowledge which includes concepts, properties and their relationships. This conceptualization process aims to transfer both human tacit knowledge and explicit knowledge into computer-understandable formats. These concepts can be further utilized to facilitate other users' problem-solving processes. More specifically, a Problem Domain Ontology

(PDO) enables solving a complex problem where the underlying domain concepts have high interdependencies by building up a problem scenario based on concepts, properties and features in the ontological knowledge structure.

Although research on ontological knowledge structure have advanced in the recent years, integrating such structure with visual analytics system is still an open research area. In the following subsections, the dissertation briefly introduces the understanding about how to integrate these two components is presented first, and presents the prototype of a knowledge-assisted visual analytics system.



If Bridge has Components Under Water
If Bridge has ADT Above 32000
If Bridge has rebuilt within Three Years

(a) (b)

Figure 26: (a) the Venn Diagram that suggests the complementary relationship between visualization and ontology. (b) the knowledge window provides updated knowledge rules inside the ontological knowledge structure.

The Relationship between Visualization and Ontology

In order to integrate a visualization with an ontological knowledge structure, it is important to understand what their relationship is and why the integration is meaningful. By examining visualization and ontological structure separately, a complementary functional relationship is discovered between these two components when represented as a Venn diagram (Figure 26). As shown in the overlapping region of the Venn diagram, both visualization and ontological structure share similar functions that could provide specific information in the forms of visual selections and data queries respectively. Due to the different foci and strengths of the two approaches, the functions of the visualization and the ontological knowledge structure are not always the same. Visualization, on one hand, usually allows the user to interactively explore patterns of the underlying data from various perspectives; the ontological knowledge structure, on the other hand, focuses more on representing the conceptualization of domain knowledge and the interdependencies among the concepts.

A further analysis of this complementary relationship suggests that the integration of these two could be beneficial. If reasonably integrated, users could discover new concepts and knowledge through exploring the visualization and externalize such knowledge into the ontological knowledge structure for future references. Users could also directly access the knowledge structure to acquire predefined domain concepts and rules to guide them through visual explorations and assist their decision-making processes.

Therefore, IRSV is designed to integrate visualization and ontological knowledge base and to utilize the encapsulated knowledge from a domain specific ontological knowledge structure.

The Ontological Knowledge Structure

An ontological knowledge structure is integrated in IRSV to provide domain concepts and information. Using an ontology-driven modeling approach [103], this ontological knowledge structure contains bridge domain concepts, such as bridge structural types and locations. These individual bridge concepts are further connected through their interdependent relationships, which is modeled based on the experience of bridge managers and other domain users. By connecting concepts in such a manner, additional domain rules can be identified and created. For example, a rule can be described as: if a bridge's sufficiency rating is below 50 and its super-structure rating is less than 5, this bridge has potentially undergone severe structural damage. This rule is then stored in the knowledge structure and can be executed upon request. Utilizing such a rule-based ontological knowledge structure allows for great flexibility for IRSV to support precise examination of bridges and enables the system to better facilitate bridge management processes.

Communication between components

Through a server-client web interface, IRSV tightly coordinates the visualization interface with the ontological knowledge structure. Since these two components share the same underlying bridge ID number, the message passing becomes clear and feasible. For example, any results from the executed rules in the ontological knowledge structure will be immediately updated in each visualization window. Thus, exploring within visualization could lead to new concepts that can be further added into the ontological structure; while the knowledge stored in the ontology could assist decision-making during the visual exploration.

To assisting bridge managers in executing the domain rules, IRSV presents an interactive knowledge window (Figure 26(b)) which is automatically synchronized with rules within the ontological knowledge structure. With these two components tightly integrated together, the users always have access to the most up-to-date rules and concepts. The users simply have to execute the relevant rules, and they can see and interact with the bridges in further detail immediately in the visualization environment.

Furthermore, IRSV enables the bridge managers to directly modify the knowledge structure. This function provides bridge managers an important interface to update the externalized knowledge and maintain its accuracy. Based on their discoveries during their interactions with the visualization, bridge managers could create new concepts or rules and directly insert them into the ontological knowledge structure. For example, through their interaction with visualization, bridge mangers may find that the combination of low ratings (less then 4) on both “supporting structure” and “water adequacy” suggests water erosion and flood damage. The bridge managers could then insert this new discovery into the ontological knowledge structure and further re-apply it to check how many bridges have been affected by water-erosion or damage.

Embedded Knowledge Processes

There are four different knowledge conversion processes - internalization, externalization, collaboration, and combination in IRSV’s knowledge-integrated functions. These functions are corresponding to Nonaka et al. [119]’s four knowledge conversion processes, and are depicted specifically in the context of visual analytics [167].

- The Internalization process embodies the transfer of knowledge from a computer to a user through the interactions with a visualization. In IRSV, this process mainly happens through the user’s interaction with the coordinated visual analytics views. These views help the users inspect the data from different perspectives and assist the potential discovery of unexpected data patterns and trends that could become new domain knowledge.

- The Externalization process happens upon the user’s acquisition of new domain knowledge or information that does not already exist in the ontological knowledge structure. This knowledge could come from both discoveries from interacting with the visualization system or from collaborating with other co-workers. Once acquired, user could directly insert this new knowledge into the ontological knowledge structure to augment its knowledge database. The ontology will then store this knowledge and re-apply it during a user’s future investigations.

- The Collaboration process takes place when a user interacts with our integrated system that incorporates domain knowledge of multiple experts. Through the integrated knowledge interface, each bridge managers connects to the same ontological knowledge structure. New knowledge or domain rules created by one manager would immediately be reflected in another bridge manager's visualization system. In this manner, through the use of the ontology as a central repository of knowledge, IRSV facilitates collaboration between multiple bridge managers.

- The Combination process occurs when inserting new knowledge into the existing knowledge structure. The new knowledge could come from a new set of domain data, new perspectives or regulations on bridge inspections, etc. Since bridge inspection rules vary for different inspection cycles due to new federal bridge inspection guidelines or regulations, the Combination process is particularly important in ensuring that each bridge manager is inspecting their data with the most suitable domain knowledge. For example, to handle changes in the standards of water adequacy, IRSV combines different sets of that criteria and applies them accordingly to different inspection cycles.

Summary IRSV

Maintaining bridges is a multi-faceted operation that requires both domain knowledge and analytics techniques over large data sources. Although current bridge management systems are very efficient at data storage, they are not as effective at providing analytical capabilities.

This dissertation presents an interactive visual analytics system that extends the capabilities of current BMSs. As shown in Figure 22, the IRSV system was designed in collaboration with bridge managers in national, state, and local DOTs, and has been implemented specifically to provide them with interactive data exploration, cohesive information correlation and domain-oriented data analyses. The IRSV system enables bridge managers to customize the visualization and data model to fit each individual's task routines.

In addition, based on the understanding of knowledge base systems and the four knowledge processes, IRSV is also designed to incorporate domain ontological knowledge source. It allows bridge mangers to interactively analyze the data with access to the expert's domain knowledge.

3.5.3 Summary: Recommendation 3

This section introduces the third recommendation in the design of an organizational visual analytics system. This recommendation presents the use of iterative prototyping in transforming the actionable knowledge into the design considerations for a visual analytics system. Three successful visual analytics systems are described and discussed in this section to demonstrate the utility of this transformation process. This section further concludes a list of possible design considerations (see Figure 11) to encapsulate the general domain task activities into the design of a visual analytics system.

3.6 Recommendation 4: Design for individual's Analysis Practices Integrations and Customization

3.6.1 Design Considerations to Achieve User-centric Refinements

This dissertation initially thought of the organizational visual analytics design as a holistic decision that could be informed by expert representatives. It instantiates the design by gathering and coordinating organizational knowledge and analytical processes based on inputs from various analysis groups. The first stage in the proposed design framework follows this approach, and concentrates on synthesizing these inputs into a single, unified, and consensus perspective. In this stage, the design framework thus presents both general analytical workflow model and detailed design recommendations to help promoting this synthesis.

Over the course of these projects, however, this research learned that individual domain analyst did not always share the same perspectives on analysis workflows. Although the aforementioned general workflow is valid in presenting the synthesis of the majority of domain analysis activities, individuals may have different opinions on how these activities are carried out. At the heart of these diversified analysis routines are the different combinations and sequences of the above analytical processes. For example, the study with bridge managers 3.3.1.1 revealed that the bridge managers often need to develop their own analysis routines. Depending on available resources, a bridge manager's strategy can be very different from his/her peers', and would require a different combination of the above analysis processes. In addition, sometimes even the same manager needs to take alternative analytical approaches due to changes in priorities.

Consequently, this research recognized the need of a "feedback" process to integrate the individual's analytical practices with visual analytics systems, and achieve the customization of such system based on different analysis perspectives. Particularly, the second stage of the proposed framework (*User-centric Refinement stage*) utilizes

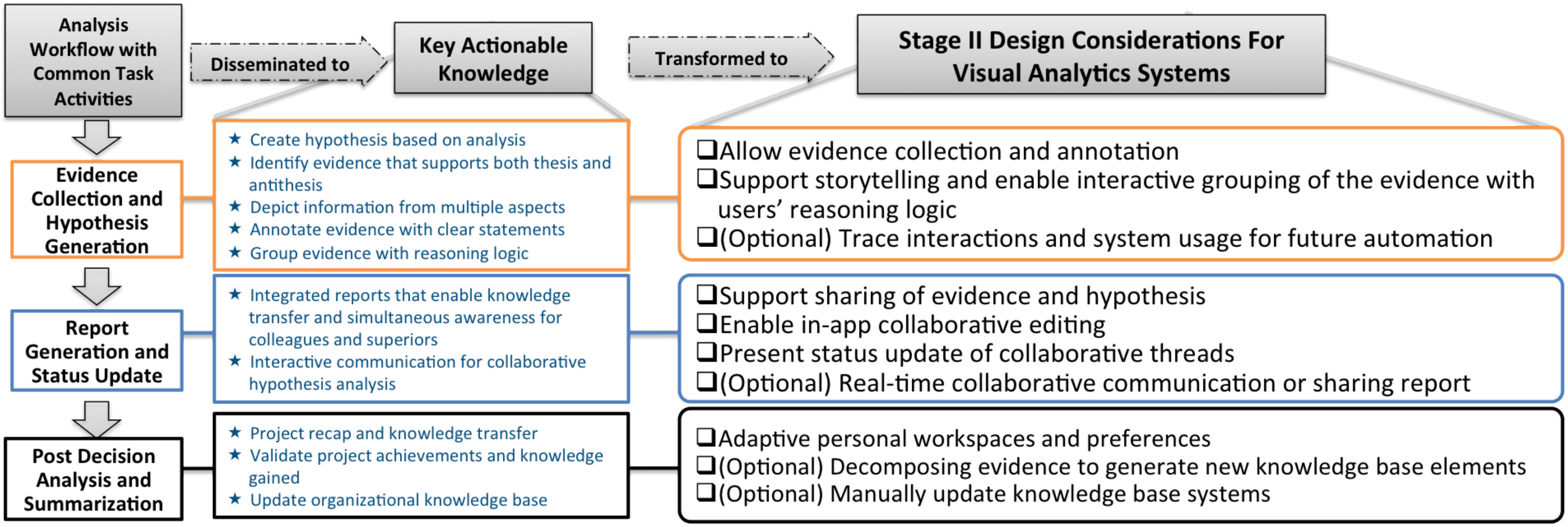


Figure 27: The design considerations for Stage-II, the user-centric refinement stage.

the rich information embedded in users' interactions, and captures and reapply the individual analytical practices.

Prior research on using interactions to model users' analysis process has shown great utility in capturing and recovering analysis processes. For example, by asking human coders to analyze experts' interactions logs, Dou et al. [49] demonstrated the possibility of recovering high-level semantic reasoning processes of domain experts. In addition, by encoding user's behaviors in a visual analytics system, Gotz et al. [58] presented a visual analytics system that can record and reapply users analytical processes. Last but not least, through the use of pre-defined scripting language, the Czsaw system [86] have shown capabilities in capturing and visualizing the users' analysis processes.

In keeping with the need to this "feedback" process, the design considerations expanded. This recommendation focuses on presenting the related considerations (See Figure 27) to enrich and refine domain knowledge through capturing and analyzing knowledge workers' task behaviors. Two essential techniques that can be useful in achieving in these considerations are described and discussed in this section, namely interaction capturing, storytelling and reporting. These considerations have been applied to both the Taste and IRSV system, and extend these systems to encapsulate the users' reasoning practices through either analyzing their storytelling reports or abstracting from their task behavior logs. While the research on this direction is still at its early stage, however, the results from empirical evaluation with domain analysts has demonstrate its efficacy in supporting customized analytical processes.

In the following section, this dissertation will introduce each design practices in details, including the utilized design considerations, the refinement process, and the extension of the for both existing visual analytics systems:

3.6.2 Action: Extending Existing VA Systems to Support Refinement Stage

3.6.2.1 Case: Enabling Analysis Refinement and Knowledge Sharing using Web-based IRSV

3.6.2.1.1 Reassessing the IRSV systems

While the IRSV systems supports data analysis that can assist bridge managers in identifying candidate bridges that are structurally deficient, finding these bridges is just the first step in achieving overall maintenance decisions. An important prioritization process is performed to decide which bridge actually receives maintenance that takes place after the candidates have been identified. Due to the limited resources such as budget and construction timings, this maintenance prioritization is typically determined based on the collective discussions among multiple bridge managers. During these discussions, bridge managers often need to find a balance between limited resources and maintenance requirements to maximize the overall stability of the transportation system as well as the safety of the public.

Unfortunately, such an optimization process is often not well defined, and the maintenance decisions vary depending on the goal of individual bridge managers. The analysis of high-dimensional bridge data is generally performed by bridge managers who have a great deal of experience and special training, both of which are valuable domain knowledge. For example, some managers may focus on repairing supporting structures of bridges, which they believe is crucial to the bridges' structural integrity, while others may spend the resources on fixing the bridge deck where visible damages occur.

Therefore, to maximize the utilization of the dearth resource, the bridge managers are in need of a way to annotate and share their analyses, and balance the requirement resource with others. With the need for a closer collaboration between groups of managers [164], many state DOTs indicate that it would be useful to collect and annotate their analysis findings and bring these findings into the prioritization discussions.

Further, they would like to have the ability to share these findings with other managers to determine the proper distributions of resources.

3.6.2.1.2 Designing a Web-based System to Support Analysis Customization and Knowledge Sharing

It is increasingly noticed that the bridge managers are facing hurdles in sharing the analysis information. This research began designing to influence their collaboration as well as their prioritization of actions to resolve analysis gaps. This research further observed whom the bridge managers communicated with when they have analysis results and whom they involved in prioritization discussions. As shown in Table 12, four design considerations is concluded to support this situation.

Table 12: Design considerations for the Collaborative IRSV System

1	Allow evidence collection and annotation
2	Trace Interactions and system usage for future automation
3	Support sharing of evidence and hypodissertation
4	Real-time collaborative communication or report generation templates

Extending on existing literature on collaborative visualization [69], this research transformed these design considerations into an architecture of web-based interactive visual analytics that is tailored to support the prioritization need. As illustrated in Figure 28, this architecture a web-based service-client model, and provides web-services to establish communications between both client and server side.

Following these design recommendations, the extension of the project is to design a visual analytics environment that can be customizable to support complex bridge management process as well as enabling effective collaboration. As a result of extensive literature research [69, 68] and discussion with bridge managers, this dissertation developed in a asynchronous web-based visual analytics system that is used internally with multiple state DOTs.

On the one hand, the web-based IRSV system is designed and implemented following

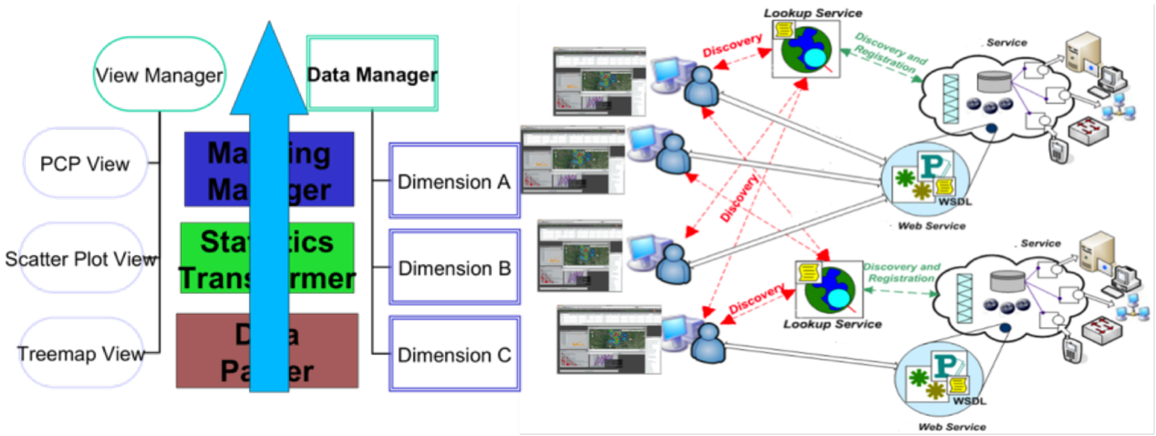


Figure 28: The architecture of the web-based collaborative visual analytics system. (A) The diagram on the left denotes the Client-side architecture for an interactive visual analytics system. (B) The graph on the right shows the overall service-oriented Server-side collaborative environment between individual clients.

the design considerations listed in Section 3.5.2.3. It encodes the three analysis processes into a set of coordinated visualizations, and thus provide bridge managers with their familiar analytical capability. This system focuses on supporting information analysis from the similar aspects as the desktop version (e.g. geospatial, temporal and structural analysis). Note, since most of the data analysis components in the client-side interface follow with same design considerations previously mentioned in 3.5.2.3, this dissertation will not reiterate these designs but will demonstrate the visual changes for web usage in Figure 29.

On the other hand, since analyzing bridge conditions is a sophisticate process, IRSV enables the bridge managers to record their analysis process at every step. With built-in evidence finding mechanism, the users can directly capture their analysis findings and annotate those findings to intermediate maintenance reports. Bridge managers can further share their analysis results with colleagues, and collaboratively maximize the utilization of maintenance resources.

In the following paragraphs, this dissertation describe the key features in this design that helps to achieve the knowledge refinement and sharing:

Customizations of the Domain Analysis Process

ADT	Approach Roadw	Approach Roadw	Bridge Improve	Bridge Median	Bridge Number	Bridge Posting	Bridge Roadway	Bypass Detour L	Channel Protect
27000		44	663000	2	590020		58	6	6
27000		44	737000	2	590020		58	6	7
			737000	2	590020		58	6	7
							58	14	6
							58	14	7
							58	14	7
								1	8

Evidence Title: A Wierd Brid

Annotation

Hi George, This is a wierd structural pattern for a steel bridge. Can you help me look at it and figure out what was the matter? I am forwarding you my workspace now.

Verdana 12

Figure 29: The overview of the entire system of iMonitor client. In this client system, iMonitor provides views for Google Map(D), Parallel Coordinates (E), Scatter Plots (F), Temporal Analysis (B), Bubble Chart (A), Comparison Matrix (C), and the original data (top row). Per-Bridge Detail View (H) and Evidence Annotation Panel(G). Several items are highlighted with colors. (D1) is used in Section 6 as part of the scenario

Table 13: Extended Design considerations for support customization of analysis processes

6	Easily accessible cross platform application or web-protal
7	Customize personal workspace and preferences
8	Construct coordinated views for linked information
9	Trace interactions and system usage for future automation

Follows the design considerations shown in Table 13, the web-based IRSV system is designed to enable customization of individual analytical process. It allows bridge managers to extend the system to incorporate advanced visualizations and additional data models. As shown in Figure 28(A), the client architecture utilizes the concept of coordinated multiple views [133] as well as a modular software architecture to customize the visual analytics system.

Depending on the bridge manager's preference, the web-based IRSV system provides both explicit and implicates ways to support the customization. Explicitly, IRSV enables bridge managers to interactively design and arrange visualizations based on their needs and analytical routines. Each visualization component integrated in IRSV is interchangeable plugins that can be efficiently coordinated based on the identical bridge identification numbers. In doing so, IRSV allows bridge managers to upload their data, and select specific data dimensions for analysis.

Following on the bridge manager's selections of data and visual representation, the mapping between data and the desired visualization are computed on real-time. Thanks to the client-server design, IRSV processes this mapping on the server and utilizes its computation power to support on-fly data aggregation and information correlation. During this process, multiple statistic analyses would be performed to the data for additional information, including such as standard deviation, linear regression and coefficient that are commonly applied to the bridge analysis. IRSV would then stream this computed information back to client side and create customized visualization and interactions accordingly.

On the other hand, through capturing and analyzing users' interactions, IRSV also presents a less intrusive way to customize visual interface for individual analytical processes. With the user's permission, IRSV implicitly collects information about her exploration trails during each analysis session. This information includes visualization parameters (interaction logs and view parameter), temporal event information (e.g. the duration spent on particular visualization, the frequency of using certain views), and data parameters (e.g. what data is used, which dimensions have been focused on).

This collected data is analyzed on the server-side to retrieve reasoning preferences for individual bridge managers. Based on the association between the collected data and its owner, a pair-wise ranking matrix is automatically generated to rank the importance of each visualization, data and the mapping between them both. This ranking matrix establishes the baseline for the user's analysis steps, and provides the IRSV system with a set of visual-data combinations that are associated to that particular bridge manager. It further indicates preliminary analysis sequences that may fit into individual's task routines. Using this ranking matrix, IRSV can utilize the interaction logs to customize the analysis environment, and adjust its date models and visual representations in accordance with the specific preferences for bridge managers.

Since the web-based IRSV has only been deployed recently, the results on how effective our approach is are still limited. With the growth of our online users, this dissertation is expecting to gain more insights and conclude more details about its utility in the future.

Annotating and Sharing Findings

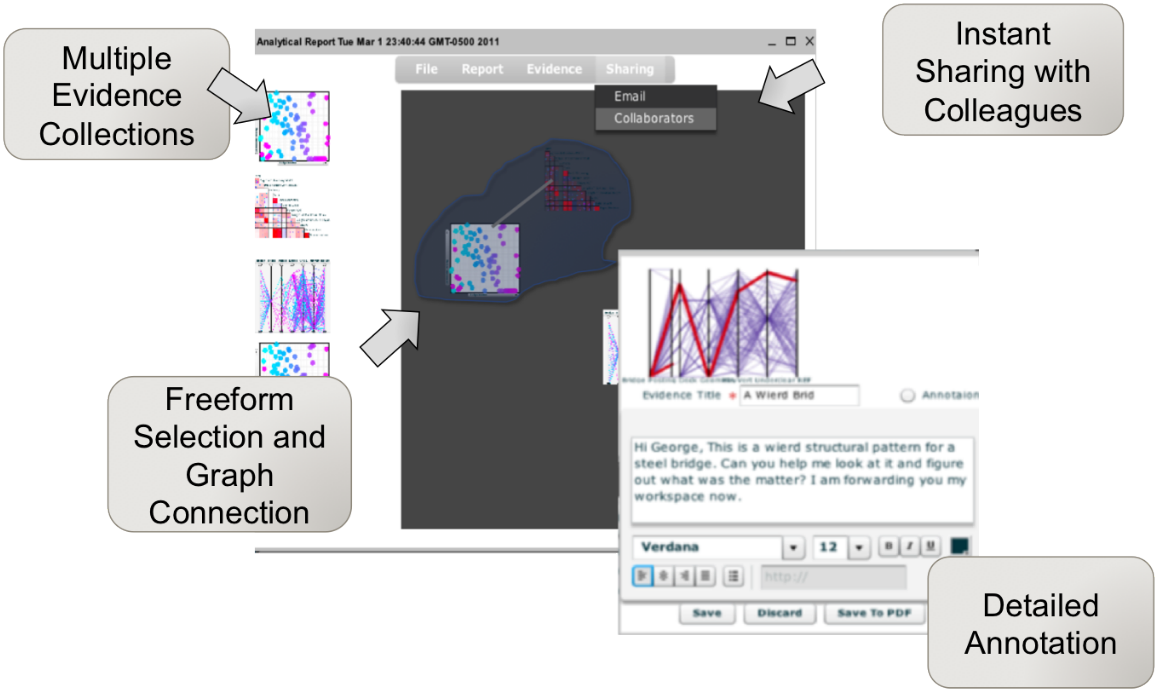
Collaborative visual analytic environments require considerations of a number of factors. One must consider the specific requirements of the problem domain, and what the overriding and imperative questions to be answered are. In the case of

Table 14: Design considerations for annotating and sharing findings

10	Support sharing of evidence and hypodissertation
11	Real-time collaborative communication or provide report generation templates

bridge analysis, this research primarily concerns with the sharing of analysis findings for individual bridge manager.

As such, IRSV chose to not enforce any strict limits on who can do what where, and designed our asynchronous collaborative environment based on recommendations from Heer et al. [69]. Rooted with Scott's [140] observations about intuitive division of collaborative space, this research recognizes that the client-side architecture is obviously the center point for individual bridge manager to perform analytical process. Similar to the Scalable Reasoning System [128], IRSV focuses on the collaborative aspects of organizing the analytical processes among multiple users and sharing their results to enhance the bridge maintenance prioritization process.



Evidence Title * A Wierd Brid

Hi George, This is a wierd structural pattern for a steel bridge. Can you help me look at it and figure out what was the matter? I am forwarding you my workspace now.

Figure 30: Annotation example

In keeping with this design, as shown in Figure 30, IRSV supports interactively collecting, annotating, and sharing analysis findings between different collaborators. Incorporating the listed design considerations (Table 14) into the web interface(see Figure 29(G)), IRSV enables bridge managers to create diversified visualizations (See Figure 29), and allows them to group the visualizations into workspaces. Like a Windows desktop, a workspace initially presents a blank visualization canvas for individual bridge managers to customize their analysis environment. Unlike a Windows desktop, user can create multiple workspaces during the same analysis process, and compare and correlated information that is represented in them.

IRSV further allows bridge managers to use the collected evidence to support their analysis hypotheses and create analysis reports [164]. IRSV treats individual visualizations and group workspaces as collectable items. It enables bridge managers to directly drag and drop these items into a sandbox, designed to collect all the findings. As illustrated in Figure [173], IRSV each collected items to the sandbox, and sort them temporally. Through a predefined template, IRSV further allows bridge managers to annotate each collected evidence. As illustrated in Figure 29, IRSV provides a rich set of annotations tools for bridge managers to directly insert their comments for each evidence, including inserting textual descriptions, and highlighting key patterns in a visual evidence.

To capture the causal relationship between these evidences, IRSV supports bridge managers to interactively create a semantic graphs between individual evidence, and more importantly, it allows bridge managers to combine and organize the evidences based on their own reasoning processes. IRSV further deploys a semi-structured reporting function to help bridge mangers to construct a preliminary analysis report based on these organized evidences.

As illustrated in Figure 28(B), our server architecture follows a server-client model. The architecture aims at supporting services that can enable interpersonal collaborations.

The main functions for this architecture are to guarantee stable hosts of online bridge management community, sharing of information between bridge managers, and unified yet protected data repository. Through a set of predefined web-services, our server-side architecture allows the instant information exchanging between both tiers, and therefore enables bridge managers to interactively collaborate on bridge prioritizing and decisions-making.

Using built-in sharing channels or emails, IRSV helps bridge managers to share their findings and reasoning processes with colleagues. Instead of merely sharing a static image, IRSV records the parameters of each evidence and its semantic connects, and shares this information with other bridge managers to recreate an interactive analysis workspace.

Incorporating Emerging Data Model

Table 15: Design considerations for incorporating emerging data model

7	Integrate multiple information channels
8	Incorporate elements from organizational knowledge base

To adapt to the development of emerging domain technologies, IRSV is built on top of a modular architecture that allows bridge managers to extend the system to incorporate advanced visualizations and more effective data models. This is made possible largely because inspections and analysis results are tightly associated using a unique bridge identification number. Therefore, if bridge managers are in need of analyzing additional data, IRSV is designed to be ready to incorporate this data and provide proper visual analysis.

In practice, IRSV follows the above design considerations, as seen in Table 15, and implemented the server-side architecture to combine and cross-reference diversified bridge datasets upon their arrivals. Serving as a dynamic data repository, IRSV's server allows bridge managers to upload their own datasets onto the server. It further facilitates the users to publish and share their data with colleagues.

Currently, this approach enables bridge managers to combine the traditional National Bridge Inspection Standards (NBIS) dataset with their locally collected information. As of the dissertation, this research has helped NCDOT bridge managers to associate bridge structural information with extensive data collected in the North Carolina region. This extensive information includes, as shown in Figure 22 (F), field inspections imageries, Light Detection And Ranging (LIDAR) scans for each structure, and pavement crack analysis results.

Summary: the web-based IRSV

Following the above design considerations, the web-based IRSV system is implements to support the user-centric refinement process. It supports methods to customize and adjust the visual analytics system, and helps to integrate the individual's analysis processes with the general analytical workflow. This web-based IRSV system not only presents a customizable visual analytics environment, but also it enables bridge mangers to collect, annotate and organize evidences found during their analysis findings. Using these functions, the IRSV system supports the sharing and collaboration among multiple managers, and in turn, facilitates the bridge maintenance prioritization process.

3.6.2.2 Case: Enhance Taste to Refine Individual Analysis Processes and Support Information Sharing

3.6.2.2.1 Reassessing the Taste system

Enterprise strategists have long been aware that the information sharing flow has tremendous influence on business success or failure [20]. A vibrant organizational culture with a strong sense of community and cross-functional network of employee relationships can significantly augment traditional management methods and processes structures [109]. Especially in a large enterprise like Xerox Corporation, where documents and information are generally decentralized among employees, it requires collaborative efforts to achieve efficient information retrieval and decision-making processes. In particular, the follow-up interviews conducted by this dissertation revealed that the success of these collaborative efforts is determined by the responsiveness of the inter-personnel communication, the efficiency of information sharing, and effectiveness of collaborations with personnel hierarchies.

However, previous studies [20, 110] have suggested that such collaborative efforts could be undermined by three major limits, namely, the communication between colleagues may not be timely, relevant information might not be shared, and mostly the collaboration may not occur to the right person or the right organization. Because most current tools lack support for these critical functionalities, enterprise employees often resort to paper formats or email to communicate with other colleagues about the business information that they have found or their need for help finding it. This therefore largely slows down the decision-making process and causes breakdowns the information sharing and collaboration, thus resulting the unwillingness of knowledge workers to share insights and pass along experiences.

Therefore, to retain an effective information flow, the research collaboration with Xerox had been extended to design tools that could enhance such organizational collaboration processes. Specifically, this research extension is conducted based on the evaluation and assessment of Taste with domain knowledge workers, and focuses on extending Taste to support such needed information sharing flow. In general, these professionals expressed their need of tools that not only can effectively help organizing business information, but also would allow them to share their analysis results and collaborate with others in a timely manner.

3.6.2.2.2 Utilizing storytelling to refine and share individual business analysis

According to Pike et al. [129], the typical goal of information analytics is to create new understanding and communicate it to others. This sharing and collaborating factor is especially valuable in a large enterprise environment, where information can be spread out among multiple employees. In such workplaces, seeking information may require an organized effort by collaborating employees. Therefore, Taste supports the information seeking process for both individuals and groups.

By utilizing an interactive Story Telling view, Taste allows users to collaboratively find information, organize it and share it. In this view, the user takes a more active role in information tracking. Like the Detail view, this view is universally supported throughout Taste. Whenever a user comes across an interesting information object in Taste, they can right click on that object to add it to a new or existing Story view.

Once an element is in a Story view, as shown in Figure 31, the user can annotate or tag it. The user can also perform basic grouping and sorting on the story elements. In addition, the user can drill down to more information about a given document or person by clicking on its icon.

Figure 31: A story that has been organized into 3 Story views.

The story created by one user around a collection of people and documents may be of interest to other users as well, so Taste allows stories created on one instance of the system to be shared with users on another instance. Other employees, who receive this shared story, are able to modify it based on their understanding of the topic and add or suggest removal of story elements. By sharing their stories about document activities, groups of employees can now understand those activities better and improve information retrieval for all members of the group.

While the story feature in Taste is quite new, we can already see that it may be useful for a number of applications, including capturing a related set of ideas, building an annotated bibliography, recording the history of a project, or sharing information about recent exchanges with a customer. We are still testing the usability and utilities of this feature in enterprise environments.

By utilizing an interactive storytelling view, shown in Figure 12 (D), Taste allows users to interactively collect evidence, annotate it, and share it with others. The storytelling view allows the user to take a more active role in information tracking, and enables them to express the information relationship based on their own knowledge. Whenever a user comes across an interesting information object in Taste, they can directly add that object to a new or existing story view. Once an element is in a storytelling view, the user can further annotate or tag it, and can group different story elements based on their reasoning logic.

The story created by one user around a collection of people and documents may be of interest to other users as well, so Taste allows stories created in one instance of the system to be shared with users in another instance. Analysts who receive these shared stories, are able to modify them based on their understanding of the topics, and add or suggest removal of story elements. By sharing their stories about document activities, groups of employees can now understand those activities better, and improve information analysis for all members of the group.

3.6.2.2.3 Logging interactions to parameterize the document analysis processes

Another “feedback” loop built in the Taste system is the implicit logging of individual knowledge worker’s interactions history. While seeking business information embedded in multiple document activities is important, an individual document is weighed differently based on the analytical needs and targets. Employees generally have their own methods for organizing and relating business information, based on their job requirements and individual experience. It is therefore interesting to consider whether externalizing such knowledge and embedding it into customized visualizations would be feasible for enhancing their information retrieval processes.

Taste currently logs the interactions that are used in analyzing the document activities. The goal for this interactions logging is to use content frequency and analysis duration information (i.e. time intervals and content ordering) to update and customize visualization parameters. The logged information is used to capture, store and reuse domain analysis knowledge. As shown in Facet view (see Figure 15), the ordering of a document is determined and rearranged by this logged individual interaction information.

In addition, Taste implicitly stores the information for the annotation elements that are created by the users. It concentrate on analyzing the story elements and translating them into knowledge artifacts that can be externalized to document repositories and then used as input to personalized search methods. This implicit annotation logging helps extending the collaboration support, and suggests document information to collaboration groups.

While the interaction feature is new, many employees found the idea of the automatic reordering of information necessary. They felt that the feature was practical and useful. The employees were interested in testing this function in their daily analysis processes, and expected to use this function to increase their productivities.

3.7 Conclusion

In this chapter, this dissertation presents four design recommendations that are useful in the implementation of organizational visual analytics systems. Each of these recommendations covers an individual procedure during the design process; together the four recommendations presents a coherence design process for an organizational visual analytics system. Three successful visual analytics systems are described and discussed in this chapter to illustrate the utility of these recommendations. These three examples systems represent the four recommendations for an important aspect of visual analytics design.

CHAPTER 4: A TWO-STAGE DESIGN FRAMEWORK

4.1 Objectives

The objectives of this chapter are to synthesize a two-stage visual analytics design framework for organizational environments. This framework is concluded based on the four design recommendations that are summarized from extensive collaborations with three large organizations, namely Microsoft, Xerox Corporation and the USDOT. Details of the four design recommendations can be seen in Chapter 3.

This framework presents knowledge beyond the mere summarization of the aforementioned recommendations. It focuses on the concept of a two-stage design process in achieving a comprehensive modeling of a organization's analytical workflows. The two stages in this framework, namely ***Observation and Designing*** stage and the ***User-centric Refinement*** stage, are aims at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their analysis processes.

Since there is no such visual analytics design framework exists in current literature, as suggested in Chapter 2, this chapter presents the building process and details of the proposed two-stage design framework. Every important design component is described and discussed in this chapter. This chapter further introduces the general ways to use this design framework for actual implementations.

This chapter:

- presents a two-stage visual analytics design framework for organizational environments.
- illustrates the detailed design steps of the framework , including its individual component and process.

- demonstrate the design process and the utility of this general two-stage design framework.

4.2 Overview

The modern organizations, including government agencies and commercial enterprises, depend upon timely and effective flows of information and knowledge through its organizations for success. The larger, more time-critical and multi-information channeled an organization, the more that it relies on the efficacy of such analysis workflows [118].

Since different analytical activities in the workflow demand tailored information and require distinctive domain-knowledge, the organizations demand new kinds of decision support systems that incorporate the knowledge workers' reasoning and task-solving processes. In turn, many of the prominent organizations depend upon information support systems to support their analytical processes, in order to excel and compete through innovation more than production and service [79, 111, 3, 117].

As shown in many successful examples in Chapter 2, visual analytics systems have played an important role in accommodating these domain analysis requirements and facilitate their analytical processes. While these outcomes have demonstrated the utility of visual analytics, however, one might say that the contemporary visual analytics field is inherently diversified, lacking of unified theory and design foundations. It also introduces one key question that is critical in presenting the science of visual analytics: "What's the generalization and design foundation of visual analytics?". To identify such generalization, it requires the visual analytics researcher to summarize their success from multiple domains and conclude them into a cohesive design framework that holds the field together.

Despite the variety of systems development approaches that are in practice, this design research is anchored to the identification of the essential components, processes, and techniques that function as the general building blocks or ingredients of various visual analytics systems design and development methodologies. For instance, it considers the fundamentals of visual analytic design from multiple aspects, including the *characterization of general analytical workflows* based on domain analysis, the *basic considerations and techniques* for extracting the domain information (e.g. task and context analysis), the *general visual design and encoding methodologies* for many systems development, the *structural approach* to systematically development a visual analytics system, and finally the *evaluation* to assess efficacy of a visual analytics system.

Based on research on these design fundamentals, this dissertation presents a general design framework to provide future visual analytics developers and researchers with starting point, or the "generational experience" [132], to address various systems development situations.

4.3 The Fundamentals of a Design Framework from a Visual Analytics Perspective

Like many empirical sciences (e.g. HCI, InfoVis), the field of visual analytics does not solely research on existing technologies, styles of interaction, or interface solutions. The design of organizational visual analytics systems concerns beyond merely the final user interface and representations. While these interface features are undeniably essential considerations in visual analytics development, but they are not the major concern of this dissertation.

The core foundation of visual analytics designs is, in the view of this dissertation, rooted in the **generalization of domain analytics processes**, the visual facilitation to domain analytical tasks and finally, the customization for users' analytical workflows. Essentially, the design of visual analytics systems is analysis centric in a way that it needs to encapsulate the organizational *analysis discourse* and support its' related reasoning tasks and user behaviors.

In particular, the design of visual analytics systems needs to consider and alleviate the potential incompatibilities and challenges that could affect organization users' acceptances of, and reactions to, such systems. The design of visual analytics system needs to provide supports for various domain analysis tasks, match properly between the nature of the task and the support from the systems, provide logical organization of the targeted data, utilize the accurate statistics to meaningfully transform data and most importantly, guarantee the consistency between the analytical workflows and the visual analytics system operations.

Human-centered design is another very significant factor in visual analytics design. This recent research and practice in HCI has shown the users affective reactions and their holistic experiences with technology are gaining more attention and becoming increasingly important [4, 169, 178]. As suggested by many empirical studies that, a better understanding of various human cognitive, affective, and behavioral factors involved in user tasks, problem solving processes and interaction contexts is required to address these problems [178].

The central tenet of human-centered design is to fully engage domain users during the modeling and design step of developing a visual analytics system, ensuring their requirements and demands to be clearly understood and conveyed to visual analytics designers. The human-centered design is a bi-direction process: on one hand, the elicitation of system requirement demands visual analytics designers to communicate fluently in same the "analysis language" of the targeted domain; on the other hand, it also requires the designers to introduce the merits of visual analytics to influence the analysis domains and help them to reform the organization of their existing analysis processes.

Both the generalization of core domain analysis and the human-centered design are regarded as the two fundamentals in constructing a general visual analytics framework. Therefore, such framework need to comply with both fundamentals in a ways that: 1) the framework must reveal the generalizability of visual analytics in encapsulating and facilitating domain analysis processes, and more importantly 2) the framework must clearly instruct a systematical development process that guarantees the efficacy and validity of a customizable visual analytics system.

4.4 Developing a Two-stage Design Framework for Designing Visual Analytics Systems

Following Zimmerman et al.'s [179] definition of design research as "an intention to produce knowledge and not the work to more immediately inform the development of a commercial product", instead of intending to produce a commercial product, this dissertation focuses on producing a framework that informs the design of a visual analytics system.

Similar to Hirschheim and Klein's [73] view on developing an information system, this dissertation proposes a two-stage framework for systematically designing visual analytics systems in organizational environments. As shown in Figure 32, the proposed framework concentrates on incorporating both the generalization of domain analytics processes and human-centered designs into the conceptualization of the framework. More importantly, this dissertation conflates the other two essential fundamentals in a cohesive manner to augment the design of a visual analytics system.

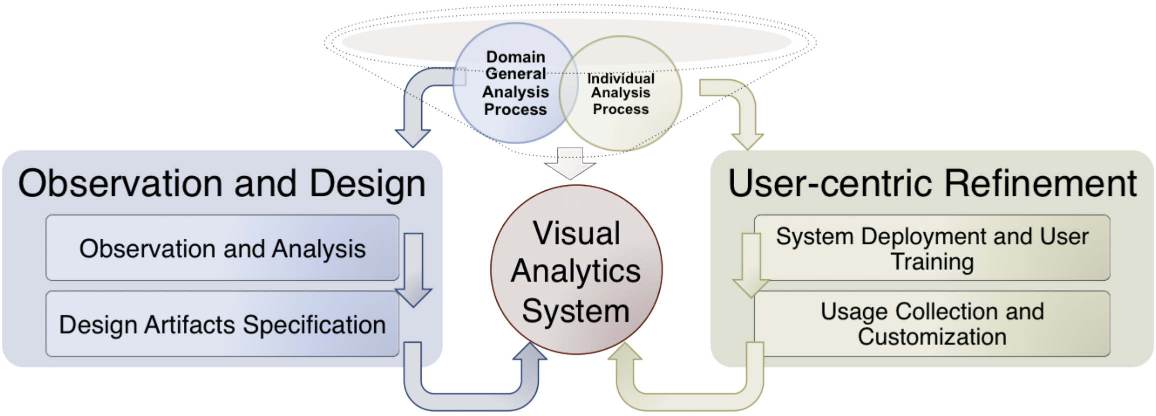


Figure 32: An overview of the targeted framework. The two stages in this framework emphasizes on the incorporation of both general domain analytical process and individual analysis approaches

In particular, the first stage in this framework is an ***Observation and Designing*** stage (OD-stage) (Section 4.5.1), in which a visual analytic system is designed and implemented to abstract and encapsulate general organizational analytical processes. As detailed in Section 4.5.1, there are typically a two design steps used to achieve this stage. The second stage is the *User-centric Refinement* stage (UCR-stage) (Section 4.5.2), which aims at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their analysis processes. Two design steps related to the UCR-stage are described in Section 4.5.2 to show the steps in developing requirements for this stage.

As shown in Figure 32, the goal of this framework is therefore four-fold:

- Generalize domain analytical workflows to present high-level problem-solving direction
- Incorporate both general domain analytical process and individual analysis approaches
- Bridge the gap between high-level design concepts and fine-grain implementation of such concepts
- Augment organizational information analyses through modeling domain users reasoning approaches

On the conceptual level, this framework is designed to effectively engage the domain users during the design process and more importantly, to alleviate their lack of incentives in adopting novel analytical tools. Through revealing the high-level sense-making process in a general analytical workflow, this framework presents a cross-domain perspective that facilitates the design of a visual analytics system. In addition, it further provides a consistent and efficient starting point for designing a visual analytics system, and serves as a great initial approach to engage and communicate with domain users.

On the implementation level, this framework is constructed to inform the design of a visual analytics system through disseminating and incorporating the general analytical workflows into the design artifacts. It also presents visual analytics researchers and designers with more tractable design processes and more importantly, it provides them with a basis for assessing visual analytics use patterns and evaluating their impact.

To incorporate a comprehensive analysis environment, this framework focuses on covering both the objective generalization of domains' analytical workflows and the subjective experiences of individuals domain knowledge workers. The fundamental principle of designing a visual analytics system from which the four recommendations are derived is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an design artifact.

On the one hand, it emphasizes incorporating the observed general domain analysis methods and models into visual analytics systems. These models and methods are derived from the characterization of the targeted organizations, and they represent the domain requirements with a more abstract and generic development description. This framework requires the characterization of an organizational analytics processes (Recommendation 1).The design is based on the clear depiction of the analysis processes utilized in an organizational environment. These analysis processes are the core components in the designed visual analytics system. This framework further demands dissemination and transformation of the domain analyses into tangible design artifacts. These fine-grain artifacts are used to inform the creation of an innovative, purposeful visual analytics system for a specified problem domain (Recommendation 2).

On the other hand, this framework concentrates on the integrations of domain users' perspectives into the system development process. It considers and analyzes the possible interactions between users and visual analytics systems at an early stage, and provides the users with flexibility to customize and refine their analysis workspaces.

Because the design artifact identified in Recommendation 2 is purposeful, it must yield utility for the specified problem. Hence, to inform accurate visual designs, Recommendation 3 focuses on transforming these design artifacts into programmable visual elements. The designed system needs be innovative, solving a heretofore-unsolved problem or solving a known problem in a more effective manner. The incorporation of various "feedback" loops to enhance a designed system would provide opportunities to engage domain users and identify novel domain analytical processes (Recommendation 4). The process by which it is created, and often the artifact itself, incorporates or enables a search process whereby a problem space is constructed and a mechanism posed or enacted to find an effective solution.

Both these focuses guaranteed the proposed framework would comply with the above two required fundamentals in that: (1) it presents visual analytics design from the perspective of the encapsulation and facilitation of general domain analytical practices; (2) it supports the design processes for customizable analysis environment, and informs a systematical process for the development of an effective visual analytics system.

4.4.1 Design Methodology

To materialize such framework, a series of research processes were conducted in this dissertation: it began by categorizing the design experiences gained from collaborations with various organizations into a general organizational analysis workflow. Then, validated by domain users, this research incorporated the general workflow into a two-stage design, and listed the necessary design considerations for each stage. It further followed these considerations and developed visual analytics systems through iterative prototyping with domain users. Through extensive empirical evaluations of the two design stages, this research finally encapsulated both stages into a coherent general design framework.

The design process of this framework is intertwined with the actual implementation practices. This bootstrapping process emphasizes using both theoretical design aspects and practical experiences to construct a coherent and comprehensive design framework. On the one hand, the implementation practices present essential functional components that need to be incorporated into the framework. They further provide a testing ground to verify and validate the design framework. On the other hand, the theoretical framework instructs proper implementation steps, including the use of formative and summative evaluations. It details the natural progression for designing a visual analytic system and presents it in a cohesive manner. From the initial communication with targeted domain users and to the prototyping and iteration of visual analytics system, this framework illustrates the necessary actions and recommendations to design a visual system that augments organizational analytics processes.

In this section, this dissertation presents the design methodology that is related in the materialization of the two-stage framework.

4.4.1.1 Settings: Organizational Environments

The proposed framework mainly targeted facilitating the information analysis processes in organizational environments. Given the analysis-intensive nature of these organizations, a general design framework is believed to hold significant performance implications in instructing the development of visual analytics systems that can help employees solve increasingly complex and often ambiguous problems.

However, as presented in previous research [118] [36], an organizational analytical task is a process of handling multiple channels of information through the utilization of trained knowledge and current resources. Characterizing the analytical process in an organizational setting, such as a company or a governmental agency, is a complex process:

To guarantee the unbiased nature of the domain analysis and the generalizable outcomes, qualitative interviews and observations were conducted with multiple large organizations.

In the past three years, this dissertation performed interview studies within three groups of professionals in different organizational settings, including bridge-asset managers in The U.S. Department of Transportation, who propose and execute strategic bridge maintenance plans; business analysts from Xerox, who retrieve and analyze documents for information essential to the operation of the business; and network operational manager from Microsoft, who monitors the status of physical servers and network health.

All of the above organizations gave the opportunity for close, in-depth interactions with their knowledge workers and to conduct surveys and interviews, which were crucial in studying their analytical processes.

4.4.1.2 Data Collection and Analysis

The starting point of the data collection process is, typically, to investigate the analysis workflow for the individual domain. Following the business process notations used in these organizations [150, 162], this dissertation undertook a similar approach as Sukaviriya et al. [150] in identifying the basic analysis processes used in organizations. It focuses on examining information for the occurrence of a task, the inputs and outputs of that task and overall, the sequence of analytical tasks.

In particular, to understand the bridge maintenance domain, this dissertation conducted a nation-wide survey in the United States, and followed up with the participants through close examinations to understand their general analysis workflow and task requirements. To portray business analysis workflows, this dissertation interviewed 30 Xerox employees to gain insight about they analytical workflows. Finally, to better study the complex network analysis processes, this dissertation conducted several on-site interviews and discussions with cloud-service operators at Microsoft to observe the day-to-day operations performed by the back-end cloud service teams.

In doing so, this dissertation aims to identify the generalization of these organizational analytical workflows. Through analyzing the results from these collaborations, this dissertation gained domain analysis insights through close observation of these domain users' analytic workflows, and leaned their actions required for achieving each analytical task. While all these organizations' analysis processes are unique considering their problem definitions and solution trajectory, all of the three organizations share general project responsibilities that required them to be effective and efficient at generating pragmatic solutions to move the decision-making process forward. Therefore, this dissertation considers these organizational analysis processes to be quite representative in revealing the needed general analytical workflows for visual analytics designs. Details of how these collaborations lead to the characterization of the general domain workflows can be found in Section 3.3.

4.5 The Two Design Stages

The efficacy of a visual analytics system puts emphasis on the support for both organizational and individual's analysis requirements. As shown in Figure 33, the design framework is therefore proposed based on the same philosophy.

In particular, this framework follows the typical software development life-cycle model [106, 178], and incorporates the analysis requirements into two design stages (OD-stage and UCR-stage). Both of these design stages serves as the basic building blocks to construct the components and techniques for a visual analytics system. Individually, each of these stages is detailed to inform visual analytics designers with tangible development guidance. As shown in Figure 33, a total of four individual design steps are described and utilized the two stages to direct the process of a system development, namely "domain observation and analysis", "design artifact specification", "system deployment and user training", and "usage pattern analysis and customization".

Collectively, both stages contribute to the development of a visual analytics system. They support iterations among different design steps, and inform the transitions from

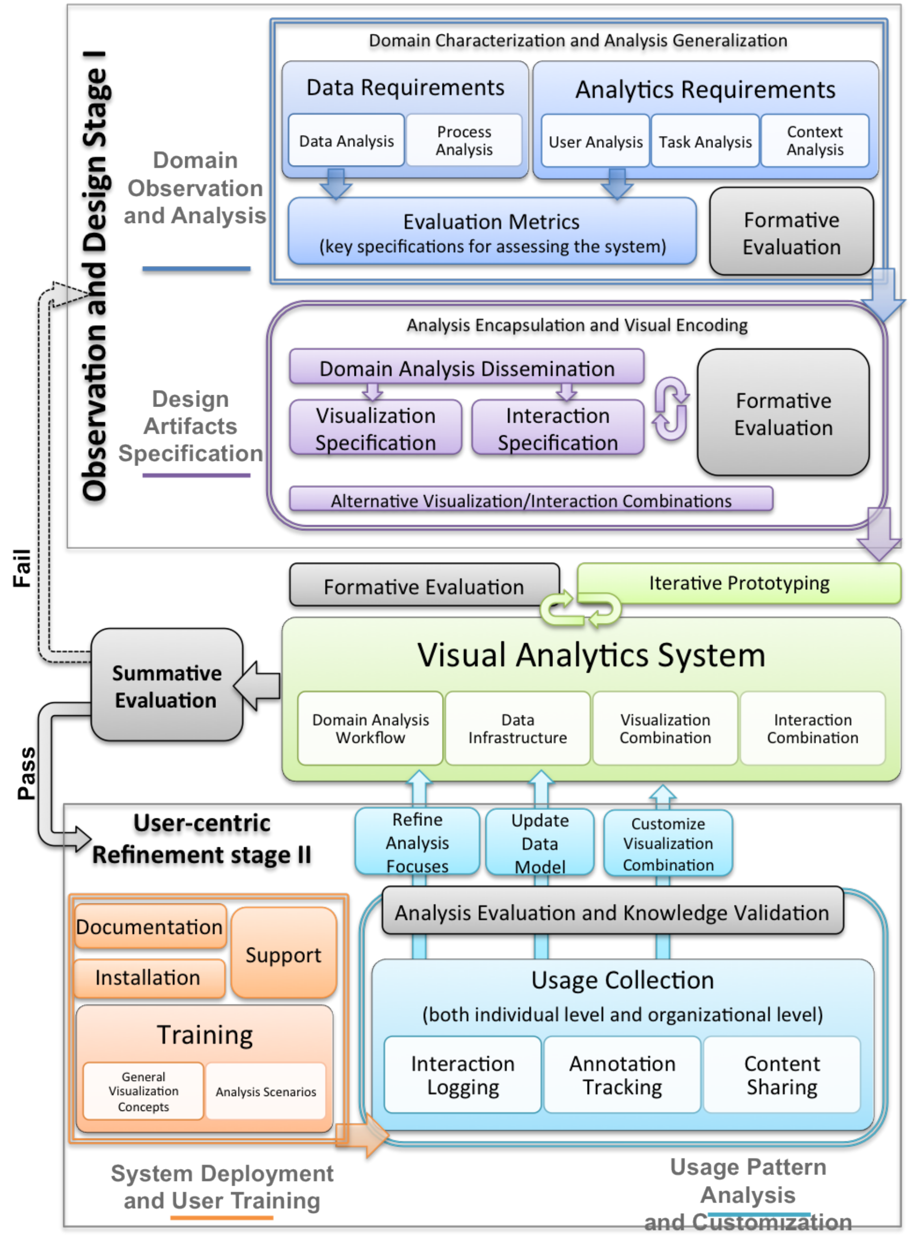


Figure 33: An overview for the two-stage framework. The Framework is listed as a system development lifecycle. The ***Observation and Designing*** stage is consisted of the top two design steps, while the ***User-centric Refinement*** stage is listed at the bottom. Dash line means potential development direction, but not necessarily required for every development.

one step to another. The combined efforts from both stage grant visual analytics designers a holistic design perspective, suggesting design considerations and processes from domain characterization to individual refinement. Together these two stages show the key ideas of the visual analytics development approach: domain analysis, iterative prototyping, swift feedback (such as soliciting user feedback), and human-centered analysis customization.

4.5.1 Stage I: *Observation and Designing* stage

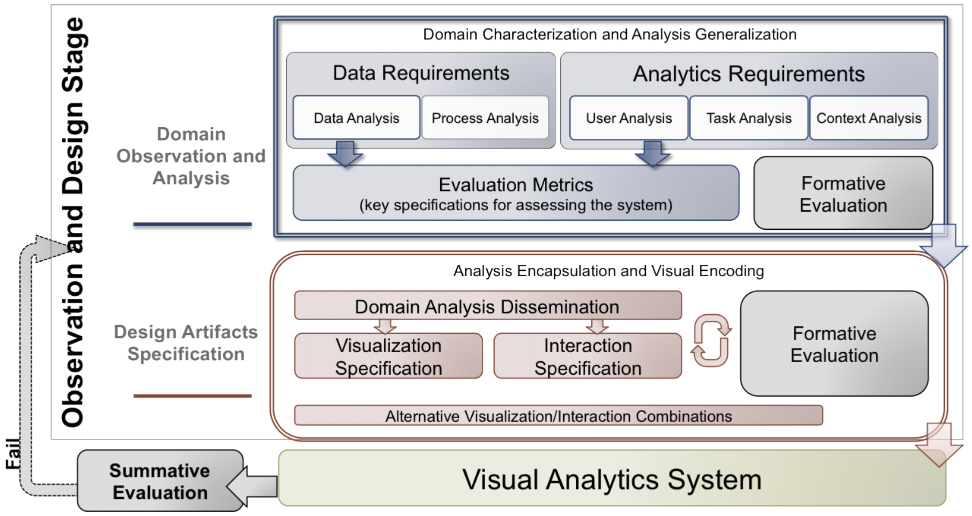


Figure 34: The ***Observation and Designing*** stage (Stage I) and its two design steps.

As shown in Figure 34, the primary goal of the ***Observation and Design*** stage is to identify the target domain's analytical processes and to determine the required system functionalities. It further specifies the design artifacts that are appropriate to implement such functionalities. Therefore, the use of two consecutive development steps were suggested in this dissertation, namely the domain observation and analysis step and the design artifacts specification step. Both steps are used in achieving the first stage of the iterative prototyping of a visual analytics systems.

4.5.1.1 The *Observation and Analysis* step

Following previous design research (e.g. HCI [44], Knowledge Management [119] and visualization community [115, 161]), the observation and analysis step emphasizes determining the system requirements, structuring requirements according to their interrelationships, and developing the visual designs and evaluation metrics. There are four essential implementation components in this step, including Identifying analytics requirements, specifying data requirements, constructing evaluation metrics, and finally performing formative evaluation to fine tune the domain characterizations.

As the core foundation in visual analytics rests on understanding domain analytical processes, this domain characterization and analysis generalization step is one of the most important design activities in the framework. It urges the visual analytics designers to communicate and interact with domain users to learn about the data and the analytical tasks within the targeted domain. This design step motivates the designers to categorize both the data and analytics requirements through interactions with end-users, and further assists the designers to transform these requirements into tangible system functionalities. Both the data requirements and analytics requirements are used to derive the evaluation metrics that are essential for assessing the efficacy of the designed visual analytics system. In the follow sections, this dissertation presents details of each implementation components.

The goal for *Observation and Analysis* step is two-fold:

- Characterize the general domain analytical processes
- Identify design artifacts for visual analytics implementation

4.5.1.1.1 Implementation Component: Identifying Analytics Requirements

As illustrated in Figure 35, The analytics requirements are generated based on context analysis, user analysis, and task analysis. Following ethnographic methodology [6, 135, 152], typical methods perform the these analyses include semi-structured interview

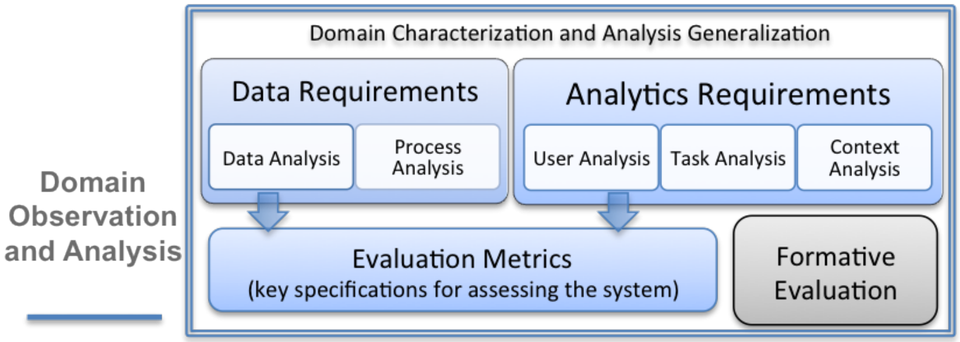


Figure 35: The *Observation and Analysis* step

(see Section 3.3.1.3 for examples), surveys (Section 3.3.1.1), and on-site observations (Section 3.3.1.2).

Context Analysis

As noted in organizational design and HCI research [178, 44, 18, 109], context analysis is a useful method to analyze the environment in which a organization operates. On a broader scope, context analysis provides a constraint to ensure that all factors that may affect the usability of a product are considered. It also helps to ensure that user-based evaluation produces valid results, by specifying how important factors are handled in an evaluation, and by defining how well the evaluation reflects real world use [14].

In developing a visual analytics system for organizational environments, the context analysis has been narrowed to understand the technical, analysis and collaboration settings where the visual analytics systems will be used. The main goal for a context analysis in the design process is to analyze the organizational environments in order to acquire an overall characteristic of the domain. Overall, context analysis can provide ideas for design factors such as metaphor creation/selection and patterns of communications between users and the system.

In addition, the context analysis provides insights to develop a strategic plan for the development of a visual analytics system. It examines whether and how the interaction among individual domain users and between the users and the organizations would impact or even change the design of a visual analytics system.

Typically, there are three aspects in context analysis: technical context, analytical context, and social context. While not all three are needed for every visual analytics design process, individual development project should select suitable aspects in accordance with the complexity of the targeted organization and the scope of the visual analytics system in design.

Technical context: When designing a visual analytics system for an organization, the end-user's technical skill level and the cost for adopting the visual analytics system are often two significant factors that shape the design and development strategy.

On the one hand, a properly designed visual analytics system should comply with existing technical environment and users' skill sets. This aims to minimize the potential cognitive overhead that is caused by a new analytical system. More importantly, by placing the users in a more familiar technical environment, such visual analytics system could also motivate the users to actually take advantage of the features in a visual analytics system for their analysis needs.

On the other hand, through understanding the current techniques in an organization, technical analysis provides a baseline for visual analytics designers to compare to. This analysis enables the designers to identify the insufficiency in existing organizational techniques. It further helps the designers to determine the needed analysis features and functions for a visual analytics system.

For example, the collaboration between this work and the USDOT is benefited by performing the technical context analysis. The initial design and implementation for IRSV system (detailed in Section 3.5) was targeted to be a multiple-coordinated visual analytics system that runs on each bridge manager's desktop. This design complied with the technical environment at that time and was proven useful with the wide deployment to bridge managers across the States.

Over the course of past two years, the overall technical environment in USDOT has gradually changed based on technical advancement introduced by the use of the original IRSV design. Since then, the bridge managers have become more familiar and confident in visual analytics systems. Thus, the collaboration has led to the reassessment of the bridge managers' technical environments, and further resulted in the enhancement of the existing visual analytics system to a web-based collaborative analysis environment (see Section 3.6.2.1).

Analytical context: While similar to the task analysis (detailed in the following section), analytical context analysis focuses on depicting the overall analysis environments rather than specific task elements. It aims on conceptualizing the relationship between general analytical activities and the organizational operations. In particular, this analysis focuses on mapping out the resources (personnel, data, technique) and restriction that need to be considered in the design of a visual analytics system.

A typical approach to perform the analytical context analysis involves investigations on the analysis process of an organization. Following literature in business process learning [20, 150, 109], these investigations should be constructed towards the depiction of the five Ws (Who, When, Where, What, and Why) that are used in an analysis operation. Summarizing on the domain users' feedback on these questions, this analysis further concludes an overall concept of an organization's analysis environment.

For instance, Table 16 listed five sample questions that has been asked during the collaboration with Microsoft (as shown in Section 3.3.1.2).

Through investigating these questions with domain users, the dissertation was able to depict the general analytical workflow for cloud-service managers, in

Table 16: Sample questions that are asked to analyze the analytical context in Microsoft

Who are the main operator for these tasks?
Where are the tasks carried out?
When will the information get updated?
What resources are utilized in the entire task operation?
What are the organizational restrictions that may limit or diverge the task workflow?

which they need to manually integrate individual tools to construct a cohesive cloud-service analysis. Particularly, the answers to the questions revealed the current practice of most of the cloud-service managers in that: they typically consider the cloud-service system in terms of connections and clusters; and they monitor their systems mainly through independent tools, using one custom tool to check one machine's details, switching to another for its connections, and query a database for its status.

Social context: Due to the dynamic and diverse nature of analysis activities, modern organizations produce large amount of data throughout their business operations. Most of the data is distributed across the entire organization domain, making it difficult for individual knowledge worker to find desired information. Therefore, the social context analysis is used to identify the information pertinent to individual's analytical needs and more importantly, to portray the dynamic analysis flow that are essential to the organization's operations [178, 20, 166].

In the context of visual analytics design, the social context analysis mainly characterizes the analysis needs, knowledge worker, and the interactions between these two. Especially in a large enterprise, where documents and information are generally decentralized among employees, performing an analytical task requires collaborative efforts to achieve efficient information retrieval and decision-making.

A proper implementation of the social context analysis provides visual analytics designers with insights on the dynamic nature of an organization's analytical flows. It helps addressing various design questions, such as: which streams of data to be fused together, where to retrieve information for certain analytical tasks, how to share individual's analysis with others and finally, in what way could one collaborate with others to reach business decisions.

As shown in Section 3.6.2.2, the collaboration with Xerox corporation has utilized the social context analysis to identify the characteristic of the analysis processes that are used in managing document activities. In general, Xerox employees must utilize and analyze information from multiple channels, and are required generating shared products effectively (e.g., a maintenance proposal or analytical report). Subsequently, they need to coordinate with multiple colleagues from different locations to agree on strategic decisions.

Based on the results from the social context analysis, this dissertation was able to capture the employees' strong needs for information sharing and collaboration. The resulting visual analytics system, Taste (discussed in 3.5.2.1), is therefore designed to accommodate these needs. By utilizing an interactive Story Telling function, Taste allows employees to collaboratively search for information, annotate the desired information, and share it with others.

User Analysis

The user analysis is used to ensure the information and characterizations about the target users of a proposed visual analytics design are accurate and explicit. This analysis provides the designer with perspectives on the different categories of domain users who will ultimately use the visual analytics systems. It requires the designers to identify characteristics of the user population that are likely to influence their acceptance and effective use of a visual analytics system.

Conceptually, the user analysis may seem obvious; in action however, it is not trivial. As suggested by Dillon et al. [48] that, the user analyses are typically highly context sensitive and vary from one generalization to another, never mind agreement across proponents. It demands the visual analytics designers to actively engage domain users and elicit the design requirements through extensive interactions with these users. As emphasized by previous research [115, 6, 135, 152] that, in the process of user analysis, it is rather important for visual analytics designers to *not* make their own assumptions, but to ground the user characterization strictly on the information collected from domain users.

The user analysis often means distinguishing users broadly in terms of expertise with technology, task experience, educational background, usage constraints and personal traits, gender and age [56, 116]. As summarized in Table 17, the user analysis provides designers with information on:

Table 17: The information provided by user-analysis.

Demographics data	Occupation, Organizational position, Specific task focuses, Computer skill, and Experience with similar analytical systems
Task related factors	Job characteristics, Work styles, Frequency of analytical tools used for the tasks, and Usage constraints and preferences
Personal Traits	Cognitive styles, Affective traits, and Skill sets or capabilities

Task Analysis

Once user and context analysis has passed, the task analysis is conducted to specify the tasks and workflows in an organizational environment. Particularly, task analysis [63] has been used for documenting user tasks and processes.

Much previous research has suggested and conducted task analysis to investigate existing situations [130, 8, 47]. Task analysis includes scenarios and conditions under which users perform the tasks. It focuses on analyzing and articulating the nature of analytics tasks that domain users or organizations normally performs. Particularly, this analysis emphasizes on the understanding of what people do in order to achieve their analysis goals and how their task activities are performed during the analysis processes.

Specific to the design of visual analytics systems, task analysis is performed in a more narrowed scope. This process emphasizes understanding organization analysis flows and knowledge workers' roles in this flow. It is used to help visual analytics designers to more effectively define the task structure, the needed fine-grain analytical actions, and the strategy of the organization. The task analysis provides a detailed understanding about the domain analysis processes and their related requirements. Instead of a broad focus, the scope of task analysis in visual analytics is specified to the identification of actionable knowledge that is utilized in the domain analytical workflows.

On the one hand, task analysis helps the visual analytics designers to gain basic understandings about the targeted domain. It motivates the designers to learn the domain vocabulary in describing the problems and challenges and more importantly encourages the designers to share the same analysis perspective with the domain users. In turn, the task analysis enables the designers to effectively communicate with domain users, and leads to the elicitation of a set of proper analytics design requirements. It depicts the information needs, patterns of information, and routines that are specific to users or organizations during their analytics processes.

On the other hand, the task analysis enables the visual analytics designers to discovers patterns of exceptions. The objective of task analysis is to determine the necessity and sufficiency for visual analytics systems to support user and organizational task activities. Task analysis provides the designers the access to have close collaboration with domain users, and enables these designers to determine the likelihood of target users accepting of a system's functionalities. It reveals the errors in design requirement specifications, and helps to avoid the unnecessary cost in the development of a visual analytics system.

Based on significance of task analysis in instructing the design a visual analytics system, this dissertation follows the research of task analysis in general domain [84] and regards task analysis as one of the most important analyses in the *observation and design* stage. This dissertation recommends the visual analytics designers to conduct task analysis at the earliest stage to improve the efficiency of the overall design process.

When developing organizational visual analytics systems, it is useful to analyze tasks at two levels: individual level and organizational level. The overall task analysis should start with identifying the tasks or goals on the individual level. In this process, the visual analytics designers should focus on learning about the nature of domain users' analytical workflows. The designers need to depict tasks activities that are meaningful to individual's job or work within the analytical context, and identify the workflows that is horizontal in each individual analytical workflows. Details about this process and its results can be see in Section 3.3

The task analysis, on the organizational level, typically involves generalizing the high-level common task activities of the analytical process. Such generalization focuses on cross-process analytical tasks that are commonly applicable to multiple domains. It examines the task workflow and the distribution of work and work skills within an organization. Development of a visual analytics system must take into account of the movement from one type of structured work environment to another. As summarized in Section 3.3.2, through organizational task analysis, the dissertation has identified a set of six task activities common to organizational analysis processes, namely content gathering and aggregation, content filtering and customization, content organization and information analysis, evidence collection and hypothesis generation, report generation and status update, and post-analysis and summarization.

4.5.1.1.2 Implementation Component: Evaluation Metrics

The results from both analytics and data requirement analysis are important inputs to establish the visual analytics evaluation metrics. The evaluation metric specifies the expected analysis goals from the domain users for the designed visual analytics systems. Such metrics often presents key features that a visual analytics system, when designed and implemented, should incorporate. These metrics are used to measure performance, either of the system alone or of the combination of the user and the system for interactive technologies.

The visual analytics systems focuses on the analysis capabilities in provided by visualization, interaction, and collaborations. Thus, to determine the effectiveness of a visual analytics system, the visual analysis designers need to develop proper evaluation metrics. In developing a visual analytics system, the evaluation metrics typically provide benchmarks for both the formative evaluation [52, 141] and the summative evaluation [141]. Such metrics guide the rest design steps of the visual analytics development process, and aims to address the concerns listed in Table 18.

It is worth mentioning that a number of visual analytics researchers have looked at ways to evaluate a visual analytics system using heuristic evaluation [180, 137, 153, 155, 13]. While such a method has demonstrated its utility in evaluating the usability of an interface, the heuristic evaluation is limited in evaluating the analytics process embodied in visual analytics systems [154, 138].

Following the evaluation methods commonly used in intelligence community [71], this dissertation proposes the use of qualitative evaluation metrics [178] for evaluating the analytical utility of visual analytics systems. Qualitative evaluation metrics provides a top-down approach for instructing goal-directed analysis software evaluations. Such metric provides visual analytics designers with reusability for evaluating similar system, and further enables them to perform cross-domain comparison for the visual analytics systems.

As shown in previous work [178], such evaluation metrics, when properly crafted, would cover a number of issues including the visualizations, how the visualizations facilitate analysis, user interactions with the visualizations, and the support the environment provides for the analytic process. As shown in Table 18, Scholtz [137] summarized five areas of visual analytics concerns. These concerns are particularly relevant to the development of the qualitative evaluation metrics for visual analytics. Table 18 also lists some of the ways to measure these concerns that are proposed by this dissertation.

The actual developed qualitative evaluation metrics may vary from organizations to organizations, depending on the organization settings and their diversified tasks. Consequently, the qualitative evaluation metrics need to reflect the needs for visual analytics designers in terms of assess the efficacy of the designed system; it also have to comply with the analysis needs of the targeted organization.

The specific measures or quantitative aspects of the metrics are typically determined based on the above analysis results (context, user, and task analyses), formative evaluation tests on mockups or prototypes, as well as the goals and constraints of the visual analytics system being developed. Note that not all concerns are required at the same time. Visual analytics designers are encouraged to select subsets of the metric in their design work to determine the most appropriate evaluation approaches. Examples of the utilization of these evaluation metrics can be seen in Chapter 5.

4.5.1.1.3 Summary: the *Observation and Analysis* step

The goal in the *Observation and Analysis* step is to characterize the targeted organizational environments. It emphasized on the identification of the both the analytics requirements and data requirement. This step further outputs the domain analytical workflow that will serve as inputs to derive the design artifacts. Detailed process of performing this analysis stage is described in previous chapter (see Recommendation

Table 18: Areas of visual analytics evaluation concerns

Concerns	**Description**	**Sample Measure Items**
Situation Awareness	Visual analytics system supports the analysts' knowledge on performing domain-specific analytical tasks.	Ability to track the change of information Provide contextual analysis environments; Selfdescriptiveness of the action;
Collaboration	Visual analytics system enables communication and information sharing between collaborators.	Ability to share evidence; Support intuitive communication between collaborators; Capable to reveal information flows;
Interaction	Visual analytic system provide sufficient visualization and interaction combinations to facilitate domain analytical processes	Suitability for the task; Controllability; Customization
Creativity	Visual Analytics system supports the flexibility and diversified analysis processes for individual domain analyst.	Support individual's tasks; Effective in searching for analytical results Ability to lead to high quality of analysis solutions;
Utility (Analytical Process)	Visual analysis system fits in analysts cognitive strengths and reduces its cognitive workload on analyst	Easy to use; Engaging; Comply with existing technical context; Conformity with user expectations or consistency;

1 & 2 in Section 3.3).

4.5.1.2 The *Design Artifacts Specification* step

Once the *Observation and Analysis* step is performed, the *Design Artifacts Specification* step need to be executed to inform the design specifications that are used for the implementation of a visual analytics system. The goal in this step is to support the identified analysis requirements concluded by context, task and user analyses; it further helps designing the functionalities for a visual analytics system to meets the evaluation metrics requirements.

As shown in Figure 36, this step focuses on disseminating the domain analysis processes, and transforming them into tangible visualization or interaction specifications. It emphasizes the depiction of the programmable instructions from the previously identified data and analytics requirements. In doing so, key interface elements of a visual analytics system—such as the selection of visualization, the choices of interaction, and the combination of these two—are derived based on the high-level analysis results outputted from the *Observation and Analysis* step.

The concluded general actionable knowledge and its related design considerations are listed in Figure 10. This list presents a clear connection between the aforementioned key common task activities and the needed visual analytics design considerations. The previously characterized six common task activities (see Section 4.5.1) are disseminated into fine-grained actionable knowledge. The corresponding design considerations are consolidated by transforming this actionable knowledge into practical functions.

The goal for this design step is to:

- Perform analysis encapsulation and visual encoding
- Disseminate high-level task activities into actionable knowledge
- Transform actionable knowledge into visual encoding

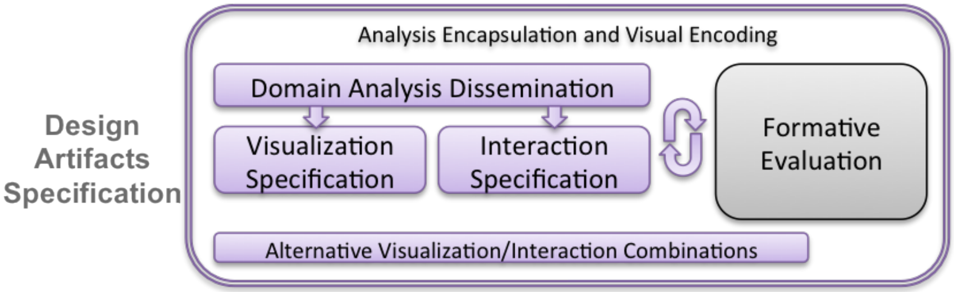


Figure 36: The *Design Artifacts Specifications* step

4.5.1.2.1 Implementation Component: Domain Analysis Dissemination

The general domain characterization concluded in the *Observation and Analysis* step presents a overall task structures in an organization. While these activities are useful in describing a general analytic process, they are often too general to provide any specific considerations in actual system designs. In order to designing a visual analytics system requires support for the analytical workflows of the knowledge workers, it requires the identification of tangible design artifacts that can connect the design of a system with the general domain analytical workflows.

The first component in this step, therefore, focuses on disseminating high-level analytical workflows to tangible design artifacts. This step is established based on the determinants of the tangible artifacts in terms of characteristics of the knowledge worker's analysis tasks, information source, and their relationship between both parties. It is typically conceptualized into a two-stage action to identify the nature and criteria for the tangible design artifacts, and to describe the general analytical workflow using these artifacts.

As extensively described in Recommendation 2 (see Chapter 3.4), the key to disseminate these high-level task activities is the identification and use of actionable knowledge. Actionable knowledge shows the pragmatic view of knowledge utilization and application towards specific analytical ends [25]. The actionable knowledge is the knowledge used to instruct domain user's actions when addressing a task. The actionable knowledge details the relations between domain analytical tasks and the related knowledge actions. It presents the analytics activities from domain users perspectives, for example, what data to look at and which person to communicate to in achieving a certain task. Such examples can be seen as targeting a direct marketing campaign in a Bank's operations, or planning infrastructure maintenance aimed at repairing those assets with lowest health in asset managements.

A situational domain task characterization is recommended by many researchers [10, 170] in the identification of actionable knowledge. A typical approach is to perform interviews and real-time discussions with targeted organizations. As suggested by Shani et al. [142], there are five processes that are needed to guarantee a successful creation of domain actionable knowledge: the emergent task process, the inquiry process, the integration process, the experimental process and the diffusion process. These processes focuses on utilizing an emergent collaborative inquiry process in which behavioral and social science knowledge is integrated with existing organizational knowledge for the purpose of generating simultaneously scientific and actionable knowledge [38, 170]. All these characterization processes guaranteed the nature of actionable knowledge would fit well with the above two requirements in that: (1) it represents the fine-grained elements of each analytical task, and thus is quite instructive for the design of a visual analytics system; and (2) it is extracted from domain users' knowledge actions, and therefore can be consumed without additional cognitive overhead.

During these processes, the organizational members are fully involved in the inquiry process and share the responsibility for the effort. The visual analytics designers are suggested to establish close working relationships with domain users, and observe and discuss their detailed actionable knowledge through a dialectical process [40].

For example, during the design process of the GTDVis [168], all the terrorist analysts participated in the interviews were asked to envision the hypothetical process of carrying out their usual tasks with their regular tools and working environments. As shown in Figure 40, they were encouraged to also think about additional functions that might be useful but not yet available in any of the tools they typically used. Based on their analysis interests, the designed system targeted on depicting one of the most fundamental actionable knowledge in investigative analysis, the five Ws (who, what, where, when, and why); it emphasizes revealing the transition of a terrorist groups, the temporal trends of that group, and its similarities with other groups in the dataset. The interviews were semi-structured with the ambition to encourage the respondents to give a narration more than just answer questions. Specifically, these terrorist analysts were asked about their fine-grained analysis actions that are used in their daily practices: what the essential analysis methods they have, and how they utilized these tools to execute each action. In doing so, the GTDVis was designed to support the key actionable knowledge that could improve the productivity of the domain users analysis processes.

4.5.1.2.2 Implementation Component: Transforming the Actionable Knowledge into Visualization and Interaction Specifications

To properly encapsulate the domain user's actionable knowledge into the design of a visual analytics system, the other important component in the *Design Artifacts Specification* step emphasizes transforming the actionable knowledge into visualization and interaction specifications. This transformation process needs to be concluded consecutively with the identification of the above actionable knowledge.

As shown in Figure 2, this research provides a list of design considerations to effectively instantiate the design of a visual analytics system. Following commonly established design theories [143, 109, 178], these considerations are identified through several iterations of prototyping with the targeted domain end-users. Detailed examples for these deductions are described in Chapter 3.5.

These considerations are summarized based on the implementation experiences with the aforementioned three organizations. As exemplified in three successful visual analytics systems, these design considerations are demonstrated to have sufficient information to instruct successfully implementations for visual analytics systems. They represent the details on how to transforming each general analytical task into system's functionalities. These general design considerations emphasize encapsulating both the general analytical workflow and individual's analysis process into users' accepting visual analytics functions. They are further evaluated by domain users through extensive empirical evaluations, as reported in chapter 5.

Both formative evaluations and summative evaluation presents an invaluable role in transforming the actionable knowledge into these design considerations. They essentially helped to encapsulate the domain users' actionable knowledge into functions, and led to the acquisition of critical functionalities required to build a visual analytics system. Similar to the design of an information system [160], before transforming all gathered actionable knowledge from the *Domain Characterization and Analysis* step into design considerations, the organization must select the final alternative design strategy for the proposed visual analytics system because (1) different users offer competing concepts on what the system should do, and (2) multiple alternatives are available for an implementation environment for enhancing the proposed system.

Therefore, the formative and summative evaluation is needed for fine-tuning the design considerations, and verifying the specifications before the actual system implementation. They are further instructive to the iterative prototyping process that is used for the system implementation. The goal and procedures of both formative and summative evaluations are described in the following sections.

4.5.1.2.3 Implementation Component: Formative Evaluation

Formative evaluations identify defects in designs thus inform design iterations and refinements. A variety of different formative evaluations can occur several times during the design stage to form final decision decisions. In fact, formative evaluations occur during the entire visual analytics development life cycle. It is involved as an iterative process throughout the design life cycle. Adopting rigorous formative evaluations process is also beneficial to designers as they can verify requirement of a system at an early stage; and it further guides the designers to gradually move on to the UCR-stage (see Section 4.5.2).

Formative Evaluation is a bit more complex than summative evaluation. It is done with a small group of people to "test run" various aspects of instructional materials. It's like having someone look over the participants' shoulders during the development step to help them catch things that they miss, but a fresh set of eye might not. At times, the participant might need to have this help from a target audience. Although all three organizations shared similar common analytical tasks, the implementations of the prototyping methods for them were quite different (considering their diverse workspaces and time constrains). Specifically, the evolutionary prototyping [122] method was adopted for the collaboration with Xerox. Given the requirement of a deployable product to the enterprise, the evolutionary prototyping guaranteed more design iterations and, more importantly, allowed the build of a robust system in a structured manner.

In the proposed framework, formative evaluations are recommended in every design step to ensure the final implementations are sufficient to support the domain's analytical needs. The proper conduct of formative evaluation can also minimize the cost of development and increase the likelihood of users' acceptance of a visual analytics system.

4.5.1.2.4 Implementation Component: Summative Evaluation

In many visual analytics design cases, however, the use of formative evaluations all the way through the design cycle may not be feasible. The use of summative evaluations is then recommended in this framework to evaluate the design before the deployment of the designed system. At the end of the design and implementation process, summative evaluation need to be performed to provide information of the efficacy of the designed visual analytics system.

Using the identified evaluation metrics, this evaluation focuses on verifying and validating the product's ability to do what it was designed to do. It emphasizes the comparison between the designed functionalities and the desired analytical workflows. The summative evaluation is used to inform the changes needed in new releases. Compared with the formative evaluation, this evaluation focuses on the signs, analysis structures, and visualizations presented to users at the interface level, signaling the immediate interpretations assigned to them and the role they play in facilitating the domain analysis processes. In summative evaluation, the visual analytics designers should systematically break down the system functions into components (e.g. measurable tasks, resources, goals and constraints) that can then be analyzed and compared with the domain task activities.

The summative evaluation is an important indicator to determine the success of the designed visualization, and helps to decide if the design should proceed to the UCR-stage. As show in the overall framework (Figure 33), failing in passing the summative evaluation may results in the need to re-assessing the domain characterization process, and thus may require a new implementation of the designed system.

4.5.1.3 Summary: the *Observation and Design* stage

The first stage of the design framework considers the design of an organizational visual analytics system as a holistic design decision. This stage focused on characterizing domain's analytical processes through generalizing the domain analysis processes. Through interactions with domain expert representatives, this stage aims to identify specifications that should be considered during the design process. In particular, this stage emphasizes the need to identify the actionable knowledge that is associated with the domain's general analytical workflow. It suggests methods to transform the characterized task activities into design artifacts, and it further illustrates the design considerations that are useful in implementing a visual analytics system.

4.5.2 Stage II: *User-centric Refinement* stage

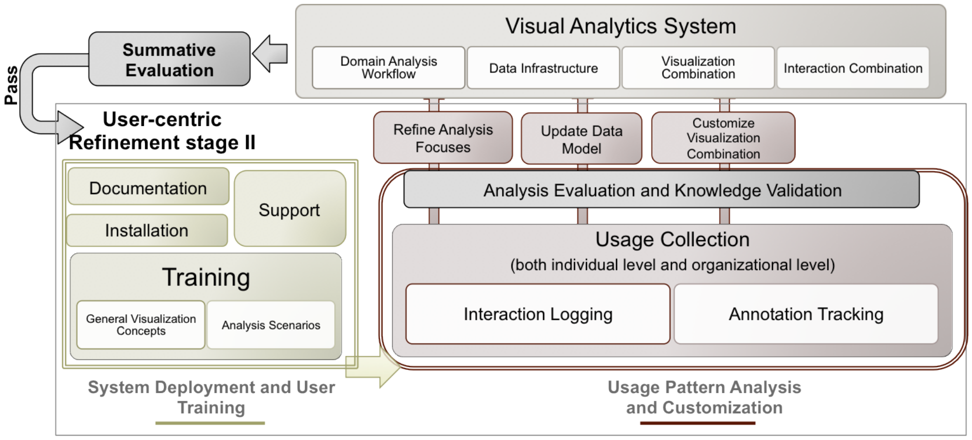


Figure 37: The ***User-centric Refinement*** stage (Stage II) and its two design steps.

Visual analytics systems typically are implemented within an organization setting for improving the efficiency of that organization. The realm of visual analytics research is therefore at the confluence of people, organizations, and technology, much alike the information system design [45].

When considering design of such systems for an organization, as mentioned in previous task analysis, the visual analytics design framework needs to cover both the organizational level (high-level generic task workflow) and the individual level (specified task scope and operations). While both levels are tightly connected to augment each other, their divergent scopes on how analytical tasks should be conducted have resulted in the need of a diversified approach for a visual analytics system.

Specifically, task performance on an organizational level means a generic, cohesive and holistic view of the general analysis workflow; on the contrary, on the individual level, different knowledge workers may share individual perspectives on task activities [109, 4]. As the essential operators of all the analysis activities, knowledge workers have different organization roles, capabilities, interests, and analysis characteristics. Their perspectives on organizational analysis processes therefore are different, resulting in diversified analytical needs and personal task routines. At the heart of these diversified analysis routines are the different combinations and sequences of the generic analytical processes.

Although the general workflow, identified in the ***Observation and Design*** stage, is valid in presenting the synthesis of the majority of domain analysis activities, individual differences weren't captured in that stage to customize the visual analytics system to support individual ways to carry out analysis. For instance, the interview with Xerox employee suggested that the organizational knowledge workers often perform information analysis and analytical reasoning in their own way. This makes externalizing and accommodating individuality almost impossible in a holistic design approach.

Consequently, this design framework recognized the need of a "feedback" process to integrate the individual's analytical practices with visual analytics systems, and achieve the customization of such system based on different analysis perspectives. Particularly, a ***User-centric Refinement*** stage is introduced.

As shown in Figure 37, this stage presents two design steps. After passing a summative evaluation (introduced in Section 4.5.1.2.4), the visual analytics system is deployed to domain users through the "System Deployment and User Training" step. Then, the "Usage Pattern Analysis and Customization" step emphasized on using the actual usage data collected from domain users to refine the visual analytics system, refining the users' analysis focuses, updating the data model (data focus), and helping users to customize the visualization combinations.

The goal of this stage is twofold. First, it primarily emphasizes identifying design elements that could inform the visual analytics designers with methods and techniques to incorporate end-users' diversified analytical processes. In general, the methods and techniques (as shown in Section 4.5.2.2) utilizes the rich information embedded in users' interactions, and captures and reapplies the individual analytical practices.

Secondly, this stage presents considerations when deploying a visual analytics system to domain users. It focuses on explaining how to approach and deliver the system to motivate domain users to adopt the novel analytical systems. While the research on this direction is still at its early stage, however, the results from empirical evaluation with domain analysts have demonstrated its efficacy in supporting customized analytical processes.

4.5.2.1 The *System Deployment and User Training* step

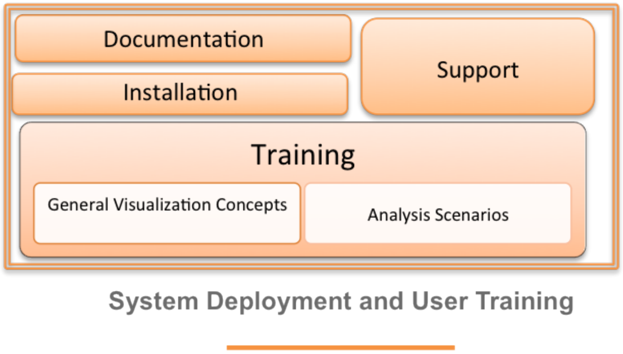


Figure 38: The system deployment and user training step.

An important fundamental of this design framework is the ***human-centered design***. This fundamental request placing the domain users and their *analysis process* at the center of a visual analytics system. Thus, comparing to typical software development life cycle, where the deployment is the final step to the domain user, this framework emphasize that the *System Deployment and User Training* is actually an essential part of the design process. As human-centered design research, the visual analytics field needs to emphasize on the importance of this deployment and the training step in placing analysts closer to the center of their analysis process.

The benefits of carrying out a rigorous system deployment design is twofold: first, it not only guarantees proper means for visual analytics designers to deliver their designed systems to vast domain users. In turn, it provides the designers a ground to evaluate the success and drawbacks of their systems based on inputs from various domain users.

Secondly, this deployment process grants the designers opportunity to access and collect the diversified domain analytical methods from individual knowledge workers, which encourages the visual analytics designers to come up with solutions to incorporate individuality into the refining process of visual analytics systems.

During the *System Deployment and User Training* step, there are multiple factors that need to be considered, namely documentation, installation, support and training with analysis scenarios. As show in Figure 38, the first three of these steps follow the typical software engineering and development life cycle [178].

Since it's efficient for visual analytics system follows the general software engineering approaches, therefore, these steps—documentation, installation, supports—are not the focus of this dissertation. On the other hand, considering visual analytics system' uniqueness in supporting domain analysis process, this dissertation emphasizes on the discussion of the design of training visual analytics system with domain users.

4.5.2.1.1 Implementation Component: Training Sessions

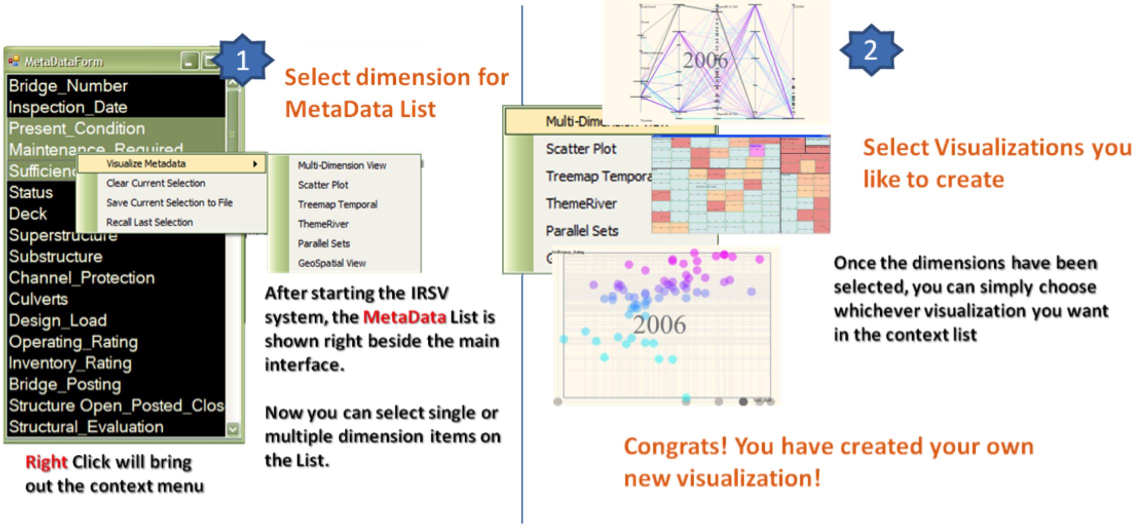
Training domain users to adopt and get familiar with a visual analytics system is not trivial. As pointed out in their extensive organizational studies, Markus et al. [108] noted that, domain users typically lack incentives to use analysis technologies, and they may share negative perceptions of the functions that the system was designed to support. Therefore, it is up to the visual analytics researchers and designers to utilize both the design interactions (see Recommendation 1 3.3) and training sessions to motivate the domain users to accept the designed visual analytics systems.

This effort starts with the formative evaluations. These evaluations provide feedback on the training as well as on the actual visual analytics system. In some instances, the visual analytics designer trains the domain analysts participating through a formal on-site technology insertion. In other cases, the designers would provide the domain users with a self-contained tutorial, listing all the core functions of a visual analytics system. In all cases, the expert representatives who have some exposure to the capabilities of the visual analytics systems should be invited to discuss effective ways to utilize the system.

General Visualization Concepts Session

While during the holistic ***Observation and Design*** step, expert representatives are familiar with visualizations and interactions that are utilized in the designed visual analytics system, other targeted domain users are generally not aware of the concepts of visualization nor visual analytics.

It is therefore necessary for visual analytics designers to introduce the details of each design decisions. They need to describe how the visualization captures the design artifacts, which are identified during analysis dissemination processes 4.5.1.2.1. In addition, the designers need to further explain the details of each visualization concepts, and the usage of these visualizations in domain's analysis activities.



Select dimension for MetaData List

Right Click will bring out the context menu

After starting the IRSV system, the MetaData List is shown right beside the main interface.

Now you can select single or multiple dimension items on the List.

Select Visualizations you like to create

Once the dimensions have been selected, you can simply choose whichever visualization you want in the context list

Congrats! You have created your own new visualization!

Figure 39: A sample from IRSV manual that was delivered to domain analysis.

As shown in Figure 39, a detailed manual was developed for deploying IRSV system (Section 3.5.2.3) to general bridge managers. In this document, details were listed about each visualization, including its design purpose, its related interaction, and its relevance to the users' analysis processes. The effort in providing such detailed document aims to ensure the domain analysts' familiarity with the visual analytics concepts, making them feel comfortable using the systems.

Analysis Scenarios

In many cases, the effort from merely introducing general visualization concepts is not sufficient for domain users to gain enough understandings about how to use a visual analytics system. Domain users mainly reason about the cases they have and the constrains that occurred in their analysis process. Differs from the designer's perspective on introducing system functions, the domain users emphasize how these functions would be useful in action, and are more willing to use a system if they see it fits in their existing problem-solving needs.

Therefore, concluded on the collaborations efforts, this dissertation emphasizes the necessity of using case scenarios in engaging and motivating domain users. It considers the introduction and showcase of analysis scenarios using the design system would largely complement the efforts of explaining the system on the function level.

A case scenario describes a general process in addressing the domain's problems. Such a case scenario should include the description of the entire analysis case, including the statement of the problem, the challenges to the existing system, and the solutions provided by visual analytics. These scenarios are generally identified through the collaborative efforts from both visual analytics designers and the expert representatives who participated in the holistic design.

For example, the following is a short paragraph in demonstrating the use case of GTDVis [168], developed by this dissertation in collaboration with U.S. Department of Homeland Security:

> "By examining the system overview (see Figure 40 left), we can see that a great deal of terrorist attacks took place in the Philippines. Zooming into that specific region and selecting the entire country in the map view (*what*) lists all the terrorist groups active between 1970 and 1997. A

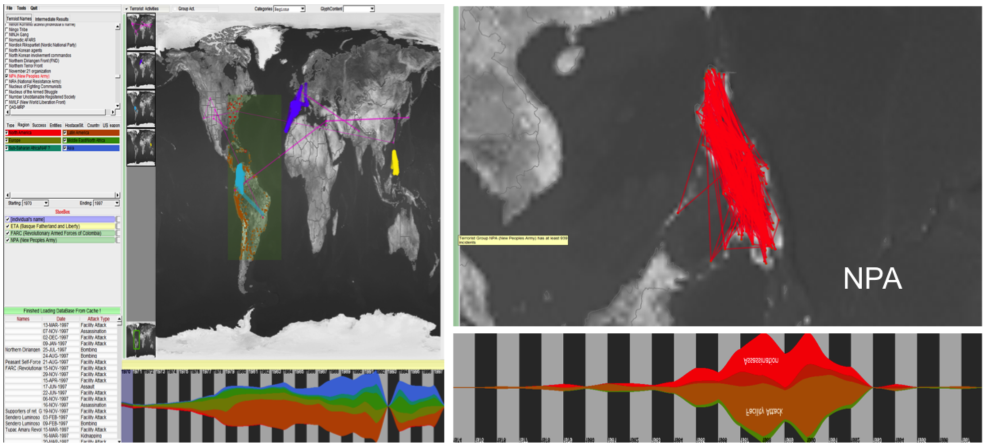


Figure 40: A sample case scenario for GTD

> quick search in the entity view (*who*) shows that the NPA (New People's Army) is one of the most active groups in the region (Figure 40 top right). Highlighting NPA reveals that although active, NPA is strictly domestic and has never performed activities outside of the Philippines."

As shown in a later chapter (see Chapter 5), this dissertation further presented three detailed case scenarios that are identified together with domain experts. These case studies, together with the training documents, have contributed to the successful deployment of the visual analytic systems to domain users. Therefore, this framework recommends the visual analytics designers to consider using both tutorial documents and case scenarios to motivate domain users to adopt the designed systems.

4.5.2.1.2 Summary: the *System Deployment and User Training* step

The *System Deployment and User Training* step focuses on deploying the designed visual analytics system to domain users. It aims to address the users' lack of incentives to use a new system, and focuses on motivating them to adopt advanced analytical tools and practices.

Through interactive training sessions with domain users, basic visualization and interactions should be presented to suggest the usefulness of the system in action. In addition, the identified analysis scenarios that describe a general process in addressing a domain problem should also be detailed and introduced to engage domain users. Since the user's acceptance of a system is a key factor in the success of incorporating the individual's analysis processes, this *System Deployment and User Training* phase is therefore of great importance in the overall design framework.

4.5.2.2 *Usage Pattern Analysis and Customization* step

The key fundamental in visual analytics is the support of the ***human-centered*** organizational analytical processes. Many organizational analyses centered around the individual knowledge workers' analysis capabilities in linking information, identifying and associate patterns, sharing findings, and finally communicating and collaborating with other colleagues. While the analysis outcomes may be similar among different domain users, the ways for these individuals to accomplish such analysis process may vary from one another.

For instance, based on the collaboration with the USDOT [165], this dissertation found that, bridge managers often need to develop their own analysis routines to maximize the use of their limited maintenance resources. Depending on what's available, a bridge managers strategy can be very different from his/her peers, and could require a different combination of the above analysis processes. In addition, sometimes even the same manager needs to take alternative analytical approaches due to changes in priorities.

Therefore, while a holistic way of incorporating domain knowledge is generally effective, many organizations worked with this dissertation also recognized the benefit of having a visual analytics system that can support the diversified analytical needs and personal task routines in an organization.

From these organizations' perspectives, the customization supports can be threefold: first, at the heart of the needed support is the ability to shuffle the analytical components in the system, rearranging the sequences and combinations of the generic analytical processes. Second, the ability for individuals to collect analytical findings and trace the analysis trials that led to these findings. Finally, these organizations need to have efficient means to share these analysis findings between knowledge workers, and provide an effect environment to support collective decision-making.

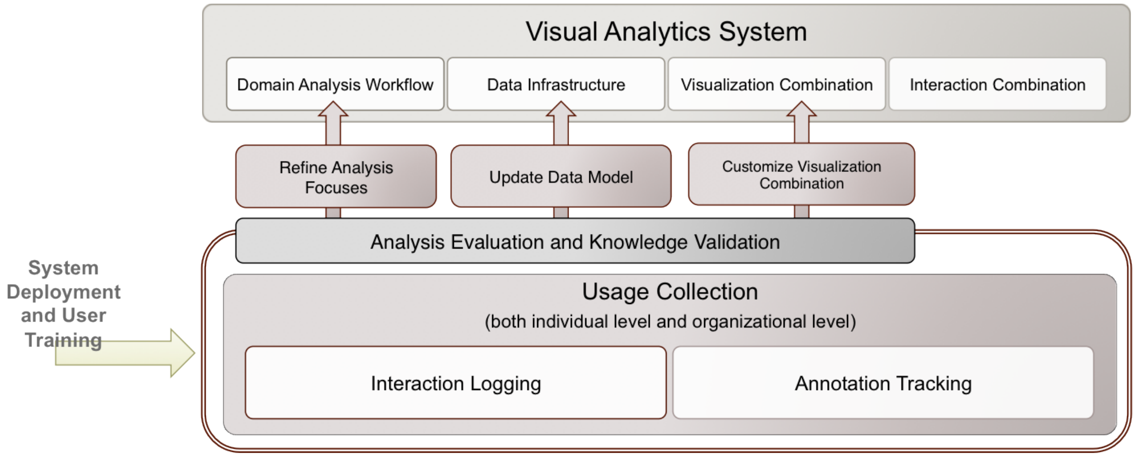


Figure 41: The *Usage Pattern Analysis and Customization* step.

Consequently, two methods for tracking and analyzing individual's task activities were proposed in this dissertation, namely interaction logging, and annotation tracking. As shown in Figure 41, all these usage collection methods focus on recreating and extending specific visual analytics results or routines through collecting and analyzing analysis behaviors on both individual and organizational levels. The goal for these methods is to fine-tune the visual analytics system and make it capable to adapt to individual knowledge worker's analysis behaviors. Such fine-tuning is reflected by allowing users to refine their analysis focuses, to update the visual analytics internal data models, and to customize the visualizations combinations.

In the following section, this dissertation describes each of the usage collection methods, and presents them as a whole in enriching and refining domain analysis processes through capturing and analyzing knowledge workers' task behaviors.

4.5.2.2.1 Implementation Component: Usage Pattern Analysis

As noted by Kindlmann [88] and Silva et al. [146], the lack of reproducibility of visualization research has the potential of hindering the advancement of visualization as a science. They argue that in order to recreate and extend specific visualization results, knowing the complete process of how the results are generated is just as important as the techniques used and the outcome. This process of recording how a user interacts with a visualization is sometimes referred to as provenance tracking, which is defined by Anderson et al. [88] as the logging of information about how data came into being and how it was processed.

In keeping with the need to customize a visual analytics system, this dissertation expanded the design consideration to incorporate more collaborative methods. Two essential methods are described and discussed in this section, including interaction logging and annotation tracking. Each of this technique represents a unique perspective in extending the visual analytics systems to encapsulate the users' reasoning practices.

Interaction Logging and Capturing Users' Analysis Provenance

While visual representations can aid problem solving significantly on their own, they gain even more power to model a problem when interaction is introduced. Interaction is increasingly seen as central to the process of reasoning with visualization [104, 129, 158]. Lending weight to the intuition that interaction improves reasoning, Hundhausen et. al [77] found that interacting with an algorithm visualization produces better understanding than viewing an equivalent animation.

As previous explained in Section 4.5.1.2, interactions is a key component in visual analytics design. It bridges the visual interfaces with domain knowledge workers. Interactions enable knowledge workers to freely explore the targeted data space, and help them to identify patterns and outliers that are buried deeply in the dataset. This dissertation uses the term interaction in the broad sense defined by Yi et al.: the dialogue between the user and the system as the user explores the data set to uncover insights [177].

In this sense, the relationship between interaction and problem solving has been the subject of much research by cognitive scientists in the field of distributed cognition [78].

In particular, David Kirsh has argued extensively that projection and interaction with external representations are fundamental to human reasoning [93, 90, 92]. Kirsh points to the pervasive use of external representations and interaction with the world in everyday problem solving, and identifies several functions performed by interaction in the reasoning process [90].

A number of previous visual analytics research has focused on understand the utility of interaction logs in terms of retrieving users' analytical processes. Groth and Streefkerk [62] recently coined the term information provenance to distinguish systems that capture low-level user interactions from systems that record the information discovery process in using a visualization. In their model, they focus on recording the users interaction independently from the data in a way that the same set of logged interactions can be applied to a different dataset. Under Groth and Streefkerks definition, many recent visualization systems that record user interactions incorporate tracking of information provenance. Jeong et al. integrated tracking functions into a financial visualization tool and recorded semantic-level interactions that are relevant to the specific domain [82]. Heer et al. presented methods for both capturing semantic interactions within an information visualization system as well as the mechanism for reviewing, editing, and annotating on those interactions [70]. Lastly, in the Aruvi system developed by Shrinivasan et al., the users interactions are automatically stored into a visible history tree [145]. The user can also manually construct the state of the discovery using an interactive node-link diagram, which provides additional detail behind the users interactions.

While all of the aforementioned systems have noted on the benefits of capturing provenance, including communication, evaluation, training, etc., the details of how provenance could be utilized to achieve such benefits is sometimes unclear. A notable exception is the work from Dou et al. on studying financial analysts' analysis trails [49]. They examined the benefits of information provenance captured in a financial visual analytics tool by comparing the captured information provenance to the original users analysis process. Their results indicated that information provenance does not equate exactly to the analysis process and the relationship between the two varied depending on the stages of the analysis. While this research all focused on reviewing a users interaction history, there has been little research in how either data or information provenance could be used.

To better understand and confirm the connection between interaction and knowledge worker's analytical processes, this dissertation further conducted an empirical study that follows on Dou et al. [49] previous work. The goal was to demonstrate that constraining user interactions indeed affects problem-solving through exploring the relationship between interaction constraints, visual representation, and problem-solving performance as measured by response time and score.

The study recruited a total number of 117 participants (86 Male, 31 Female), and asked the participates to search for the best analytical solution using a set of interaction constraints for the Number Scrabble game [50]. The research went through multiple rounds of a refining process to design the interaction constraint conditions used in the study. And the goal was to design constraints that ranged from placing no limit on the interaction to restricting the interaction a great deal. Details of this study and how it was conducted is listed in Appendix A 3.1.

The findings from this study suggest that, there is a clear connection between the nature of interactions available in a visual representation and the types of strategies users tend to develop when working with the representation. The results showed that more confined constraints led to better analysis solutions, and better analysis solutions result in large improvements in scores on the Number Scrabble game. Overall, these results indicate that the search for analytical outcomes can be embodied in user interaction by imposing different constraints, and that certain interaction constraints can lead to a higher chance of deriving a better solution for a problem.

Therefore, this dissertation suggests that degree of constraint is an important dimension to consider when designing interactions for visual analytics systems, although this is not a common way of talking about interaction design in visualization. Specifically, the results of the study imply that highly constrained interactions can impede the discovery of the unexpected, but can also potentially guide the users to consistently identifying the expected findings. With better analytical solutions yielding higher performance, these results demonstrate that the effectiveness of problem solving activities can be captured and improved by embodying information in user interaction.

As shown in Table 19, a number of research on interaction logging are discussed and summarized into categories. In design practice, this dissertation has applied these interaction-logging methods to multiple projects. Details about these design and implementations is described in Section 3.6.

Table 19: A list of categorized interaction logging methods

Log Focus	**Log Elements**	**Examples**
Tracing detailes of analysis sessions	Low level event (e.g. MouseClick, Key Stroke)	Jeong et.al [82],Dou et.al [50]
Replay key analysis frames	Visual States (e.g. visualization parameters)	Jankun-Kelly et al. [80], Shrinivasan et al. [145]
Reconstructing user's analysis process	Low-level events and Contextual information and etc.	Robinson et al. [136], Dou et al. [49]

Annotation Tracking and Content Sharing

Collaboration and content sharing is one key process in accomplishing organizational decision-making [119, 23]. This process emphasizes on the sharing of tacit knowledge between knowledge worker, including individual's analysis findings, progress status, and analysis reports. Collaboration is the most formal inter-organizational relationship involving shared authority and responsibility for planning, implementation, and evaluation of a joint effort [75]. As Monsey [110] further pointed out that collaboration and content sharing brings autonomous organizations together to fulfill a common mission that requires comprehensive planning and communication on many levels.

However, while collaboration is beneficial to advance the analysis processes, it also comes with risk since each member contributes his/her own resources [110]. Wood and Gray [171] further outline the nature of collaboration as a process that ...occurs when a group of autonomous stakeholders of a problem domain engage in an interactive process, using shared rules, norms, and structures, to act or decide on issues related to that domain.

To accommodate the challenges in organizational collaborations, much research has focused on representing the communication and content sharing flow in a visualization system. Specifically in the context of visual analytics, building collaborative visual analytics environments also has a long history [35, 83]. Johnson [83] defined that collaborative visualization is a subset of computer-supported cooperative work (CSCW) in which control over parameters or products of the scientific visualization process is shared. More recently, Burkhard proposed a collaboration process of transferring knowledge between at least two persons or a group of perople [21]. Similarly, Ma [105] noted that sharing visualization resources will provide the eventual support for a collaborative workspace. He discussed existing web-based collaborative workspaces in terms of sharing high-performance visualization facilities and visualizations and findings. He also showed several existing collaborative workspaces such as TeraGrid [17], Many Eyes [107], etc (see [105] for detail). Finally, Heer et al. concluded a compelling list of design considerations for collaborative visual analytics, attempting to identify accomplishments which facilitate collaboration and suggest mechanisms for achieving them [70].

One repetitive key question that comes out of the above research is searching for the tangible contents that can be used to establish the organizational collaboration environment. Naturally, human-communication is based on symbols like language, gesture, and written contents [119]. Consequently, Nonaka et al. noted that a computerized human-communication method should also be based on symbolic contents that would enable knowledge workers to participate in sharing their knowledge, learning analytical practices, and building consensus of decision-making through the use of computer systems [119].

In this dissertation, such symbolic artifacts in the collaboration process is generally regard as *Annotations*. In the context of visual analytics, annotation refers to the process that users externalize their findings, such as data correlation, outliers, patterns or trends, on top of the visualization. Comparing to interaction logging, which focuses on capturing users analysis process implicitly, annotations place the users in the center of explicitly tracking and sharing analysis findings. By annotating the findings, the domain users attach semantic meanings to their analysis findings; so that these findings can then be analyzed, evaluated, reused, and exchanged for the collaborative decision-making.

Consequently, with the utilization of annotations, exchanging expert's analysis finding and establishing organizational collaboration has become more efficient. Communications centered on annotations allow domain users to efficiently share their analysis findings among peers and colleagues. The basic idea to achieve such communication is to share the annotation of an analysis finding, including its cached data and visual parameters, to other users.

Concluded on previous research [164, 31], this dissertation categorizes the design of an annotation sharing mechanism onto two levels: the sharing of the statics annotation level and the exchanging of the dynamic annotations level. This dissertation considers both level of annotation sharing effective; their categorization is based on the difficulty in implement such sharing mechanism.

On the one hand, this dissertation considers the sharing of static annotations. In this sharing method, annotations are typically a static content (e.g. image, snapshot, and bookmarks or comments) that can be shared between one knowledge worker and the others. This sharing mechanism is typically easier to achieve and can be added onto the existing visual analytics system in a straightforward manner. Sharing static annotations allows users to save and share with remote peers additional semantic information about certain data features and visualization patterns. While the remote peers can view and depict the shared information, their flexibility in continuing the original analyst is limited.

On the other hand, this dissertation presents the sharing of dynamic annotations. The dynamic annotations include not only the final products of an analysis, as the static annotations do, but also capture the visual states and its configurations for reuse. In doing so, multiple users can gain access to the same analytical process as the original analyst. These additional collaborators can then review or continue that analysis process within the same interactive visual analytics environment. This allows collaboration between groups of analysts to contribute to the analysis of a large volume of data. It further helps to create and share analysis applications for smaller subsets or contexts of data. In this way, an analyst is allowed to discover an interesting feature using a combination of interactive visualizations, bookmark and comment on the feature, and share the interactive visualization with another analyst who could then be able to contribute to the understanding of the feature.

The details of these two mechanisms are summarized and compared in Table 20. In keep with the diverse collaboration needs for different organizations, this dissertation has designed three visual analytics systems that support the annotation sharing mechanisms for all three aforementioned organizations (Microsoft, USDOT, and Xerox). This examples can be see in their corresponding implementation shown in Table 20

It is worth noting that In existing approaches, users are usually required to manually input notes or drawings to record the semantic meaning of an analytical finding [59]. One drawback of this manual approach is the possible introduction of interruptibility

Table 20: The comparison between sharing static annotations v.s. dynamic annotations

Sharing Mechanisms	**Efficiency**	**Efficiency**	**Effectiveness**	**Information Sharing Flow**	**Design Case**
Sharing Static Annotation	Fixed Image; Textual Information; Drawing	Easy to construct. Can be add on to existing visual analytics system	More effective in a small-to-mid-size collaboration group.	Typically one-way. Information comes from original analyst and shared with other colleagues.	OpsVis (Section 3.5.2)
Exchanging Dynamic Annotation	Parameters that can be applied to in another instance of the visual analytics system	Needs to modify the existing visual analytics system.	Support larger collaboration teams and departments.	Bi-direction, both original analysts and peers can collectively modify and extend the analysis results.	IRSV (Section 3.6.2 and Taste (Section 3.6.2

during the analysis process. This dissertation is currently working on identifying more automated annotation methods, including annotation templates and report templates, to reduce the interruptions to users to attach semantic meanings to their analysis findings.

4.5.2.2.2 Implementation Component: Visual Analytics System Customization

The ultimate goal of collecting and analyzing the domain users' interaction and annotation is to incorporating their analysis individuality into the holistic design of a visual analytics system. As demonstrated in previous section, both interaction logging and annotation sharing methods can enable a visual analytics system to collect such information.

Analysis Evaluation and Knowledge Validation

Although these customized analysis processes were coming directly from domain users, this dissertation emphasizes on the necessity of validating such processes before merge to individual's visual analytics system. Due to individual experiences and understanding, different experts have their own ways of performing analysis processes. Their views of an analytical process may be imprecise, duplicated and even conflict with the organizations generic analytical workflow. Therefore, this dissertation concerns that, if new analysis process or knowledge is not carefully validated, inserting unrelated or incorrect knowledge could potentially degrade the value of the design of visual analytics system.

The validation process is therefore of great importance in the knowledge mapping structure. While verifying and validating diverse domain users' analytical process is difficult in nature, this dissertation has considered the design of such validation process to be applied to two levels:

The first level happens internally in a visual analytics system. The key point for validation on this level is the cost of updating the targeted visual analytics environment. It is important for a visual analytics system to attach costs to system customization. As shown in Figure 42, such cost of customization could be a combined factor of cost of interaction (concluded by Lam [99]), cost of visualization (suggested by Amar et al. [5]), and cost of cognitive overload (proposed by Green et al. [61]). The designed visual analytics system needs to apply threshold to control the cost, and maximize the cost/benefit valuate in determining the need for system customization.

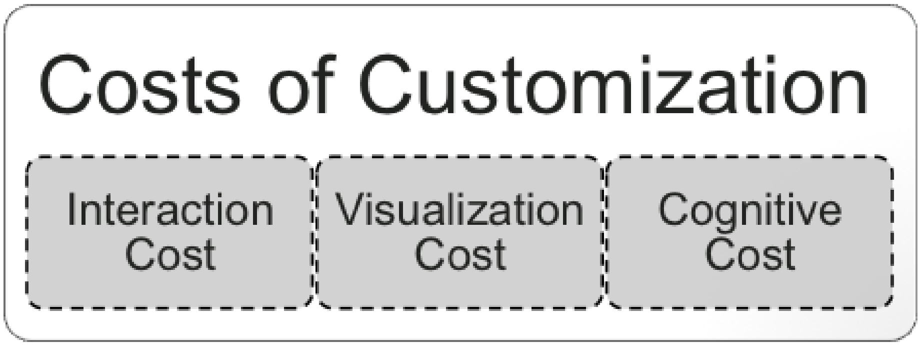


Figure 42: The cost of customization

Therefore, when a customization request is submitted by analyzing user's interaction logs or annotations, the visual analytics system should impose another level of validation on this candidate prior to updating the visual renderings.

Specifically, at a low cost level (i.e. only part of the screen needs to be updated), the visual analytics system should be able to automatically validate the update either through algorithmically applying predefined criteria in the runtime or by users direct instructions. However, if the cost of change the visual representation is too high based on the analysis candidates, such as a drastic change in graph layouts, user should be notified as to the possible changes prior to the visual actions. For example, such validation processes can be derived from Heer et al.'s [69] elaborated study on differentiate the cost of individual graph layouts.

The second level requires the user's participation. Typically, after validating the analysis updates internally, the visual analytics system should present the changes to the user through visual hints. At this level, the visual analytics system should be able to allow users to accept or dispute these update candidates, based on their judgments on whether it is relevant to their on-going analytical reasoning processes.

This validation presents an important feedback loop in the entire customization process. Especially, the proper utilization of this process would affect the construction of the user's individual knowledge structures. For example, the KEF system [39] presented user with the ability to review the suggested materials and accept/dispute based on user's preferences; through actively tacking the visual changes, the HARVEST system [144] also provides users capabilities to revert the visual updates.

In summary, by enforcing the above two level of analysis update validation, this dissertation believes that a visual analytics system could then begin to incorporate more accurate and suitable individual analysis processes. While currently there is no definitive measure for such cost, the research presents the supporting evidences of the need for such measure. As discussed in Chapter 6, the identification and verification of such cost would be one of the most important directions of this research.

Customize the Visual Analytics Systems

As mentioned at the beginning of this Chapter 4.5.2.2, there are three needed customization supports; First, it is the ability to shuffle the analytical components in the system, rearrange the sequences and combinations of the generic analytical processes. Secondly, it needs to support individuals abilities to collect analytical findings and trace the analysis trials that lead to their findings. Finally, these organizations need to have efficient means to share these analysis finds between knowledge workers, and provide an effect environment to support collective decision-making. Consequently, after evaluating the cost of customization, as shown in Table 21, there are three ways to individualize the system: refine analysis focuses, update data model and customize visualization combination. Examples of the implementation of these customization methods can be found in previous sections (Section 3.6.2.1 and Section 3.6.2.2).

Table 21: Three system customization methods

Mehod	**Description & Examples**	**Implementation**
Update Data Model	Based on users' data focuses, modifying and updating the underline data model. For example, the visual analytics should prioritize the more frequently used statics based on users's analyzing histories.	Section 3.6.2.1 Section 3.6.2.2
Customize Visualization Combinations	Rearranging the visualization combination based on the users' interaction logs and annotation histories. Built upon a modular design, the visualization system needs to adjust the primary and entry view of the system based on the behavior analysis.	Section 3.6.2.1
Refine Analysis Focuses	Utilizing the recorded annotation, the visual analytics system needs to understand the important analysis focuses for a user. It needs to guide the users toward that analysis focus through interactions	Section 3.6.2.2

4.5.2.3 Summary: the *User-centric Refinement* stage

The ***User-centric Refinement*** stage is proposed to incorporate the individual's analysis processes. In this stage, the first step is to deploy the design visual analytics systems to domain users. This deployment process involves two typical training sessions (e.g. general visualization concepts session and analysis scenarios session) to motive the targeted users to use the system and adopt the new ways of performing analytical tasks. Since the user's acceptance of a system is a key factor in the success of incorporating the individual's analysis processes, the *System Deployment and User Training* step is therefore of great importance in this design stage.

The next step in this stage is the *Usage Pattern and Customization* step. This is the key step to support the incorporation of individual analysis processes. In this step, two essential methods are described and discussed in this section, including interaction logging (implicit method) and annotation tracking (explicit method). Each of this technique represents a unique perspective in customizing the visual analytics systems to encapsulate the users' reasoning practices.

Although these customized analysis requests were collected directly based on domain users' analysis behavior, this stage emphasizes validating such requests before merge to individual's visual analytics system. An important concept discussed in this stage is the cost of customization, which is directly associated with the validation of the system customization process. After evaluating the cost of customization, three typical methods—namely refining analysis focuses, updating data model and customizing visualization combination—are described to individualize a visual analytics system.

4.5.3 Conclusion

This chapter presents the two-stage framework for designing visual analytics systems in organizational environments. This framework emphasizes the benefit and efficacy of incorporating both general domain analytical workflows and individual analysis practices. Both the ***Observation and Designing*** stage and the ***User-centric Refinement*** stage in this framework aim at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their analysis processes.

CHAPTER 5: EVALUATIONS

5.1 Objectives

In this chapter, this dissertation presents extensive evaluation for all three designed visual analytics system, namely Taste, IRSV and OpsVis. The goal for these evaluations aims at thoroughly validate the effectiveness of the aforementioned design considerations, and verify their usefulness in instructing the design of a visual analytics system. Three detailed human subjects experiment is conducted. Note that this experiment is not a comprehensive validation or proof of the Two-stage Framework. Such a proof is inline with the future work of this dissertation. Instead, this experiment provides some partial support for the claimed predictive power of the framework.

5.2 Overview

All of above systems, including Taste, OpsVis and IRSV, were designed following our recommendations. These visual analytics systems are implemented to support the analytic processes encountered in organizational environments. Through iterative prototyping processes, each was tailored to the analytical workflow of its target domain. As shown in Figure 21 and Figure 12, the design recommendations actually incorporated within each system are illustrated separately by marked checkboxes. We also conducted user-studies to evaluate the utility of these systems.

Instead of emphasizing technique details, the following sections focus on evaluations for the effectiveness of our systems to support domain analysis processes. Specifically, the users' feedback and comments are summarized to assess the performance of the aforementioned three systems for their efficacy of facilitating the common task activities.

5.3 Taste: Supporting Business Information Analysis

To evaluate Taste, two studies were conducted in this dissertation: (i) a controlled laboratory study to assess the usability and utility of Taste's interface with a fixed set of data and (ii) a field study to evaluate the effectiveness of Taste, which embedded with the tree retrieval cues, in aiding people's information seeking process in users' own work environments.

To evaluate Taste, 21 Xerox employees participated in both lab and field studies using the tool. In the following sections, this dissertation presents how Taste was found to be useful and effective in facilitating each of the six common task activities in the domain analysis process. Detailed statistical results are presented along with the users comments and feedback:

5.3.1 Study Design

Study Goals and Experimental Setups For both studies, two conditions were examined: (i) the use of Taste; and (ii) the use of regular Mac OS X tools [1], hereafter referred as RT (regular tools), including Microsoft Office for Mac and Google Desktop. All participants experienced both conditions in a counter-balanced order.

For the field study, the data capture module was first deployed to each participant. Document logging for Taste was unobtrusive: there were no reported interruptions to the user, but each day the logger were checked to be active on all participant machines. Taste was able to sample around 200 documents during an 8-hour working day. The total number of documents sampled over the period of 1.5 weeks per participant was in the range of 1000 - 1200. Participants were instructed not to look at their data during collection. The Taste interface was not installed on any of their machines during this period.

In preparation for the lab study, Taste was used to collect one-month's worth of workstation activity data. The resulting data became the test corpus. This data was diverse, containing different types of documents, web pages, images and emails. In total, there were 7,419 data objects, with over 24,275 pages. None of the participants had any prior exposure to this data. No new additions were made to the log during the study; so all participants accessed the same data set. For purposes of comparison, this data was also indexed by both Apple Spotlight and Google Desktop search. All participants were presented with the nature of the data set and locations (folders) of documents before the study began.

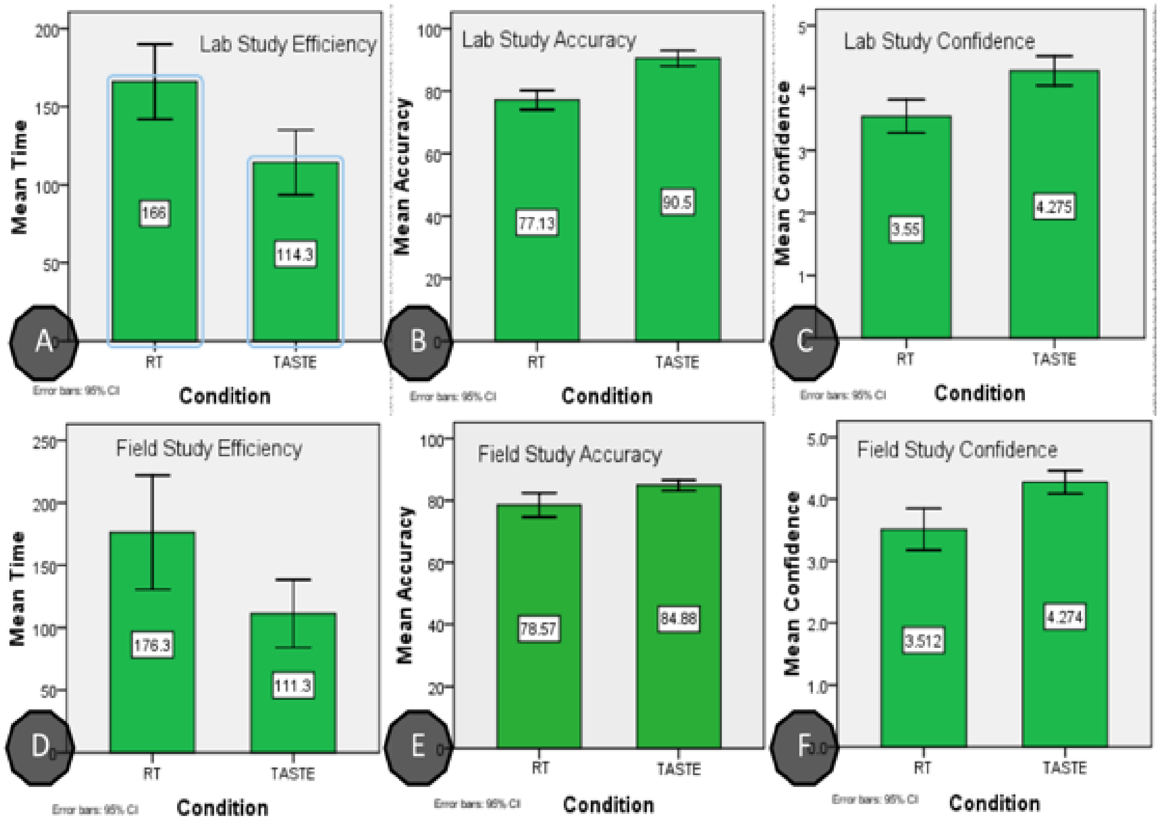


Figure 43: Study results for the three measured factors: efficiency, accuracy, and confidence. Lab study results are on top. Field results are on the bottom.

Participants All of the participants from the Xerox coorporation were volunteers consisting of researchers, administrative staff, managers and business development staff. For the short-term controlled lab evaluation, 12 participants (5 female and 7 male) were recruited through email lists and by acquaintance. For the field study, six participants (1 female and 5 males) volunteered to use Taste on their personal data for 1.5 weeks during which their document activity was continuously logged. All were proficient users of the Mac OS X operating system and its applications. None of the participants had any prior knowledge of this project nor any prior experience with Taste.

Procedures Since the participants would inevitably get more familiar with Taste and the sample dataset during the process of our studies [102], control steps were taken to isolate learning factors. In particular, each participant was asked to attend two sessions, separated by a two day time period. It is expected that over this time period participants would forget most of the details of the data set.

To balance performance gain, the sequence of conditions (using Taste vs. Regular Tools) across participants were randomized. In addition, the allocation of time to tool was balanced across sessions. Each participant used either Taste or RT at the first session and switched to the other tool two days later.

To ensure that the study results were comparable across sessions, participants were given the same task questions for each tool; participants were asked to find information on the same time frame or about the same person.

Retrieval Tasks and Study Measures In total, each participant was asked to carry out 12 tasks (6 tasks in 2 conditions). These tasks were identified based on our pre-design interviews, all of which are at least common in corporate daily tasks, if not predominant in all jobs. As shown in Table 22, these most representative tasks were selected to evaluate the capability of Taste.

For each task, the following factors were collected and analyzed to measure the utility and usability of Taste:

Accuracy: the percentage of accurately retrieved items in the participant's answers. In the lab study where standard answers were known, this factor was measured by comparing participants' answers to the standard answers; in the field study, this factor was analyzed together with each participant after the two study sessions.

Efficiency: the time (in seconds) spent on answering each question.

Task 1	Please use tool(s) to find the most mentioned contacts within a particular time frame.
Task 2	Please use tool(s) to find the most visited information sources within a particular time frame.
Task 3	Please use tool(s) to find and list the user's recent activities with Mr. Manager X.
Task 4	Please use tool(s) to locate a recent threaded email with attachments.
Task 5	Please use tools(s) to locate such an email in the file system and find its attachments and, if possible, find other related documents.
Task 6	Please use the tool(s) to prepare an activity report for the user's activities in time frame A. Please write down contents and artifacts.

Table 22: Task questions for both conditions in the laboratory study (the word 'tool(s)' was replaced to reflect the condition)

Confidence: a 5-point Likert scale score measuring how confident users were in the accuracy of their answers.

When participants completed their tasks, each participant was asked to answer two open-ended question about their experience using the assigned tool, and also asked them to score their answers on a 5-point Likert scale: 1)"How well do you think the provided tool covers your retrieval cues?"; and 2)"How do you like the design of the tools provided?"

5.3.2 Statistical Results

The results for both studies were significant. In general, Taste has significant advantages over regular tools in the following four aspects:

Taste provided better retrieval accuracy Results from both studies suggest that Taste provided participants with more accurate retrievals. In our lab study where participants had no prior knowledge about the data, the results suggested a significant ($ANOVA : F(1, 78) = 45.49, p < 0.001$) 17.3% accuracy gain (as computed by comparing mean values (90.5-77.13) /77.13) during both sessions, as shown in Figure 43 B. Even though participants had prior knowledge about the data, our field study also indicated a significant difference ($F(1, 82) = 8.80, p < 0.004$) between Taste and the regular tools on the accuracy of retrieving information. As shown in Figure 43 E, Taste still delivered nearly an 8% increase ((84.88’78.57)/78.57) in retrieved information.

Taste significantly improved retrieval efficiency As shown in Figure 43 A and 43D, both studies suggested significant reductions in participants’ information retrieval time, with 36.8% ((176.3-111.3)/176.3) reduction in the lab study and 31.1% ((166-114.3)/ 166) in the field study. The ANOVA results (lab: ($F(1, 82) = 6.13p < 0.015$) and field: ($F(1, 78) = 10.84, p < 0.001$)) show that Taste helps participants perform their tasks more efficiently.

Taste largely increased participants’ confidence Initially, since participants had prior knowledge about the location of their own data, it was expected that the confidence value to be similar between Taste and RT in the field study; on the other hand, Taste was expected to increase user confidence in the lab study. While the lab results ($F(1, 78) = 17.12, p < 0.001$) partly supported our expectation, the field results yielded a significant positive confidence increase (ANOVA ($F(1, 82) = 16.16, p < 0.001$) when using Taste. Therefore, as shown in Figure 43 C and 43 F, participants trusted Taste more than the regular tools for retrieving their document activities.

Taste provided a more advanced interface In response to our post-task questions, Taste received an average 4.57 out of 5 in supporting users’ retrieval cues, even though participants had interacted with Taste for less then an hour. By contrast, the regular tools scored a 3.36 on these questions. Taste also received an overall score of 4.34 out of 5 for usability, suggesting that participants felt comfortable using Taste to perform the tasks. One participant noted, ’This interface [Taste] provides me more control over the data. ' it makes me feel more confident in searching for information'.

Given that participants had prior knowledge of the collected data, it seems likely that the accuracy of their answers depended largely on the tools provided. Since Taste provides a cohesive visual interface that incorporated highly rated retrieval cues, many participants positively rated Taste as an aid for information seeking. The field study supported this result by suggesting that Taste performed better than regular tools on our three measured factors. One participant commented that, ' By looking at all the [Taste] interface, I can easily relate all the information together and effectively examine my activities from different aspects.'

Even if there is no previous knowledge about the data, our laboratory study suggests that Taste's interface is designed sufficiently enough for users to follow important clues in the tasks and outperform the results from using regular tools. One participant noted that, '[Taste] is very good at giving a quick impression of data across the board. I like how you can mix different types of files and people together and represent them interactively.'

While Taste has only been used by a small number of people so far and for a short period of time, it appears to be a promising technology. Participants enjoyed using Taste to retrieve information and shared comments like 'I can definitely see myself using it regularly'; and "it can be quite helpful when I need to quickly put together some research reports'.

The following sections presents the detailed feedback that demonstrates the usefulness of Taste in helping corporate employees:

5.3.3 User Feedback on the Design Consideration

Gathering content into a unified visual interface At the heart of Taste is a transparent, real-time, contextual data capturer, which was designed to capture the user's activities around office documents, calendars, emails, etc. Taste creates an index of documents on a user's machine, and logs information about the user's activities with these documents. Taste stores this information, along with copies of the documents, in a unified repository. All captured information is then indexed and grouped with its related documents, and is interactively presented to the user through Taste's visualization interface, as shown in Figure 12 (Right).

All participants indicated the usefulness of this unified interface. They agreed that integrating multiple information streams into a single interface sufficiently encapsulates their actionable knowledge, reducing search times for related information. They believe this could greatly assist them in gathering and aggregating contents from multi-channels

Enable facet search for content filtering As shown in Figure 12 (A), Taste utilizes the Facet view to aggregate both the documents and the people with whom a user has previously interacted. This visualization allows the users to filter and sort information based on automatically extracted data facets, including type (person or document) and format (email, text document, etc.). Facet view further sorts and displays document activity by importance, which is measured by frequency and users' dwell time.

When presented to the participants, they spontaneously formulated a variety of facet filters to find information. They were generally satisfied with the efficiency of using Taste to 'slice and dice' information, and appreciated the flexibility to perform customized analysis.

A common suggestion was to be able to also create formulas to sort the documents with customized measures. One analyst indicated that introducing customized time factors (such as increasing the importance of a more recently created documents over older documents) would be especially useful for filtering.

Interactive Information Analysis Besides the facet view, Taste also supports high-level content analysis based on both temporal information and content keywords (See Figure 12 (B) and (C)). Taste utilizes the temporal view to show how a user's activities unfold over time, and presents the temporal trends and patterns of a user's document activities. This view allows the user to interactively drill down to a specific time, and helps the users examine the content, which occurred in that time span. In addition, an entity tag view is used to enable fast entity browsing. This is implemented using an automated entity extractor, which extracts entities, such as company name, contacts, etc., from all of previous documents. As shown in Figure 12 (C), Taste enables users to focus on a specific entity, and examine any information related to it.

In the low-level view, Taste incorporates a detail view (Figure 12 (D)) for depicting a single document from multiple perspectives, such as its related temporal information and other versions of the document. All views in Taste are coordinated, such that updates in one view are immediately reflected in the others.

In both lab and field studies, Tastes was compared with other existing tools to assess its analysis capabilities. The participants were generally positive about Taste's effectiveness for retrieving and analyzing business information. All participants agreed that the ability of viewing information from different granularities could largely help them filer and analyze information.

One suggestion was to provide finer-grained categories, and display more information for entities. One participant suggested that the current categorization is too broad by referencing a common expectation: Instead of general, high-level categories like browsing, email, etc., usually the categories of interest are more narrow like "email with Bob" or "browsing about JAVA")

Using Storytelling to generate and share reports By utilizing an interactive storytelling view, shown in Figure 12 (D), Taste allows users to interactively collect evidence, annotate it, and share it with others. The storytelling view allows the user to take a more active role in information tracking, and enables them to express the information relationship based on their own knowledge. Whenever a user comes across an interesting information object in Taste, they can directly add that object to a new or existing story view. Once an element is in a storytelling view, the user can further annotate or tag it, and can group different story elements based on their reasoning logic.

The story created by one user around a collection of people and documents may be of interest to other users as well, so Taste allows stories created in one instance of the system to be shared with users in another instance. Analysts who receive these shared stories, are able to modify them based on their understanding of the topics, and add or suggest removal of story elements. By sharing their stories about document activities, groups of employees can now understand those activities better, and improve information analysis for all members of the group.

While the story feature is new, many participants found the idea of collaboratively searching for information interesting and the way Taste approached feature practical and useful. Although there was no setup for a collaborative environment for participants due to privacy concerns, participants were still interested in utilizing the Story view and tried to share findings between different instances of Taste.

5.3.4 Summary: Taste Evaluation

While Taste has so far only been evaluated by a limited number of participants (albeit actual target users), it appears to be a promising technology and a successful design. Based on the feedback from participants, it is indicated that the design of this visualization successfully encapsulates the actionable knowledge and supports the analytical workflows that are essential for business information analysis. Through the on-going collaboration, this research is further refining its basic functions and enriching it with more advanced features.

5.4 IRSV: Facilitating Bridge Maintenance Planning

The evaluations of IRSV and its variations were performed iteratively throughout the collaboration, and were mainly conducted with a group of bridge managers from both North Carolina DOT and Charlotte DOT (CDOT). These 12 (10 male, 2 female) bridge managers participated in at least three sessions of onsite evaluations

5.4.1 Summative Expert Evaluation

First, a training session (30 - 40 minutes) was conducted with the participants. During this session, the design of the system and the utilities of each visualization were demonstrated through interactive training session. Then, bridge managers were invited to perform their domain analyses using the system for 45 - 60 minutes. During this hands-on process, these bridge managers were encouraged to carry out these analyses in a think aloud manner. The details about their analysis processes were observed and documented. Finally, the summative evaluations were conducted using a set of semi-structured questionnaires. These questionnaires are used to collect bridge managers feedback and comments about the IRSV system. Since bridge managers may need time to familiarize themselves with the all features provided by our visual analytics system, several email follow-ups were also conducted to see if there were additional comments they would like to share.

As of the current dissertation, the communications with bridge managers on their comments of the system were continued in the past 7 months. The results from this longitude study have provided the research significant insights for the continuation of the collaboration project. Although the degree and depth of analyses differed in each evaluation, the bridge managers generally agreed that our system provided more analytical capability than any existing BMSs, and that it is flexible enough for them to quickly incorporate the use of the IRSV systems into their daily routine.

In the following sections, this dissertation first presents the analysis scenarios that are identified together with bridge experts. These scenarios are used in the evaluation process. It further presents the summarized feedback from these evaluations, and assesses the systems (multiple IRSV variations collectively) for their effectiveness in facilitating each task activity encountered in bridge maintenance planning.

5.4.2 Example Scenario

5.4.2.1 Investigating Causes for Bridge Deteriorations

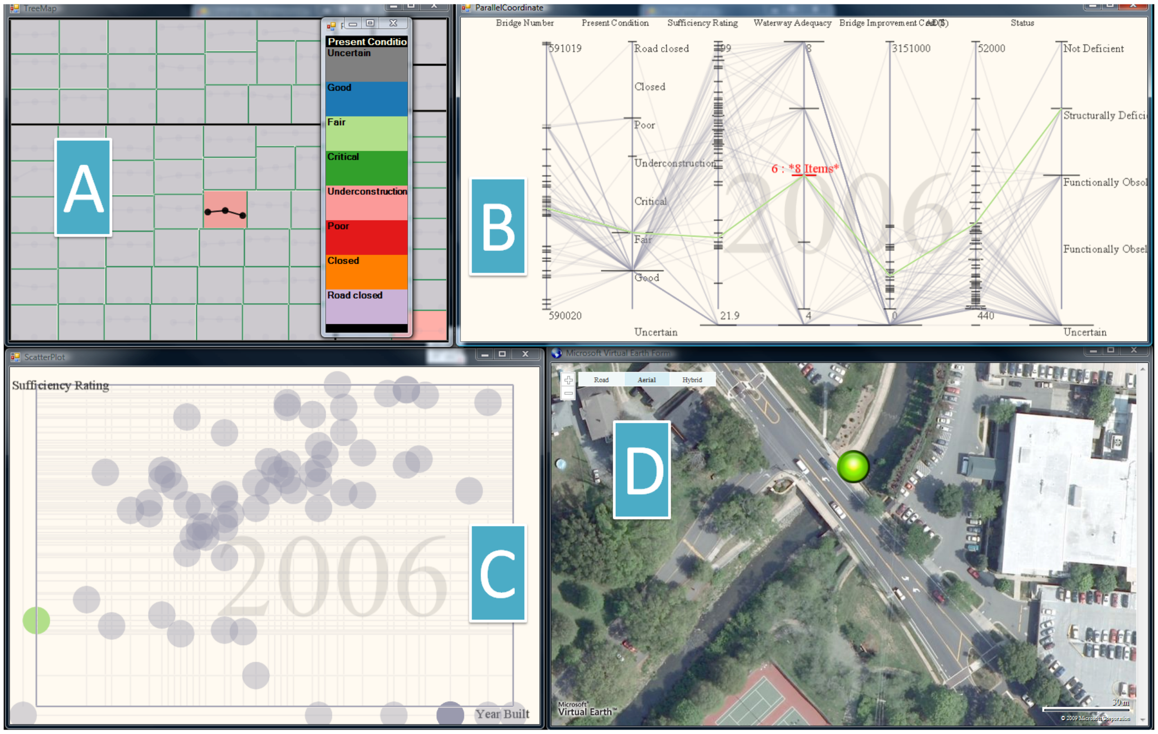


Figure 44: (A) A significant downward temporal trending indicates an unusual pattern (B) Using PCView to compare different structural attributes (C) Examining a certain bridge on SPView, indicating this is the earliest constructed bridge in the database (D) The Geospatial view shows that this bridge is constructed on top of a river stream.

Identifying and understanding the cause of bridge deterioration is a key step for bridge managers to come up with corresponding maintenance strategies. Based on our discussions with 9 bridge managers and bridge experts, it has been observed that there are generally three stages in achieving this step, namely, selecting bridge candidates, detailed examination, and identifying potential causes for damage. The following scenario was identified together with 5 bridge engineers (4 males and 1 female) from Charlotte DOT's bridge management team for their annual bridge maintenance planning. During this process, these engineers were encouraged to collaboratively discuss the issues in a think aloud manner. Their analysis processes were further documented to help them to familiarize with the system. Along the hands-on period, additional explanations were also provided to explain certain features in IRSV system.

IRSV system was initialized with data from previous three inspection cycles: years 2000, 2004, and 2006. The bridge management team started the maintenance process by searching for bridges with significant changes in sufficiency rating in the previous years. They utilized the small multiples view to see if any interesting bridge changing patterns could be identified. As shown in Figure 22 (E), the team found a set of bridges with warmer colors in the small multiples view, and they also identified several bridges with significant downward trends in the past years. By highlighting these bridges in the scatter plot view (see Figure 44 (C)), the team noticed that one of them was actually the oldest bridge in the Charlotte area. Suggested by both the small multiple view and the scatter plot view, this bridge actually shared the lowest overall rating in that year and had had drastic deteriorations since 2004.

To have a closer look at the bridge, the team used our geospatial view and zoomed into the bridge to check its surrounding environments. As shown in Figure 44 (D), this bridge was constructed over a river stream, and had supported high traffic volume because it had been chosen as a part of a detour route for a major interstate highway. These findings immediately raised several questions: could the bridge's deterioration be caused by water erosion, overloaded traffic, or flood damage? Although these were all possible causes of the deterioration, bridge managers had no definitive answers to confirm these hypotheses by looking at the geospatial view alone.

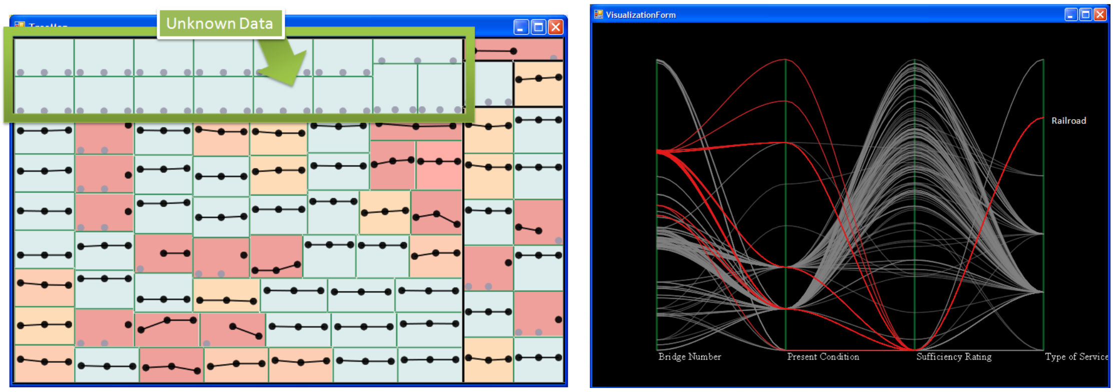
Trying to verify their hypotheses, the management team started to find clues from the structural reports of that bridge. By plotting the corresponding criteria in the parallel coordinate view, they found that the traffic amount on that bridge had not changed significantly in the previous years, and therefore ruled out the possibility of traffic pattern being the cause of the deterioration. However, the PC view showed that the water adequacy rating had dropped significantly during the past two inspections, suggesting the bridge had undergone severe water damage. To extract more detail, the team brought up the bridge's detailed structural view. As shown in Figure 23 (D), the supporting pillar for this bridge had shown heavy warping, and the bridge showed clear marks of water erosion near the bottom of the pillar. A quick reference check on the county's flood history confirmed that three significant flooding took place in years 2003, 2005, and 2006 around that area, which gave the bridge managers significant reasons to conclude that water damage, especially flooding, was a key factor in causing the deterioration of this bridge.

Given the poor condition of its supporting structure, the bridge managers concluded that this bridge definitely needed maintenance attention. After the exercise, the management team commended the effectiveness of our system in assisting the identification of the deficient bridge, as well as the cause of the deterioration. Although simple, this scenario demonstrates a successful application of the IRSV tool in an actual domain analysis process.

5.4.2.2 Augmenting Visualization through the use of an Ontology

According to bridge experts, water erosion and flooding can cause severe damages to bridges. The pattern for this type of deterioration is in general typical along river streams. In this scenario, we demonstrate how our system could help bridge

Figure 45: Close examination of the geospatial view shows that although these three bridges are on the same river stream, their conditions are different. The bridge over the upper stream is currently under repair and reconstruction.



(a) (b)

Figure 46: (a) A large group of unknown data is shown in the temporal view, which lead to the search of its cause. (b) Visualization views indicating that these are Railroad bridges.

managers to quickly identify the cause of unexpected bridge deteriorations through the knowledge internalization process. This scenario is identified together with city of Charlotte bridge management team during their examination of causes of water damage.

Since the criterion for "bridge above water" has already been externalized in our ontological knowledge structure, the bridge managers can easily highlight all these bridges in the geospatial view and examine them individually. Through quick examination on the geospatial view, the bridge managers immediately noticed an interesting pattern in South Charlotte. Although located over the same river, as shown in Figure 45, the three bridges over that river showed different "present conditions". The one over the upper stream has already been filed for replacement and has been under construction. However, the other two are still in good condition. This pattern is interesting because if there was a flood, all three bridges should share similar deterioration patterns; or at least, they should deteriorate at a similar pace. Even though temporal information suggests that these bridges were built at similar times, the changes in their conditions are drastically different. This inconsistency raised the bridge managers' interests.

After a detailed examination of these bridges in the geospatial view, the bridge managers realized that the cause of this inconsistency was due to the different turns of the river. According to one of the bridge managers, although there was flood in both the upper and lower parts of this river, the bridge over the upper stream received the most impact since there were no bends in the river before the water hit it. On the other hand, due to the slow down of the river's speed when the water passed the second and third bridges, these two bridges received much less impact. Based on this observation, the bridge management team was able to quickly identify and internalize this pattern and re-use it for future reference.

In this scenario, the bridge managers gained insightful knowledge from interacting with our visualization system and incorporated it into their tacit knowledge (internalization).

5.4.2.3 Updating and Sharing Knowledge through Visualization

Since managing bridges is a complex process that often requires precise analysis, it is important for a bridge analyst to quickly determine the most relevant information to focus on during an investigation. In this scenario, we demonstrate how our system facilitates bridge experts through the externalization of their discoveries and sharing of the findings (collaboration) to filter out unnecessary data and focus on analyzing the most relevant information.

A local bridge expert was using our visualization system to explore the bridge distributions around the Charlotte region. After a quick examination of the temporal view, the expert noticed that a large group of bridges did not have any ratings information (Figure 46(a)). Based on this bridge expert's experience, this situation was most likely caused by two reasons: one, it could be caused by a loss of data or errors during the data entry process. Two, these bridges could be outside of the bridge management team's jurisdiction. As shown in the coordinated visualization views (Figure 46(b)), the bridge experts identified that these bridges were all railroad bridges, which fell into the second category.

Since the city bridge management team is not responsible for maintaining these bridges, showing them together with other bridges can be confusing. In order to reduce this confusion, the bridge manager created new rules in the ontological knowledge structure to identify and filter out these railroad bridges. Other bridge managers of the same team will then be able to reuse these rules to reduce the irrelevant information and concentrate on the relevant bridges.

This scenario shows how a user could gather information during visual exploration and further update (externalization) and share his knowledge discoveries with other co-workers (collaboration).

5.4.3 User Feedback and Evaluation Results

Integrating heterogeneous data into one interface As shown in Figure 21, IRSV provides bridge managers with a unified content interface that combines multiple streams of bridge information. It can incorporate a range of data sources, including National Bridge Inspection Standards (NBIS) datasets, high-resolution aerial images, and Light Detection and Ranging (LIDAR) scans. In addition, IRSV provides an advanced feature, incorporating knowledge contents from an ontological knowledge structure. As detailed in our previous report [167], using a service-oriented-architecture, IRSV has been extended to communicate with the knowledge base, access and fetch the inference results, and present them in a cohesive visual interface.

Through comparisons to existing bridge management systems, it was clear that IRSV was appreciated for its efficiency in contents aggregation. All participants considered the visual interface well addressed their information retrieval needs, representing cohesive and useful for bridge information. Moreover, they were excited about the ability to access and follow prior practices and recommendations that were embodied in the knowledge base.

Customizing analysis workflows Because it was built with a modular architecture, IRSV allows bridge managers to extend the system to incorporate advanced visualizations and more effective data models. Each visualization component integrated within IRSV was designed to be interchangeable with other equivalent visualizations. Furthermore, IRSV provides bridge managers with the flexibility to combine and sequence different visualizations to fit their individual analysis routines.

All participants appreciated the flexibility of the interface, finding it useful for customizing the system to only utilize the necessary visualizations in their particular practices. They spontaneously formed a variety of visualization combinations in order to find bridge assets. The most common strategy used was to combine a geospatial window with scatter plot view to gain information for the most recent changes of a particular bridge. A manager from NCDOT further pointed out that,"[IRSV] will greatly shorten the catch-up time between my learning to use the system and my actual use of it."

Analyzing information from multiple aspects All participants noted that IRSV provided a visual exploration environment to help them analyze information from multiple aspects. The capability to perform not only geo-temporal analysis, but also structural analysis was of great value to their decision-making process (See Figure 21 (G)(I)(H)). One of the managers commented that, "[the] linked visualizations provide me with a cohesive understanding about the data that I am working on. It reduces the time I spent on manually searching for information, and helps me focus more on the task itself."

In particular, seven out of the 12 bridges managers pointed out that the temporal analysis in IRSV provided them with the capability to effectively monitor changes in bridge conditions and identify maintenance candidates. In addition, after familiarizing themselves with the concepts and usage of the visualizations, most bridge managers (9 out of 12) noted that the capability to examine bridge structures simultaneously from multiple levels (overview and detailed view) allowed for effective transitions from examining large amounts of data to inspecting bridges one at a time.

Evidence collection and report generation As shown in Figure 21(J), IRSV also supports interactively collecting, annotating, and sharing analysis findings between different collaborators. Using a web interface, IRSV treats individual visualizations and group workspaces as collectable items. It enables bridge managers to directly drag and drop these items into a sandbox, designed to collect all the findings and sort them temporally. IRSV further allows bridge managers to use the collected evidence to support their analysis hypotheses and create analysis reports. The bridge managers can directly combine findings that can support their reasoning and share them with colleagues, through built-in sharing channels or emails.

Most participants found the idea of collaboratively managing bridge information intriguing. They consider our approach practical and useful for creating preliminary analysis reports. There was significant interest in utilizing the features that allowed evidence to be reported and shared with others. While these features are still being refined, the IRSV has shown great potential to support the inherently collaborative nature of bridge maintenance planning.

5.4.4 Summary: IRSV Evaluation

In summary, IRSV was designed by following our design recommendations set forth earlier in this paper. It has been deployed to USDOT for daily use and testing. Based on feedback from bridge mangers, IRSV appears to be a successful design and a useful visual analytics system that effectively supports the bridge maintenance management process. The effort to enrich IRSV is still on-going; the research with bridge mangers are still continuing to identify new actionable knowledge that requires advanced features, including web-based collaboration and post-analysis.

5.5 OpsVis: Enhancing the network management operations

In section 3.3.1.2 above, this dissertation presents details about the initial meetings and observations conducted with ABCH. After these initial interactions, a rudimentary prototype was delivered to the ABCH team aiming to replace (or supplement) the multiple tools and screens, with a single visual perspective, supported by data fusion across the underlying topology and usage data stores. The response to this formative evaluations was extremely positive and encouraging. However, in the same breath, the team pointed that the prototype entirely missed several key data sets, and so could not, at that time, provide any real help. This learning led to a key turning point in the OpsVis project'a key component would have to be extremely flexible data integration in a centralized system. At this point, the XAML interface was invented to the data, completely re-factored and generalized the software.

In addition, the following three scenarios were identified to discuss the utility of OpsVis for those most critically analysis processes that are in need of visualization support: health monitoring; software upgrades; and crisis resolution.

5.5.1 Scenarios

Health monitoring Of the tools operators use today, most are request-driven, waiting for a user to make specific queries; the rest use alarms to alert operators to changes in status. Unfortunately, this structure makes it difficult to monitor the health of systems smoothly. Operators are unlikely to notice a problem until it goes very wrong, setting off an alarm; fluctuations that involve some servers running abnormally but not bad enough to be at alarm levels will not be noticed at all. A visualization that allows operations staff to know at a glance how the system is operating would help anticipate problems early.

Software update No matter how well tested, deploying a software update is a fragile, multi-step process: each server must be taken off active status, updated, and brought back on-line. In the process, a far-away service may suddenly discover that it depends on a now-disabled function, or long-dormant bugs may come to the surface. Operations teams monitor upgrades carefully - continually making decisions whether to continue the upgrade process. Monitoring the progress of an update and tracking its effects on the rest of a service is critical to correctly making these decisions.

Crisis resolution When a true crisis occurs, operations staff use their extensive knowledge of the system configuration to try to figure out what factor is at fault. Visualization can help surface regularities in the failure, and give the operations team a fast way to examine their active data.

These scenarios were further used in the summative evaluation with ABCH team. By covering these analysis areas, the ABCH team recognized the utility of the enhanced visual analytics systems and considered it can provide real help.

5.5.2 Expert Evaluation

At the next interaction with the ABCH team, there was quite a bit more excitement, as the data needed for systems management could now be brought together and effectively displayed. However, the team instantly recognized artifacts that departed from reality. The errors derived from data staleness and integrity issues in operations' central inventory and topology store. As a shared resource, this store has a certain inertia. The store changes more slowly than individual cloud services, and operational processes sometimes "lie" to the store by inserting information morphed to agree with operational processes that are required by the store, but in partial discord with service reality. To get beyond these problems, the system was tested to handle upstream (as the operators' suggested) to the data that feeds the shared store.

With clean data and useful views (validated in the lab on historical data), the next series of interactions with the ABCH team shifted to installing the prototype in production. The next set of hurdles involved the uncovering and working through of important albeit nettlesome differences between the lab and production environment; for example, code incompatibilities between 32-bit and 64-bit implementations. In addition, the team had certain expected values of data; they wanted to highlight with colors values that were out of these ranges. Thus, for example, any CPU value above 60% is too high; color gradations should be saved for CPU values between 25 and 50. When the color scale was re-adjusted, the team was able to catch times when multiple machines were surging. Having overcome these incompatibilities, OpsVis was successfully deployed into ABCH production.

Comparing the current version of the system to the first version shared with ABCH, several visual differences were noticed. The original version did not represent the three clusters, which now divide the image into three parts.The previous prototype had originally included a load-balancer unit, which the ABCH team felt added excessive complexity, and had shown network traffic.

To date, using the enhanced system, working with real time data, the response of the ABCH team has been more positive. The configuration mechanism allowed the team to easily customize data sources and views to produce a new and useful system-wide view. OpsVis's color gradation design enabled the ABCH team to easily understand a node's health relative to other servers in the same cluster.

From comments gleaned from the team, this provides a huge leg up in reacting to problems to quickly localize the root cause of problems. For example, the ABCH tuned their data sources to focus on server processing outliers. The resulting view, for example, makes it trivial to spot the condition when one server's job processor load is larger than 60% while its peers' are only around 40%. The team expects even peaks to stay under 40%, which suggests that the former server is likely having a problem. In contrast, if all or most servers are above 60%, the team can see at a glance that there is a systemic problem or overload.

What's more, while the ABCH team ran OpsVis, they found meaningful patterns not easy to depict in the original tools. For instance, OpsVis directly depicts activity characteristic of the system's backup procedures. During the backup process, the contents of some databases are merged together, leaving others emptier. This creates a distinctive alternating pattern of bright and dull green stripes that stand out in views of the database clusters (Figure 19 (right)). During major upgrades, all eyes are on screens, and seeing this sort of detail assists in monitoring performance and correctness.

ABCH has suggested other features that would make OpsVis more useful. They would like additional data monitoring layers to assist with simultaneous comparisons with different data types. Although OpsVis can already be used in this manner, it requires a cumbersome configuration and style of usage. More importantly, the team wants the integration of service controls, where OpsVis becomes a dashboard for controlling the system as well as visualizing it, with real time views of response.

In addition to working with ABCH, other network teams were also invited to discuss how they could adapt OpsVis. At an internal showing of OpsVis, the OpsVis demo was visited by a wide variety of teams, excited about the idea of a tool that could be configured to let them see how their own system was working in a way that matched their own model. It was surprising that reflecting on system and network structure was of interest to more groups of users than anticipated. Customer support representatives felt that knowing system status in more detail could help them work with end-users; developers felt that monitoring the current system could help them understand the needs of Operations better.

5.5.3 Summary: OpsVis Evaluation

In summary, OpsVis helps to meet these challenges by enabling developers and operators to create visualizations that provide insight at a glance into anomalies and variability across the systems. It provides a perspective that matches the way cloud service developers and operators think about their systems. The above evaluation experience in applying the OpsVis prototype to monitoring cloud services within Microsoft have been quite positive. A wide internal deployment of this system has suggested the efficacy and utility of the visual analytics design.

5.6 Conclusion

This chapter describes the evaluations for the three designed visual analytics systems. To accommodate diverse deployment environments, these evaluations, including both formative and summative evaluations, were customized to assess the efficacy of the designed systems in individual organizational setting. All the three evaluations demonstrated the utility of the three visual analytics system in incorporating both the domain general analytical activities and the individual analysis processes. The results from these evaluations were further used to improve the visual analytics designs and implementations.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Objectives

This chapter concludes the usefulness of this dissertation. It presents the contributions of this dissertation, and further discusses the limitations and future directions for this research.

6.2 Overview

There are several parts of this dissertation. This dissertation, however, pursues a single goal. That is to create a two-stage framework for designing visual analytics systems in an organizational environment. This work has been an attempt to make a first general investigation of the problem of systematically designing a visual analytics to incorporate both general and individual analysis processes. The missing of a general visual analytics design framework was a recognized problem that had not been systematically addressed in a general way.

A review of the existing literature uncovered the fact that although many researchers had identified this topic as a critical problem, none had made comprehensive and general theoretical guidelines. Without such theoretical guidelines, the design of a visual analytics system would 1) lack of recommendations to instrument an effective system design and development; and 2) provide less tractable procedures for researchers to assess the visual analytics use patterns, and evaluate its impacts.

This dissertation therefore presented four years of iterative design efforts to explore, establish, and advance the design of visual analytics systems. This dissertation started by extending current practices pertaining to analytical workflow and focused, in particular, on investigating its dynamics to the design of visual analytics systems for organizational environments. Specifically, to achieve such framework, three extensive collaborations with organizations and groups of knowledge workers were conducted to gain insights about the general analytical tasks and workflows.

In particular, this dissertation presents a series of research processes: it began by categorizing the design experiences gained from collaborations with various organizations into a general organizational analysis workflow. Then, validated by domain users, this research encapsulated the general workflow into a two-stage design, and listed the necessary design considerations for each stage. It further followed these considerations and developed actual visual analytics system through iterative prototyping with domain users.

Through extensive empirical evaluations of the two design stages, this research finally encapsulated both stages into a general design framework, and outlined its four essential design recommendations. These four general design recommendations, when followed, empower such systems to bring the users closer to the center of their analytical processes. As shown in Table 23, these recommendations are presented as a natural progression for designing a visual analytic system. In addition, this dissertation presents the visual analytics designers with a checklist of design considerations that could be used to instruct the development of their visual analytic system. These considerations illustrate the necessary actions and recommendations to design a visual system that augments organizational analytics processes, and they are presented in Figure 47.

6.3 Contributions

Concluded based on these extensive collaborations, this dissertation proposes a two-stage design framework for designing visual analytics systems, as shown in Figure 47. The goal for this framework is to inform the design of a visual analytics system through disseminating and incorporating the general analytical workflows into the process. In particular, the first stage in this framework is an ***Observation and***

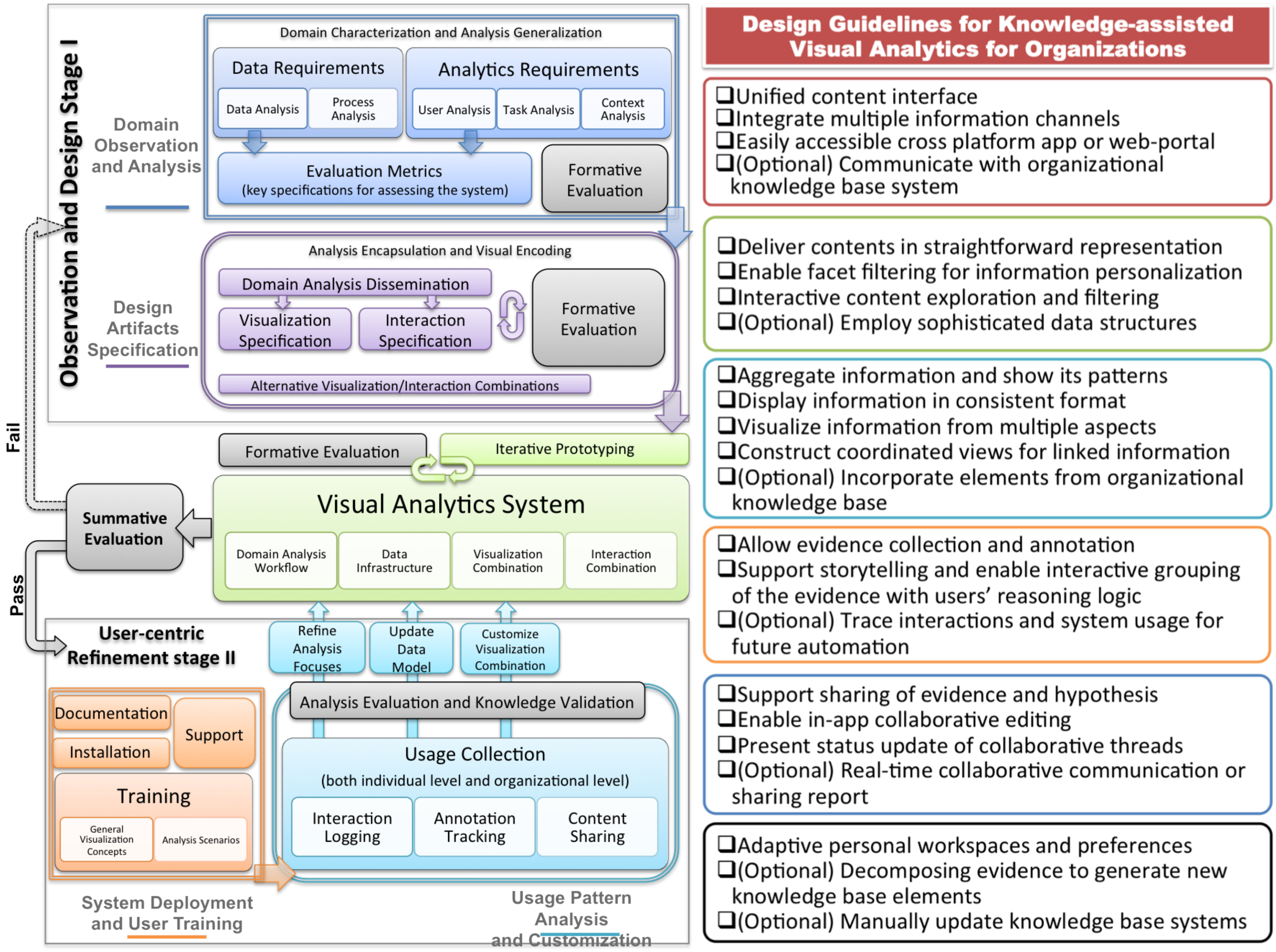
Table 23: The recommended recommendations for achieving the two-stage design framework.

Recommendation 1	Characterize Organizational Analytics Processes Through Interactions with Domain Users
Recommendation 2	Disseminate Analytics Workflows to Key Actionable Knowledge
Recommendation 3	Design for Actionable Knowledge Transformation Through Software Prototyping
Recommendation 4	Design for Integrating individual's Analysis Practices with General Analytical Workflow

Designing stage, in which a visual analytic system is designed and implemented to abstract and encapsulate general organizational analytical processes. The second stage is the ***User-centric Refinement*** stage, which aims at interactively enriching and refining the already encapsulated domain analysis process based on understanding user's intentions through analyzing their analysis processes.

The primary contributions of this dissertation are therefore threefold: first, this dissertation proposes a two-stage framework for facilitating the domain users' workflow through integrating their analytical models into interactive visual analytics systems. This design framework illustrates general design recommendations that, when followed, empowers a visual analytics system to bring the users closer to the center of their analytical processes. By integrating the analytical models into interactive visual analytics, the user directly interacts with the data in real time and makes analytical decisions in a customized reasoning environment. To illustrate the generalizability and effectiveness of the design recommendations, this dissertation further introduces and evaluates three visual analytics systems designed using them as a basis. All of these systems are deployed to domain knowledge workers and are adopted for their analytical practices. Extensive empirical evaluations are further conducted to demonstrate efficacy of these systems in facilitating domain analytical processes.

This framework emphasizes the better understanding of the pragmatic analytical



Design Guidelines for Knowledge-assisted Visual Analytics for Organizations

- ☐Unified content interface
- ☐Integrate multiple information channels
- ☐Easily accessible cross platform app or web-portal
- ☐(Optional) Communicate with organizational knowledge base system

- ☐Deliver contents in straightforward representation
- ☐Enable facet filtering for information personalization
- ☐Interactive content exploration and filtering
- ☐(Optional) Employ sophisticated data structures

- ☐Aggregate information and show its patterns
- ☐Display information in consistent format
- ☐Visualize information from multiple aspects
- ☐Construct coordinated views for linked information
- ☐(Optional) Incorporate elements from organizational knowledge base

- ☐Allow evidence collection and annotation
- ☐Support storytelling and enable interactive grouping of the evidence with users' reasoning logic
- ☐(Optional) Trace interactions and system usage for future automation

- ☐Support sharing of evidence and hypothesis
- ☐Enable in-app collaborative editing
- ☐Present status update of collaborative threads
- ☐(Optional) Real-time collaborative communication or sharing report

- ☐Adaptive personal workspaces and preferences
- ☐(Optional) Decomposing evidence to generate new knowledge base elements
- ☐(Optional) Manually update knowledge base systems

Figure 47: An overview for the two-stage framework and the detailed design considerations for developing a visual analytics system (Right).

processes in an organizational environment. It focuses on identifying practical design recommendations for visual analytics systems. To this end, this dissertation consolidated the design recommendations into characteristics for the six common analytical task activities, their related actionable knowledge, and interactions between the two. It found that actionable knowledge plays a unique role in addressing important problems in organizations, and affects knowledge workers' performance. Therefore, this work transformed this knowledge into design considerations for visual analytics systems. These considerations were intended to help others visual analytics designers provide better support for domain analytical processes within their visual analytics applications.

The detailed design considerations on incorporating individual's analytics processes were also presented in this dissertation. The considerations are used to achieve the *user-centric refinement* stage, and focuses on enriching and refining domain analysis through capturing and analyzing knowledge workers' analysis processes. Two possible techniques were discussed to achieve such goal, namely interaction capturing and annotation tracking. This work further demonstrated the utility of these techniques in understanding users' analytical preferences in order to customize their analysis processes. To exemplify the efficacy of these techniques, this research has applied them to the design of several interactive visual analytics systems. Empirical evaluations with domain analysts were conducted to demonstrate their efficacy in supporting customized analytical processes.

Secondly, this dissertation provides a general ground to bridge research and industry on design and development. It connects the academic research on visual analytics to industrial organizations, and showcases the utility of organizational visual analytics systems. It not only provides industrial collaborators concrete ideas about the impact that a visual analytics system can bring to them, but also suggests practical framework and considerations for designing visual analytics system.

This framework presents a general characterization of the analytical workflow in organizational environments. This characterization fills in the blank of the current lacking of such analytical model and further represents a set of domain analytical tasks that are commonly applicable to various organizations. Specifically, this work has identified six task activities essential for these professionals' decision-making workflows. In addition, by bridging the gap between high-level design concepts and fine-grain implementation of such concepts, this dissertation provides a pragmatic view of implementing an organizational visual analytics that can help augment organizational information analyses through modeling domain users reasoning approaches

Finally, this dissertation provides academia with more theoretical approach to understand and design visual analytics systems. It encourages researchers to search and establish the foundation of visual analytics design principles. This dissertation can also serve for educational purpose, and are intended to use as a course syllabus and materials for teaching visual analytics research.

6.4 Limitations

There are limitations to the research that must be addressed. Generalizability of these design considerations is limited because this research was conducted within only three organizations. This dissertation attempted to mitigate local biases by increasing the number of participants. Nevertheless, different training backgrounds, personal preferences, and project time constraints could engender different analytical conditions.

Moreover, the research characterizes the domain analytical workflow through interviews and surveys, which generally are self-reported by participants. This research was also limited, in that it modeled the analytical workflow from a retrospective perspective, whereas Brows et al. demonstrated that problem spaces and solutions are established and change dynamically in interactions with people and the environment [18]. Therefore, the understanding of domain analysis and actionable knowledge is constrained to the knowledge workers' general way of performing tasks.

Finally, this research is limited by its evaluations with domain experts. This dissertation evaluated Taste with formal studies, and IRSV and OpsVis with informal case studies. Developing evaluations, strategies, and methodologies to accurately assess the effectiveness of a visual analytics system is challenging. At this point this dissertation do not have a clear outline on the best evaluation approach; the design of recommendations for evaluating a visual analytic system would be one interesting future direction for the research.

Understanding each of these relationships is imperative for maintaining the validity and integrity of a knowledge base that is used in real decision-making environments. At present time, domain experts handle all three scenarios manually. However, in the Knowledge Engineering literature, researchers proposed and designed several verification and validation (V&V) techniques and tools. Some of them support the ability to automatically verify and validate underlying knowledge. But without a clear understanding of domain knowledge, most automatic techniques and tools are not always reliable. In visualization, it remains an open research area for us to create a semi-automated knowledge management system for organizing and storing diverse knowledge and rules in the same knowledge base.

However, while this dissertation recognize these limitations in our work, this work considers the support of organizational analysis processes is an important visual analytics research. The concluded design considerations illuminate the role that a visual analytics plays in such complex problem-solving environments.

6.5 Future Work

This dissertation contributes to the establishment of a two-stage design framework for visual analytics. But some components in this framework still require further solutions. The uncovered complexity of this framework implies that there is a vast amount of work that must be done before a final framework is complete. Some potentially fruitful additional work was considered during the creation of this dissertation, but which were out- side the scope of this work. These additional efforts are outlined here and left as future work.

There are three categories of future works: (1) expansion of interactive reasoning modeling capabilities for the framework; (2) understanding the cost of customization; and (3) establishing the evaluation foundation of visual analytics.

6.5.1 Expanding the interactive reasoning modeling capabilities

A first step in this work could be to expand the interactive reasoning modeling capabilities for the framework. The analysis of the identified relevant domains of research could be deepened, and other relevant domains may be discovered. This expansion of this interactive modeling process could be used to refine and improve the incorporation of individual's analytical processes in a visual analytics system.

This research is interested to consider whether externalizing such domain knowledge and reapplying it into customized visualizations would be feasible for enhancing domain decision-making process. Although there is no definitive way to achieve complete knowledge transfer, existing research that has demonstrated how to incorporate visualization with domain specific knowledge [175, 54]. To achieve similar knowledge externalization, a tight integration of the visualization with an ontological knowledge structure were proposed to interactively capture and store the user's interactions and translate them into domain knowledge [167]. This externalization could further be used in training new managers, communicating with others, and reporting decisions.

In addition, this work intends to investigate additional analysis methods for the automated analysis of user's interaction logs and annotations. For example, Hidden Markov Models (HMM) could be used for data where the segments are not explicitly defined but can be learned based on the original data sequence. These potential additions combined with the general approach of blending automated and multi-view, interactive visual analysis open the door to gain new insights that can help model the domain users' reasoning processes.

6.5.2 Understanding the cost analysis and customization validation

The validation process of the cost for customization is of great importance in the "feedback" loop for the proposed design framework. As described in Section 4.5.2.2.2, this process emphasizes the verification and validation of the customization requests that are generated based on analyzing users analytical behaviors. The key in this process is the identifications of the measures that can determine the cost of customizing a system. Currently, there is no existing research that addresses this recognized problem.

In the future, one of the most important research directions for this work is to continue investigating the measures for the costs of customization. This direction emphasizes the search for visual and/or interaction parameters that can be used to quantify such cost. Specifically, this research would focuses on creating a combined factor to attach costs to system customization. This factor could be calculated based on the cost of interaction (concluded by Lam [99]), cost of visualization (suggested by Amar et al. [5]), and cost of cognitive overload (proposed by Green et al. [61]).

A first step in this direction could be to surveying the existing literatures (e.g. visual analytics, InfoVis and HCI) for the theoretical foundation for the cost of customization. The analysis of the identified relevant domains of research could be deepened, and other relevant domains may be discovered. This expansion of the theoretical foundation could be used to refine and improve the Definition of the cost to interactively customize a visual analytics system.

6.5.3 Establishing the evaluation foundation of visual analytics

General evaluation recommendations for the assessment of the proposed visual analytics framework have not been solved here. Future work is needed to complete this research. The final solution may be some mixture of the utilization of internal evidence and external evidence [112]. Both these evidences present a coherent perspective to evaluate the framework, by placing it into the evolvement of the visual analytics field. The evidences are collected to support speculation that such a mixed solution may be more useful than any one solution in isolation.

On the one hand, these general evaluation recommendations should not only focuses on the assessment of the functionalities of a visual analytics system. It needs to verify the utility of a designed system, and validate how properly the implemented functions are in facilitating domain analysis process.

On the other hand, these recommendations should also emphasize measuring the knowledge-gain for the domain users. Much like the confidence value measured in evaluating Taste (presented in section 5.3), these evaluation recommendations need to place emphasis on measuring the impacts of visual analytics systems in affecting individual's domain analytical practices.

REFERENCES

[1] *Advanced tools for operators at Amazon.com* (Dublin, Ireland, 2006 2006).

[2] AASHTO. Transportation asset management guide. Tech. rep., NCHRP Project 20-24, January 2003.

[3] AG, S. Business management software.

[4] Agarwal, R., and Karahanna, E. Time flies when you are having fun: Cognitive absorption and beliefs about information technology usage. *MIS Quarterly* (2000).

[5] Amar, R. A., and Stasko, J. T. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics 11* (2005), 432–442.

[6] Andrews, K. Evaluating information visualisations. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (New York, NY, USA, 2006), BELIV '06, ACM, pp. 1–5.

[7] Andriole, S. Fast, cheap requirements prototype, or else! *Software, IEEE 11*, 2 (Mar. 1994), 85 –87.

[8] Annett, J., and Duncan, K. Task analysis and training design. *Occupational Psychology* (1967).

[9] Apple Inc. Apple mac os x tools. Tech. rep., Apple Inc, 2011.

[10] Argyris, C., and Schon, D. A. *Theory in Practice: Increasing Professional Effectiveness*. Jossey-Bass, 1992.

[11] Bederson, B. B., Shneiderman, B., and Wattenberg, M. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph. 21*, 4 (2002), 833–854.

[12] Bertin, J. Readings in information visualization. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, ch. Graphics and graphic information processing, pp. 62–65.

[13] Bertini, E., and Santucci, G. Visual quality metrics. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (New York, NY, USA, 2006), BELIV '06, ACM, pp. 1–5.

[14] Bevan, N. International standards for hci and usability. *Int. J. Hum.-Comput. Stud. 55* (October 2001), 533–552.

[15] Bhatt, G. D. Knowledge management in organizations: examining the interaction between technologies, techniques, and people. *Journal of Knowledge Management* (2001).

[16] Bier, E. A., Ishak, E. W., and Chi, E. Entity quick click: rapid text copying based on automatic entity extraction. In *CHI '06 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2006), CHI '06, ACM, pp. 562–567.

[17] Binns, J., DiCarlo, J., Insley, J. A., Leggett, T., Lueninghoener, C., Navarro, J.-P., and Papka, M. E. Enabling community access to teragrid visualization resources: Research articles. *Concurr. Comput. : Pract. Exper. 19* (April 2007), 783–794.

[18] Brown, J. S., and Duguid, P. Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. *Organization Science 2*, 1 (1991), 40–57.

[19] Bruls, M., Huizing, K., and Wijk, J. J. V. Squarified treemaps. In *In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization* (2000), Eindhoven University of Technology, Press, pp. 33–42.

[20] Bucher, T., Gericke, A., and Sigg, S. Process-centric business intelligence. *Business Process Management Journal 15*, 408-429 (2009).

[21] Burkhard, R. A. Towards a framework and a model for knowledge visualization: Synergies between information and knowledge visualization. *Knowledge and Information Visualization* (2005), 238–255.

[22] Butkiewicz, T., Chang, R., Wartell, Z., and Ribarsky, W. Visual analysis of urban change. *Computer Graphics Forum (IEEE Eurovis Conference) 27*, 3 (2008), 903–910.

[23] Card, S., Thomas, J. J., and Cook, K. A. *Illuminating the Path: The Research and Development Agenda for Visual Analytics. Chapter 2.* National Visualization and Analytics Ctr, 2005.

[24] Card, S. K. *Information visualization In Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.* Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2002.

[25] Carlile, P. R. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science 13*, 4 (2002), 442–455.

[26] Carroll, J. M., Thomas, J. C., and Malhotra, A. Presentation and representation in design problem-solving. *British Journal of Psychology 71*, 1 (1980), 143–154.

[27] Chang, R., Wessel, G., Kosara, R., Sauda, E., and Ribarsky, W. Legible cities: Focus-dependent multi-resolution visualization of urban relationships. In *Transactions on Visualization and Computer Graphics* (2007), vol. 13, IEEE Computer Society, pp. 1169–1175.

[28] Chau, D. H., Myers, B., and Faulring, A. What to do when search fails: finding information by association. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 999–1008.

[29] Chen, M., Ebert, D., Hagen, H., Laramee, R. S., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., and Silver, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl. 29*, 1 (2009), 12–19.

[30] Chen, M., Ebert, D., Hagen, H., Laramee, R. S., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., and Silver, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl. 29* (January 2009), 12–19.

[31] Chen, Y., Yang, J., and Barlowe, S. Click2annotate: Automated insight externalization with rich semantics. *IEEE Conference on Visual Analytics Science and Technology 2010* (2010).

[32] Chi, E. H.-h., and Riedl, J. An operator interaction framework for visualization systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization* (Washington, DC, USA, 1998), IEEE Computer Society, pp. 63–70.

[33] Chuah, M. C., and Roth, S. F. On the semantics of interactive visualizations. *Proceedings of Information Visualization, IEEE,* (1996).

[34] Cleveland, W. S., and McGill, R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association 79*, 387 (1984), 531–554.

[35] Coleman, J., Goettsch, A., Savchenko, A., Kollmann, H., Wang, K., Klement, E., and Bono, P. T. Towards collaborative volume visualization environments. *Computer and Graphics* (1996).

[36] Convertino, G., Kairam, S., Hong, L., Suh, B., and Chi, E. H. Designing a cross-channel information management tool for workers in enterprise task forces. In *Proceedings of the International Conference on Advanced Visual Interfaces* (New York, NY, USA, 2010), AVI '10, ACM, pp. 103–110.

[37] Corporation., I. D. Idc: worldwide research report. Tech. rep., International Data Corporation., 2009.

[38] Coughlan, P., and Coghlan, D. Action research for operations management. *International Journal of Operations and Production Management,* (2002).

[39] Cowell, A. J., Gregory, M. L., Marshall, E. J., and McGrath, L. Knowledge encapsulation framework for collaborative social modeling. In *In AAAI Spring Symposium on Technosocial Predictive Analytics* (2009).

[40] Cross, R., and Sproull, L. More than an answer: Information relationships for actionable knowledge. *Organization Science 15* (August 2004), 446–462.

[41] Davenport, T. H., De Long, D. W., and Beers, M. C. Successful Knowledge Management Projects. *MIT Sloan Management Review* (Jan. 1998), 43–57.

[42] Davenport, T. H., and Prusak, L. *Working Knowledge: How Organizations Manage What They Know.* Project Management Institute, Dec. 1997.

[43] Davies, D., and Thomson., G. *Memory in Context: Context in Memory.* Wiley, 1988.

[44] Davis, F. D., and Venkatesh, V. Toward preprototype user acceptance testing of new information systems: implications for software project management. *Engineering Management, IEEE Transactions on 51*, 1 (2004), 31–46.

[45] Davis, G. B., and Olson, M. H. *Management information systems: conceptual foundations, structure, and development (2nd ed.).* McGraw-Hill, Inc., New York, NY, USA, 1985.

[46] Demetrios E. Tonias, J. J. Z. *Bridge engineering.* McGraw Hill, 2006.

[47] Diaper, D. *Task Analysis for Human-Computer Interaction.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1990.

[48] Dillon, A., and Watson, C. User analysis in hci–the historical lessons from individual differences research. *Int. J. Hum.-Comput. Stud. 45* (December 1996), 619–637.

[49] Dou, W., Jeong, D. H., Stukes, F., Ribarsky, W., Lipford, H. R., and Chang, R. Recovering reasoning processes from user interactions. *IEEE Comput. Graph. Appl. 29*, 3 (2009), 52–61.

[50] Dou, W., Ziemkiewicz, C., Harrison, L., Jeong, D. H., Ryan, R., Ribarsky, W., Wang, X., and Chang, R. Comparing different levels of interaction constraints for deriving visual problem isomorphs. *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on* (2010).

[51] Fisher, D. Hotmap: Looking at geographic attention. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (Nov.-Dec. 2007), 1184–1191.

[52] Fuchs, L., and Fuchs, D. Effects of systematic formative evaluation: a meta-analysis. *Exceptional Children, v53 n3 p199-208 Nov 1986* (1986).

[53] Garg, S., Nam, J., Ramakrishnan, I., and Mueller, K. Model-driven visual analytics. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (Oct. 2008), pp. 19–26.

[54] Garg, S., Nam, J., Ramakrishnan, I., and Mueller, K. Model-driven visual analytics. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (Oct. 2008), pp. 19–26.

[55] Geczy, P., Izumi, N., Akaho, S., and Hasida, K. Analytics and management of collaborative intranets. In *Collaborative Computing: Networking, Applications and Worksharing*. Springer Berlin Heidelberg, 2009, pp. 623–631.

[56] Gerlach, J., and Kuo, F.-Y. Understanding human computer interaction for information systems design. *MIS Quarterly* (2008).

[57] Google. Google desktop. Tech. rep., Google, 2011.

[58] Gotz, D., and Wen, Z. Behavior-driven visualization recommendation. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces* (New York, NY, USA, 2009), ACM, pp. 315–324.

[59] Gotz, D., and Zhou, M. X. Characterizing users' visual analytic activity for insight provenance. *Information Visualization 8*, 1 (2009), 42–55.

[60] Gray, D. E., and Black, T. R. Prototyping of computer-based training materials. *Computers and Education 22*, 3 (1994), 251 – 256.

[61] Green, T., Ribarsky, W., and Fisher, B. Visual analytics for complex concepts using a human cognition model. *IEEE Visual Analytics Science and Technology* (2008).

[62] Groth, D. P., and Streefkerk, K. Provenance and annotation for visual exploration systems. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1500–1510.

[63] Hackos, J. T., and Redish, J. C. *User and task analysis for interface design.* John Wiley & Sons, Inc., New York, NY, USA, 1998.

[64] Hammer, M., and Champy, J. *Reengineering the Corporation: A Manifesto for Business Revolution.* HarperBusiness, Apr. 1994.

[65] Hanrahan, P. Systems of thought. Eurovis 2009 Keynote Address, 2009.

[66] Hawk, H., and Small, E. P. The bridgit bridge management system. *Structural Engineering International Volume 8* (1998), 309–314.

[67] Hayes, J. R., and Simon, H. A. Psychological differences among problem isomorphs. In *Cognitive Theory*, J. Castellan, D. B. Pisoni, and G. Potts, Eds., vol. 2. Erlbaum, 1977.

[68] Heer, J., Ham, F., Carpendale, S., Weaver, C., and Isenberg, P. Information visualization. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Creation and Collaboration: Engaging New Audiences for Information Visualization, pp. 92–133.

[69] Heer, J., Kong, N., and Agrawala, M. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 1303–1312.

[70] Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1189 – 1196.

[71] Heuer, R. *Psychology of Intelligence Analysis*. Pherson Associates, 2007.

[72] Hevner, A. R., March, S. T., and Park, J. Ram, S. Design science in information systems research. *MANAGEMENT INFORMATION SYSTEMS QUARTERLY* (2004).

[73] Hirschheim, R., and Klein, H. K. Four paradigms of information systems development. *Commun. ACM 32* (October 1989), 1199–1216.

[74] Hoffman, R. R., Ed. *The psychology of expertise: cognitive research and empirical AI*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.

[75] Hord, S. M. A synthesis of research on organizational collaboration. *Educational Leadership, v43 n5 p22-26 Feb 1986* (1986).

[76] Hundhausen, C., Douglas, S., and Stasko, J. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing 13*, 3 (2002), 259–290.

[77] HUNDHAUSEN, C. D., DOUGLAS, S. A., and STASKO, J. T. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing 13*, 3 (2002), 259 – 290.

[78] Hutchins, E. Material anchors for conceptual blends. *Journal of Pragmatics 37*, 10 (2005), 1555 – 1577. Conceptual Blending Theory.

[79] IBM. Business analytics and optimization.

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

[81] Janssen, W. C. The uplib personal digital library system. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries* (New York, NY, USA, 2005), JCDL '05, ACM, pp. 410–410.

[82] Jeong, D. H., Dou, W., Lipford, H., Stukes, F., Chang, R., and Ribarsky, W. Evaluating the relationship between user interaction and financial visual analysis. *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (Oct. 2008), 83–90.

[83] Johnson, P., Johnson, H., Waddington, R., and Shouls, A. Task-related knowledge structures: analysis, modelling and application. In *Proceedings of the Fourth Conference of the British Computer Society on People and computers IV* (New York, NY, USA, 1988), Cambridge University Press, pp. 35–62.

[84] Jonassen, D., and Reeves, T. *Learning with technology: Using computers as cognitive tools.* Handbook of research in educational communications and technology (pp. 693-719), 1996.

[85] Jones, W., Dumais, S., and Bruce, H. Once found, what then? A study of keeping behaviors in the personal use of Web information. *Proceedings of the American Society for Information Science and Technology 39*, 1 (Jan. 2005), 391–402.

[86] Kadivar, N., Chen, V., Dunsmuir, D. abd Lee, E., Qian, C., Dill, J., Shaw, C., and Woodbury, R. Capturing and supporting the analysis process. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (2009).

[87] Keefe, D., Ewert, M., Ribarsky, W., and Chang, R. Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1383–1390.

[88] Kindlmann, G. Lack of reproducibility hinders visualization science. *IEEE Visualization Compendium, IEEE CS Press* (2006).

[89] Kirsh, D. The intelligent use of space. *Artificial Intelligence 73*, 1-2 (1995), 31–68.

[90] Kirsh, D. Interaction, external representation and sense making. In *In Proceedings of the 31st Annual Conference of the Cognitive Science Society* (2009), pp. 1103–1108.

[91] Kirsh, D. Problem solving and situated cognition. In *The Cambridge Handbook of Situated Cognition* (Cambridge, 2009), Cambridge University Press, pp. 264–306.

[92] Kirsh, D. Projection, problem space and anchoring. In *In Proceedings of the 31st Annual Conference of the Cognitive Science Society* (2009), pp. 2310–2315.

[93] Kirsh, D., and Maglio, P. On distinguishing epistemic from pragmatic action. *Cognitive Science 18* (1994), 513–549.

[94] Kirsh, D., and Maglio, P. On distinguishing epistemic from pragmatic actions. *Cognitive Science 18*, 4 (1994), 513–549.

[95] Klein, G. *Sources of Power: How People Make Decisions*. The MIT Press, February 1999.

[96] Klein, H. K., and Myers, M. D. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Q. 23* (March 1999), 67–93.

[97] Koppen, M. Extracting human expertise for constructing knowledge spaces: an algorithm. *J. Math. Psychol. 37*, 1 (1993), 1–20.

[98] Kotovsky, K., and Kushmerick, N. Processing constraints and problem difficulty: A model. In *Proc. Conference of the Cognitive Science Society* (1991), K. J. Hammond and D. Gentner, Eds.

[99] Lam, H. A framework of interaction costs in information visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1149–1156.

[100] Langley, P. Lessons for the computational discovery of scientific knowledge. In *In the Proceedings of the First International Workshop on Data Mining Lessons Learned* (2002).

[101] Larkin, J. H., and Simon, H. A. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science 11* (1987), 65–99.

[102] Lee, B., Smith, G., Robertson, G. G., Czerwinski, M., and Tan, D. S. Facetlens: exposing trends and relationships to support sensemaking within faceted datasets. In *Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 1293–1302.

[103] Lee, S., and Gandhi, R. Ontology-based active requirements engineering framework. In *In: Proceedings of the 12th Asia-Pacific Soft. Engg Conference 2005, IEEE CS, p.481-490.* (2005).

[104] Liu, Z., Nersessian, N. J., and Stasko, J. T. Distributed cognition as a theoretical framework for information visualization. *IEEE TVCG 14*, 6 (2008), 1173–1180.

[105] Ma, K.-L. Creating a collaborative space to share data, visualization, and knowledge. *SIGGRAPH Comput. Graph. 41* (November 2007), 4:1–4:4.

[106] Majid, R. A., Noor, N. L. M., Adnan, W. A. W., and Mansor, S. A survey on hci considerations in the software development life cycle: from practitioner's perspective. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (New York, NY, USA, 2009), ICIS '09, ACM, pp. 21–24.

[107] ManyEyes. Manyeyes online. Tech. rep., IBM, 2011.

[108] Markus, M. L., and Keil, M. If We Build It, They Will Come: Designing Information Systems That People Want to Use. *Sloan Management Review 35*, 4 (1994).

[109] Markus, M. L., Majchrzak, A., and Gasser, L. A design theory for systems that support emergent knowledge processes. *MIS Quarterly 26*, 3 (2002), pp. 179–212.

[110] Mattessich, P. W., and Monsey, B. R. *Collaboration: What Makes It Work. A Review of Research Literature on Factors Influencing Successful Collaboration.* Amherst H. Wilder Foundation, 919 Lafond, St. Paul, MN 55104., 1992.

[111] McCartney, L. Getting smart about knowledge management. *Industry Week* (4 May 1998).

[112] McDonald, D. D. *Internal and external evidence in the identification and semantic categorization of proper names.* MIT Press, Cambridge, MA, USA, 1996, pp. 21–39.

[113] Microsoft Corporation. Mircrosoft virtual earth. Tech. rep., Microsoft Corporation, 2009.

[114] Moller, T. A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 71–80.

[115] Munzner, T. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 921–928.

[116] Nielsen, J. *Usability Engineering.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[117] Nissen, M. E. Redesigning reengineering through measurement-driven inference. *MIS Quarterly* (1998).

[118] Nissen, M. E., and Espino, J. Knowledge process and system design for the coast guard. *Knowledge and Process Management 7* (2000), 165–176.

[119] Nonaka, I., and Takeuchi, H. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, USA, May 1995.

[120] North, C., and Shneiderman, B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. 128–135.

[121] Nunamaker, Jr., J. F., Briggs, R. O., Mittleman, D. D., Vogel, D. R., and Balthazard, P. A. Lessons from a dozen years of group support systems research: a discussion of lab and field findings. *J. Manage. Inf. Syst. 13* (December 1996), 163–207.

[122] Overmyer, S. Revolutionary vs. evolutionary rapid prototyping: balancing software productivity and hci design concerns. *In: Proceedings of the 4th International Conference on Human-Computer Interaction* (1991).

[123] Pacific Northwest National Laboratory. In-spire. Tech. rep., Pacific Northwest National Laboratory, 2011.

[124] Palmer, S. E. Fundamental aspects of cognitive representation. In *Cognition and Categorization*, E. Rosch and B. B. Lloyd, Eds. Lawrence Erlbaum Associates, 1978, pp. 262–304.

[125] Pasmore, W., and Friedlander, F. An action-research program for increasing employee involvement in problem solving. *Administrative Science Quarterly, v27 n3 p343-62 Sep 1982* (1982).

[126] Pava, C. H. *Managing new office technology : an organizational strategy.* Collier Macmillan, 1983.

[127] 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. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (oct. 2008), pp. 131 –138.

[128] 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.

[129] PIKE, W. A., Stasko, J., Chang, R., and O'Connell, T. A. The science of interaction. *Information Visualization 8* (2009), 263–274.

[130] PINELLE, D., AND GUTWIN, C. Groupware walkthrough: adding context to groupware usability evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* (New York, NY, USA, 2002), CHI '02, ACM, pp. 455–462.

[131] PIROLLI, P., AND CARD, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proc. Int'l Conf. Intelligence Analysis* (2005).

[132] PRIES-HEJE, J., BASKERVILLE, R., AND HANSEN, G. I. Strategy models for enabling offshore outsourcing: Russian short-cycle-time software development. *Inf. Technol. Dev. 11* (January 2005), 5–30.

[133] ROBERTS, J. C. State of the art: Coordinated and multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization* (2007), pp. 61–71.

[134] ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B., AND STASKO, J. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1325–1332.

[135] ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B., AND STASKO, J. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14* (November 2008), 1325–1332.

[136] ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B., AND STASKO, J. Effectiveness of animation in trend visualization. *IEEE TVCG 14*, 6 (Nov.-Dec. 2008), 1325–1332.

[137] SCHOLTZ, J. Beyond usability: Evaluation aspects of visual analytic environment. *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology (VAST* (2006).

[138] SCHOLTZ, J. Developing qualitative metrics for visual analytic environments. *The Proceedings of BELIV '10* (2010).

[139] SCHREIBER, G. T., AND AKKERMANS, H. *Knowledge engineering and management: the CommonKADS methodology.* MIT Press, Cambridge, MA, USA, 2000.

[140] SCOTT, S. D., SHEELAGH, M., CARPENDALE, T., AND INKPEN, K. M. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (New York, NY, USA, 2004), CSCW '04, ACM, pp. 294–303.

[141] SCRIVEN, M. *Evaluation thesaurus.* Sage Publications., 1991.

[142] SHANI, A. B. W. A. P. Organisational enquiry: Towards a new model of the action research process. *Contemporary Organisation development* (1985).

[143] Shneiderman, B. Codex, memex, genex: the pursuit of transformational technologies. In *CHI 98 conference summary on Human factors in computing systems* (New York, NY, USA, 1998), CHI '98, ACM, pp. 98–99.

[144] Shrinivasan, Y. B., and Gotz, D. Connecting the dots with related notes. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 3649–3654.

[145] Shrinivasan, Y. B., and van Wijk, J. J. Supporting the analytical reasoning process in information visualization. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 1237–1246.

[146] Silva, C. T., Freire, J., and Callahan, S. P. Provenance for visualizations: Reproducibility and beyond. *Computing in Science and Engg. 9*, 5 (2007), 82–89.

[147] Simon, H. A. *The Sciences of the Artificial.* The MIT Press, 1996.

[148] Stenning, K., and Oberlander, J. A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science 19*, 1 (1995), 97–140.

[149] Strobelt, H., Oelke, D., Rohrdantz, C., Stoffel, A., Keim, D. A., and Deussen, O. Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15* (November 2009), 1145–1152.

[150] Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S., and Stolze, M. User-centered design and business process modeling: cross road in rapid prototyping tools. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction* (Berlin, Heidelberg, 2007), INTERACT'07, Springer-Verlag, pp. 165–178.

[151] Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2004), CHI '04, ACM, pp. 415–422.

[152] Tory, M., and Möller, T. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics 10* (January 2004), 72–84.

[153] Tory, M., and Moller, T. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications 25* (2005), 8–11.

[154] Tory, M., and Staub-French, S. Qualitative analysis of visualization: a building design field study. In *Proceedings of the 2008 conference on BEyond time and errors: novel evaLuation methods for Information Visualization* (New York, NY, USA, 2008), BELIV '08, ACM, pp. 7:1–7:8.

[155] Trainer, E., Quirk, S., de Souza, C., and Redmiles, D. Analyzing a socio-technical visualization tool using usability inspection methods. In *Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing* (Washington, DC, USA, 2008), VLHCC '08, IEEE Computer Society, pp. 78–81.

[156] Tufte, E. R. *Envisioning Information.* Graphics Press, May 1990.

[157] Tukey, J. W. *Exploratory Data Analysis*, 1 ed. Addison Wesley, Jan. 1977.

[158] Tweedie, L. Characterizing interactive externalizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1997), CHI '97, ACM, pp. 375–382.

[159] Tweedie, L. Characterizing interactive externalizations. In *Proceedings CHI* (1997), ACM Press, pp. 375–382.

[160] Valacich, J. S., and Hoffer, J. A. *Essentials of Systems Analysis and Design*, 1st ed. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[161] van Wijk, J. The value of visualization. *Visualization, 2005. VIS 05. IEEE* (Oct. 2005), 79–86.

[162] vanderAalst, W., and vanHee, K. *Workflow Management: Models, Methods, and Systems.* MIT Press, Cambridge, MA, USA, 2004.

[163] W Robert, A Marshall, R. S. Pontis bridge management system: state of the practice in implementation and development. In *9th International Bridge Management Conference* (Jan 01, 2003).

[164] Wang, X., Chen, S.-E., Hauser, E., and Ribarsky, W. imonitor: Architecture of web-based collaborative visual analytics system for bridge management. In *Transportation Research Board 90 Annual Meeting* (2010).

[165] Wang, X., Dou, W., Chen, S.-E., Ribarsky, W., and Chang, R. An interactive visual analytics system for bridge management. *Computer Graphics Forum 29* (2010), 1033–1042.

[166] Wang, X., Janssen, B., and Bier, E. Finding business information by visualizing enterprise document activity. In *Proceedings of the International Conference on Advanced Visual Interfaces* (New York, NY, USA, 2010), AVI '10, ACM, pp. 41–48.

[167] Wang, X., Jeong, D. H., Dou, W., Lee, S.-W., Ribarsky, W., and Chang, R. Defining and applying knowledge conversion processes to a visual analytics system. *Computers and Graphics 33*, 5 (2009), 616 – 623.

[168] Wang, X., Miller, E., Smarick, K., Ribarsky, W., and Chang, R. Investigative visual analysis of global terrorism. In *Computer Graphics Forum* (2008), Computer Graphics Forum, pp. 919–926.

[169] Webster, J., Trevino, L., and Ryan, L. The dimensionality and correlates of flow in human-computer interactions. *Compters in Human Behavior* (1993).

[170] Weick, K. E. *The Social Psychology of Organizing (Topics in Social Psychology Series)*. McGraw-Hill Humanities/Social Sciences/Languages, Jan. 1979.

[171] Wood, D. J., and Gray, B. Toward a Comprehensive Theory of Collaboration. *Journal of Applied Behavioral Science 27*, 2 (June 1991), 139–162.

[172] Wood, J., Dykes, J., Slingsby, A., and Clarke, K. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics 13* (2007), 1176–1183.

[173] 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* (New York, NY, USA, 2006), ACM, pp. 801–810.

[174] Xiao, L., Gerth, J., and Hanrahan, P. Enhancing visual analysis of network traffic using a knowledge representation. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On* (31 2006-Nov. 2 2006), pp. 107–114.

[175] Xiao, L., Gerth, J., and Hanrahan, P. Enhancing visual analysis of network traffic using a knowledge representation. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (31 2006-Nov. 2 2006), pp. 107–114.

[176] Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1224–1231.

[177] Yi, J. S., Kang, Y. a., Stasko, J., and Jacko, J. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1224–1231.

[178] Zhang, P., Carey, J., Te'eni, D., and Tremaine, M. Integrating human-computer interaction development into the systems development life

cycle: a methodology. *Communications of the Association for Information Systems (Volume 15, 2005)* (2005).

[179] Zimmerman, J., Forlizzi, J., and Evenson, S. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 493–502.

[180] Zuk, T., Schlesier, L., Neumann, P., Hancock, M. S., and Carpendale, S. Heuristics for information visualization evaluation. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (New York, NY, USA, 2006), BELIV '06, ACM, pp. 1–6.

Appendix A: Experiment Setup for Connecting Interaction Logging and Information Analysis Process

1 Introduction

In his insightful keynote address to the EuroVis 2009 conference, Pat Hanrahan discussed how visualization plays a role as a system of thought [65]. One remarkable topic of his talk is the use of visual problem isomorphs to make complex problem-solving seem simple and often trivial. The examples Dr. Hanrahan provided demonstrated the idea that once the right visual representation of the problem isomorph is found, solving that problem can be as simple as looking at the visual representation and identifying the right answer immediately. Of particular interest to us is the example of using a "magic square" as a visual isomorph to the Number Scrabble game (which is discussed further in section 2.1). It is clear that by transforming the Number Scrabble game into a magic square, this relatively difficult game of finding and adding multiple numbers becomes as simple as playing a game of tic-tac-toe.

This example is compelling because the process of encouraging a user to discover a useful visual isomorph for a problem can be thought of as the primary goal of visualization. However, the obvious question is, how does someone find the right visual isomorph to a problem? Unfortunately, the answer to this question is not trivial or well-understood. Building on work in cognitive science and diagrammatic reasoning [90], we argue that helping a user find a useful visual isomorph is not just a matter of presenting an appropriate visual representation. Rather, people can best discover visual solutions to problems through interaction with visual representations.

Unfortunately, while visualization researchers understand how to design visualizations to represent data, they have not exploited the relationship between interaction and problem solving to the same extent as cognitive scientists.

The goal of the research presented in this paper is therefore to bridge the gap between the findings in the cognitive science community and the visualization community. Specifically, we acknowledge research in the cognitive science community that shows interaction plays a critical role in problem solving [90]. However, given our emphasis on visualization, we do not simply seek to corroborate their existing findings. Instead, our interest lies at the intersection of the two fields where we look to understand how using interaction to solve problems can lead to the identification of potential visual isomorphs.

First we extend the notion that interaction generally facilitates problem solving. However, we further hypothesize that interactions with different constraints and amount of encoded information will lead to different solutions to the problem. In addition, we hypothesize that during the problem-solving process, the different constraints on interaction will lead to different types of isomorphs both visual and non-visual. Finally, we hypothesize that these different types of isomorphs have varying degrees of effectiveness in solving the problem, which can be measured quantitatively.

To test our hypotheses, we conducted a user study in which 117 participants were given different types of interaction constraints while developing strategies for the Number Scrabble game. We chose to use the Number Scrabble problem because it is self-contained and is known to have an optimal visual problem isomorph in the form of the magic square [147]. The participants' accuracy and time in playing the game against a computer were logged and tracked, and their strategizing session video-recorded. Based on the data obtained from the study, we find that: (1) different constraints on interactions do affect the participants' performance while playing the game, (2) with more constraints, the participant has a higher chance to derive the optimal visual isomorph (the magic square), and finally, while not all participants were able to derive the optimal visual isomorph, (3) using visual isomorphs in general leads to better performance than using non-visual isomorphs.

We begin by reviewing related work on interaction and problem isomorphs in the context of problem solving. Next, we present our experiment exploring the effect of interaction constraints on deriving visual problem isomorphs. We then discuss the implications of our experimental results and limitations of the study.

2 Related Work

Our conception of visualization as providing externalizations for problem solving draws on work in visualization theory as well as cognitive science. In particular, we study how visual representations can provide useful isomorphs of the information they visualize. Two problems or representations are isomorphic if they are informationally equivalent but present that information in different structures. As an example, we use the Number Scrabble problem and its isomorphic magic square representation.

2.1 The Number Scrabble problem

The original Number Scrabble [147] is a game played by two people with nine cards: ace through nine. The cards are placed in a row, face up. The players draw alternately, one at a time, selecting any one of the unselected cards. The objective of the game is for a player to get three cards which add up to 15 before his opponent does. If all nine cards have been drawn without either player having a combination that adds up to 15, the game is a draw.

The main reason we chose to use the Number Scrabble game is that there is a known visual isomorph of the problem called the "magic square" (figure 48). Since the magic square visually represents all possible combinations of three numbers that can be added up to 15 in a succinct manner, it can significantly help a player to perform well at the game. In other words, once this visual isomorph is identified, the Number Scrabble problem is turned into a much simpler tic-tac-toe game. THe number scrabble game represents a large number of well-defined problems that show how visual isomorphs can make evident what was previously true but obscure [147].

2	7	6
9	5	1
4	3	8

Figure 48: 3x3 magic square

2.2 Isomorphs and diagrammatic reasoning

Simon defined problem isomorphs as problems whose solutions and moves can be placed in one-to-one relation with the solutions and moves of the given problem [147]. The key to isomorphism is that even when two representations contain the same information, they can still provide very different sets of operations for accessing and inferring about that information, which can make a given problem easier or harder to solve [101]. In our example, the magic square and number scrabble are isomorphs of the same problem in that they both contain all the information needed to play the game. However, in number scrabble, the operations provided to the player to access important information about the game—such as whether your cards contain a winning combination—are mathematical. In the magic square case, that information is contained in a visual operation: seeing whether the cards form a line across the magic square grid. Since the brain processes such visual operations faster than mathematical ones, the visual isomorph is more efficient in this case.

The idea that visual representations make certain operations more efficient to perform is at the core of the theory of diagrammatic reasoning [34, 101]. However, efficiency is not the only measure of interest in visualization; our goal is to make information not just accessible, but understandable. The distinction between these goals is highlighted by Carroll et al. [26], who had participants solve a design problem presented as one of two isomorphs: a spatial arrangement problem and a temporal scheduling problem. The spatial isomorph was easier and faster for participants to solve and led to fewer failures to understand the problem. That is, in the temporal case there were several participants whose solutions did not follow the requirements of the task. Interestingly, when participants in both cases were provided a simple graphical representation (a grid) in which to work on their solution, the temporal case was as easy to solve as the spatial one, but participants in the temporal case remained more likely to fail to understand the problem requirements. The authors took this to mean that appropriate graphical representations can make problems easier to solve, but not necessarily easier to understand.

Another way to interpret this is that there is more to designing a visual isomorph than making information more efficient to access. Much of the power of visual representations comes from how they set constraints on interpretation and reasoning. Constraints inherent in visual isomorphs can encode constraints on the information they represent, leading to a more direct preservation of information structure [124]. As Stenning and Oberlander [148] argue, these constraints inherent to visual representations help to meaningfully restrict the number and kinds of inferences that can be made about a problem, focusing processing power on only valid cases. In this way, visual isomorphs can not only make operations more efficient, but can also model the constraints of a problem directly. This can affect the difficulty of solving a problem by reducing the cognitive load of remembering rules [98] or by encouraging different types of strategies [67].

2.3 Interaction and problem solving

While visual representations can aid problem solving significantly on their own, they gain even more power to model a problem when interaction is introduced. Interaction is increasingly seen as central to the process of reasoning with visualization [159, 104]. Lending weight to the intuition that interaction improves reasoning, Hundhausen et. al [76] found that interacting with an algorithm visualization produces better understanding than viewing an equivalent animation.

We use the term "interaction" in the broad sense defined by Yi et al.: "the dialogue between the user and the system as the user explores the data set to uncover insights" [176]. In this sense, the relationship between interaction and problem solving has been the subject of much research by cognitive scientists in the field of distributed cognition [78]. In particular, David Kirsh has extensively argued that projection and interaction with external representations are fundamental to human reasoning [89, 91, 90, 92, 94]. Kirsh points to the pervasive use of external representations and interaction with the world in everyday problem solving, and identifies several functions performed by interaction in the reasoning process [90]. Of these, most relevant to our work is *reformulation*, or the ability to restate ideas. Kirsh sees reformulation as a process that is frequently too complex to perform entirely in memory, and so is often managed with external tools. Since reformulation is closely related to identifying different problem isomorphs, we argue that this process can also be made easier through certain types of interaction.

3 Experiment

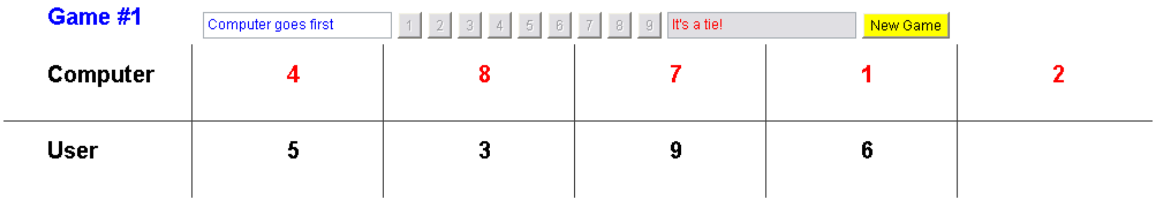
Our research objective is to investigate the question of how constraints on interaction affect problem solving through the derivation of visual isomorphs. We propose that in developing a strategy for playing a game like Number Scrabble, participants will tend to derive an isomorph for the problem that is easier for them to use than the representation in the original game, and that the availability of different levels of interaction while strategizing will lead to different types of isomorphs. If this is the case, it can help to clarify the relationship between interaction with visual representations and reasoning. To what extent does the nature of a visual representation, and the type of interactions a user is allowed to perform upon it, affect the kind of strategy that user develops for solving a problem?

We therefore designed a study based on the aforementioned Number Scrabble game due to its known optimal visual isomorph, the magic square. In our study, we developed 5 different interaction conditions, ranging from free-form to very restrictive, and studied how strategizing under these conditions affects problem solving and the development of isomorphs. In particular, we propose three interrelated hypotheses concerning interaction, problem solving, and isomorphs:

1. **Interactions and Problem Solving**: We hypothesize that different types of interactions will affect the participants' performance in playing the Number Scrabble game. Specifically, we hypothesize that more constrained interactions can encode more information, and will therefore lead to better problem-solving.
2. **Interactions and Isomorphs**: We hypothesize that the different constraints on interaction will affect the isomorphs generated by the participants. With higher constraints on interaction, a participant will be more likely to derive the optimal visual isomorph (the magic square).
3. **Isomorphs and Problem Solving**: We hypothesize that not all isomorphs developed by participants will be visual, but that visual isomorphs will be more effective for playing the Number Scrabble game.

4 Experiment Design

The main factor of interaction constraint had five levels (no interaction, pen and paper, single set of cards, multiple sets of cards, and boundary). Details of each constraint and design rationale will be discussed in section 4.3. We used a between-subjects design with repeated measures. Each subject is randomly assigned to one of the five interaction constraint conditions which determines what interactions are available to them during their strategy session. Qualitative measures in our experiment are the types of isomorphs our subject derived during their strategy session. Quantitative measures involved response time and scores on Number Scrabble games played against a computer, using the game interface shown in Figure 49. The computer was programmed to play the game optimally so that it never loses. While our subjects played the game against the computer, we recorded number of games tied or lost and the time it took them to figure out the next move for response time. We alternate who makes the first move between the subjects and the computer for every game played.



Game #1

Computer goes first

It's a tie!

Computer	4	8	7	1	2
User	5	3	9	6	

Figure 49: Number scrabble game interface

4.1 Participants

We recruited a total number of 117 participants (86 Male, 31 Female) from introduction to computer science courses at our university. Participants' age ranged from 18 to 40 with median of 25. Students were primarily undergraduates, and 80% were in computing-related majors.

4.2 Task

The experiment begins with investigators introducing the Number Scrabble game to the subjects based on a training script. The investigators were asked to play the game with the participants until they fully grasped the rules. Next, the participants fill out a demographic form on age, gender and experience with mathematical courses through a web interface. The rest of the experiment can be divided into four major sessions: pre-test, strategizing, externalizing isomorph, and post-test.

1. **Pre-test**: During the pre-test session, the participants were asked to play the Number Scrabble game six times against the computer. To make sure that our participants do not start developing strategies during the pre-test, we enforced a maximum time limit of 18 minutes to finish all six pre-test games. Failing to meet the time limit resulted in a participant's data being dropped from analysis.

2. **Strategizing**: During the strategizing session, the subjects were given 20 minutes and allowed to interact with the materials we provided under different constraints and are told to look for a strategy that can help them play the game better.

3. **Externalizing isomorphs**: At the end of the strategizing session, all participants were given 2-3 minutes to make a "cheat sheet" out of the strategy they developed so that they can refer to it during the post-test session when they play Number Scrabble again. This cheat sheet was a single sheet of paper onto which participants were told they could write anything they felt would help them play the game.

(In the case of the pen and paper condition, this was a separate sheet from those they wrote on during the strategizing session.) This gave us a record of the isomorph used by participants in forming a strategy and reduced the cognitive load on participants during the post-test. We only gave them a very short amount of time to make their "cheat sheet" so that they could not continue elaborating on it after the end of the strategizing session.

4. **Post-test**: During the post-test session, participants were asked to play the Number Scrabble game six more times against the computer while consulting their "cheat sheet." To be consistent with the pre-test and also to make sure that the participants do not refine their isomorphs during the post-test, 18 minutes was set as the upper limit for playing all six games. As in the pre-test, failing to meet the time limit resulted in a participant's data being dropped from analysis.

After the post-test session, participants were asked to fill out a questionnaire regarding how they arrived at their strategy and their experience during the strategizing session. The investigators collected all the participants' "cheat sheets" for further analysis of the isomorphs they derived during the experiment. In addition, the strategizing sessions were video recorded, which allows us to examine how the interaction constraints affected our participants' behavior during the process of searching for an isomorph.

4.3 Interaction constraints

We went through multiple rounds of a refining process to design the interaction constraint conditions used in our study. Our goal was to design constraints that ranged from placing no limit on the interaction to restricting the interaction a great deal.

- Constraint #1 (no interaction): The participants were asked to think about the problem in their head during the strategizing session to develop a strategy to help them play the game better. The participants were not allowed to interact with any materials.
- Constraint #2 (pen and paper): The participants were provided with pen and paper to work out their strategy for the Number Scrabble problem.
- Constraint #3 (multiple sets of cards): The participants assigned to this constraint were provided with multiple sets of cards, with each set consisting of the numbers one through nine. Each card is square in shape and made from paper with the numbers printed on them. Within the strategizing session, the participants were encouraged to organize the cards freely.
- Constraint #4 (single set of cards): The participants were further limited to interact with only one set of cards labeled with the numbers one through nine.
- Constraint #5 (boundary): This is the most restrictive case. Participants were presented with nine cards and a square space only large enough to fit the cards

in a grid, and were told to confine their interactions to that space. Figure 50 shows this condition.

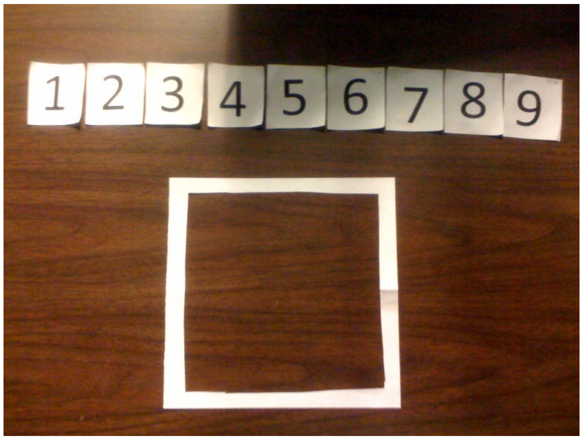


Figure 50: Cards and Boundary

Our conditions are designed so that "no interaction" serves as a control group, and "pen and paper" represents no limit on user interaction. Then, based on both the original description of the Number Scrabble problem and the optimal visual isomorph, we derived the other three interaction constraints from "multiple sets of cards" to "boundary" by adding more constraints on interaction each time, all of which encode some information about the optimal visual isomorph of the problem.

5 Results

When analyzing the experimental data, we were concerned with the impact of outliers due to random responses. Therefore, we trimmed out the data of four participants whose response times were unusually fast during the pre-test. In addition, 11 of our participants reached the 18-minute time limit during either pre- or post-test, thus their data are automatically dropped since their missing data made it impossible

Index	Isomorph category	Definition
1	Magic square (Visual)	The magic square isomorph.
2	Partial magic square (Visual)	Same layout as the magic square isomorph with different ordering or numbers.
3	Other visual isomorph	Visual isomorph but numbers are not organized in a 3*3 matrix manner.
4	Mathematical isomorph	All possible combinations of 3 numbers adding to 15.
5	Incomplete isomorph	Strategies that do not involve all 9 numbers.

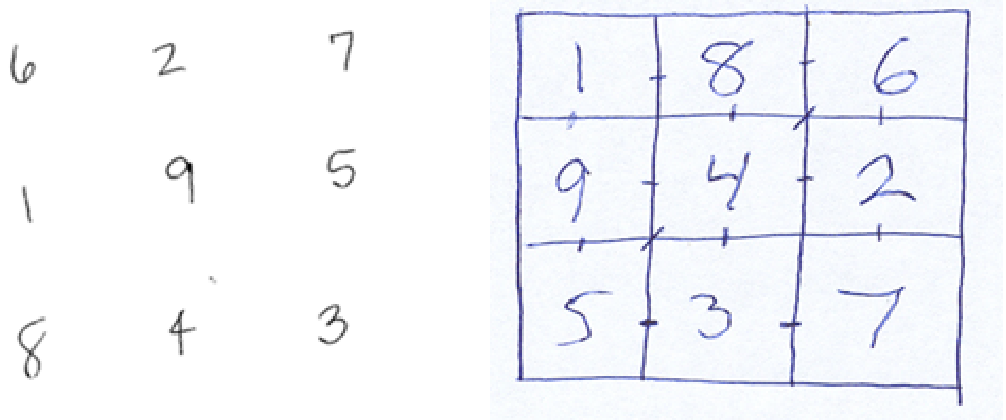
Table 24: Number of visual isomorph developed increases as interaction constrained

to fairly compare pre-test and post-test scores. As a result, we have valid data from 100 participants with 20 subjects under each interaction constraint.

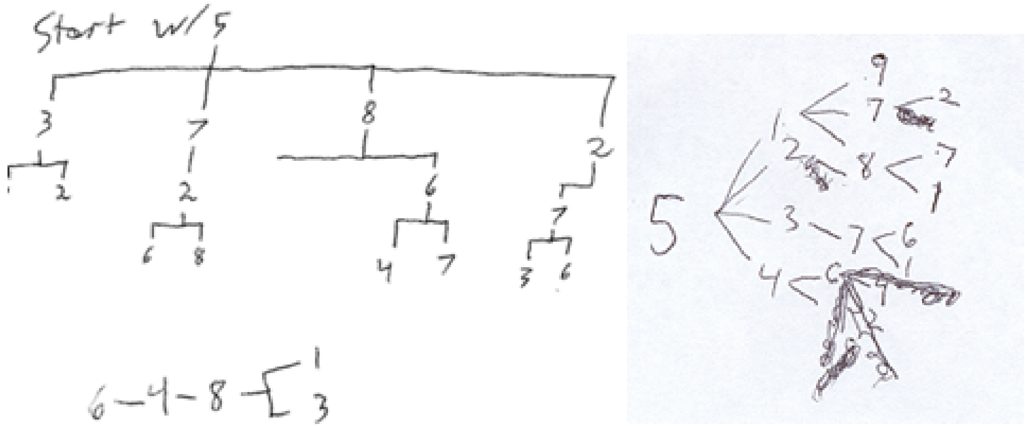
5.1 Isomorph vs. Interaction constraint

Based on the strategies recorded on their cheat sheets, our participants developed a wide range of problem isomorphs during the experiment. Some of these are visual while the others are either mathematical or purely descriptive. We classified these isomorphs into five different categories, described in Table 24. Note that categories 1–3 are visual isomorphs of the Number Scrabble problem while 4 and 5 are not. In addition, examples of different types of isomorphs are shown in figure 51.

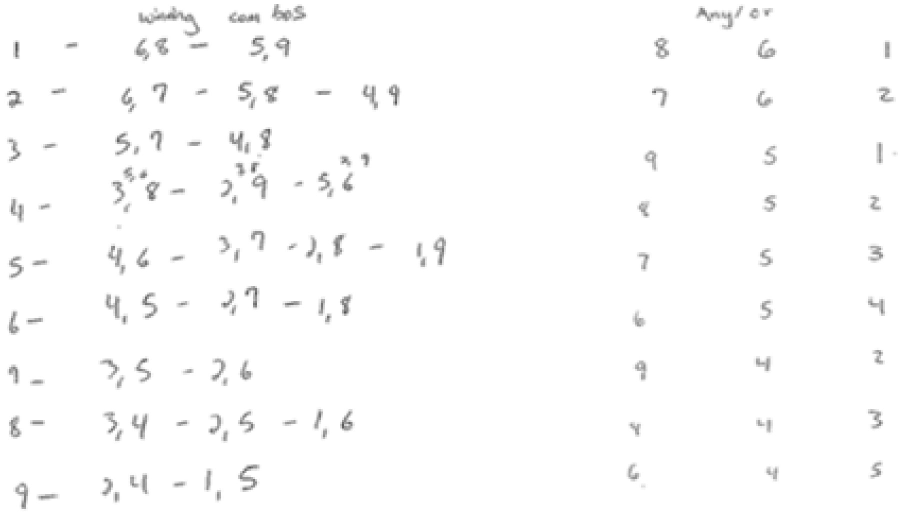
The distribution of different isomorphs developed by our subjects within each interaction constraint is shown in Figure 52. This distribution supports our hypothesis in the sense that as the interactions become increasingly costrained (from pen and paper to boundary), more participants developed visual isomorphs of the number scrabble problem. More importantly, nine out of 20 subjects under the most restrictive constraint (boundary) discovered the optimal visual isomorph (the magic square) while another six subjects developed partial magic square isomorphs. In contrast,



(a) Partial magic square examples



(b) Other visual isomorph examples



(c) Mathematical isomorph examples

Figure 51: Isomorph examples

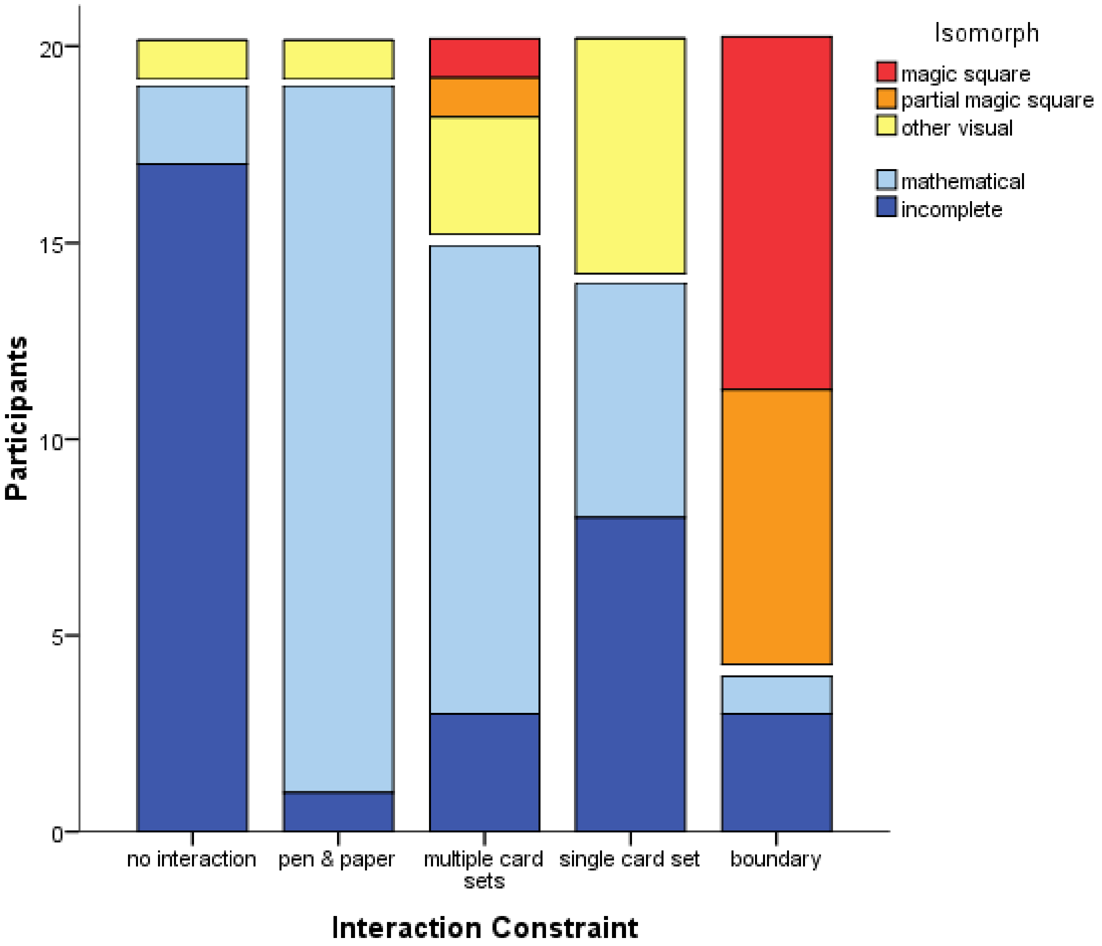


Figure 52: Distribution of isomorphs developed under five different interaction constraints. The gaps divide visual isomorphs (1,2 and 3) from non-visual isomorphs (4 and 5).

only one out of 20 participants in either the no interaction condition or the pen and paper condition discovered any visual solution. A Pearson's chi-square test of independence finds a highly significant interaction between interaction constraint and isomorph, $\chi^2(16, N = 100) = 116.9, p < .001$. Since 15 cells have an expected count of less than five, we performed a Fisher's exact test which also yielded a probability of $p < .001$.

5.2 The effect of interaction constraints on Response Time and Score

Results regarding time and score were analyzed statistically using an analysis of variance (ANOVA) followed by Tukey's HSD (Honestly Significant Difference) test for pairwise comparisons. The factor in our experiment was interaction constraint (five levels) and the dependent variables were improved response time and improved score.

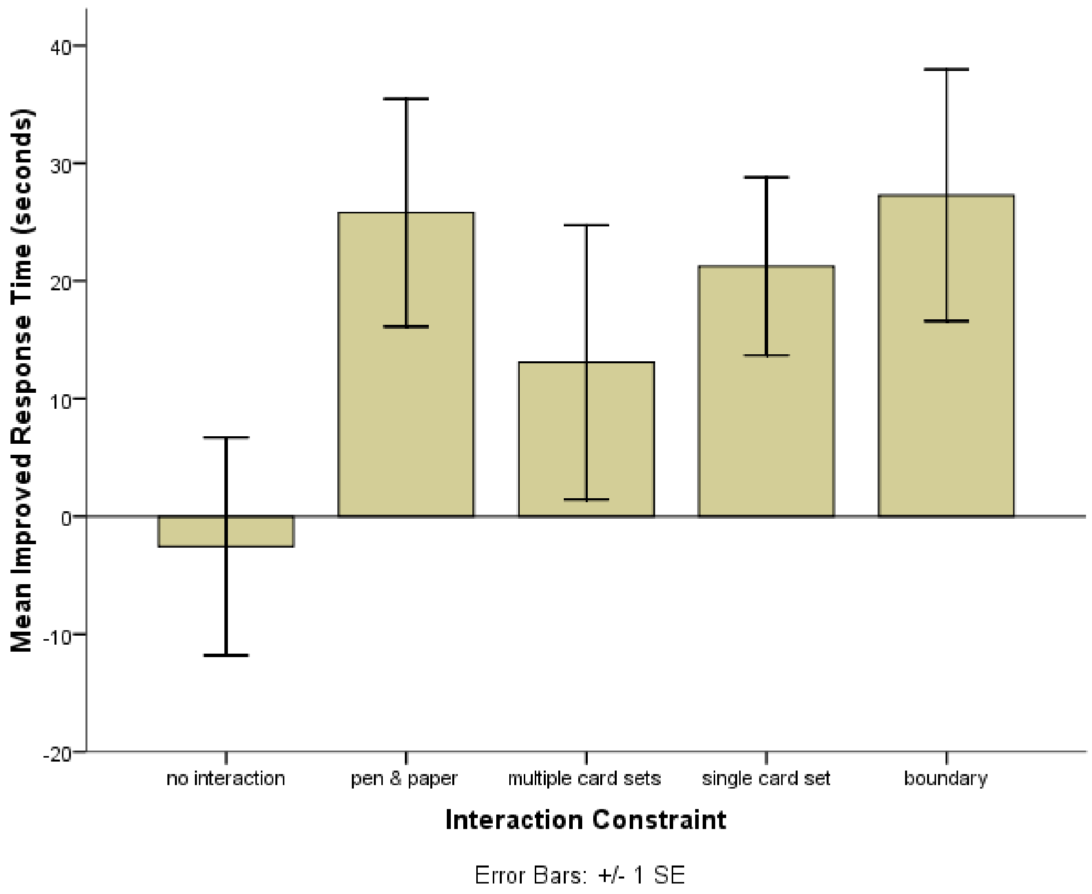
Improved response time is derived from the time it took to decide which card to choose next at each move during a game. Response time per game is defined as the average time it took the participants to choose the next card during each game, $T = \sum ResponseTime/n$, with n being the number of cards chosen following the opponent's move during a specific game. Since both the pre-test and post-test sessions comprise six games, improved response time is thus defined as $IT = \sum_{i=1}^{6} T(i, posttest) - \sum_{i=1}^{6} T(i, pretest)$. In a similar vein, improved score is derived from whether the subjects tied or lost to the computer during each game, with tying counted as 1 point and losing as 0 points. Thus improved score is defined as $IS = \sum_{i=1}^{6} S(i, posttest) - \sum_{i=1}^{6} S(i, pretest)$.

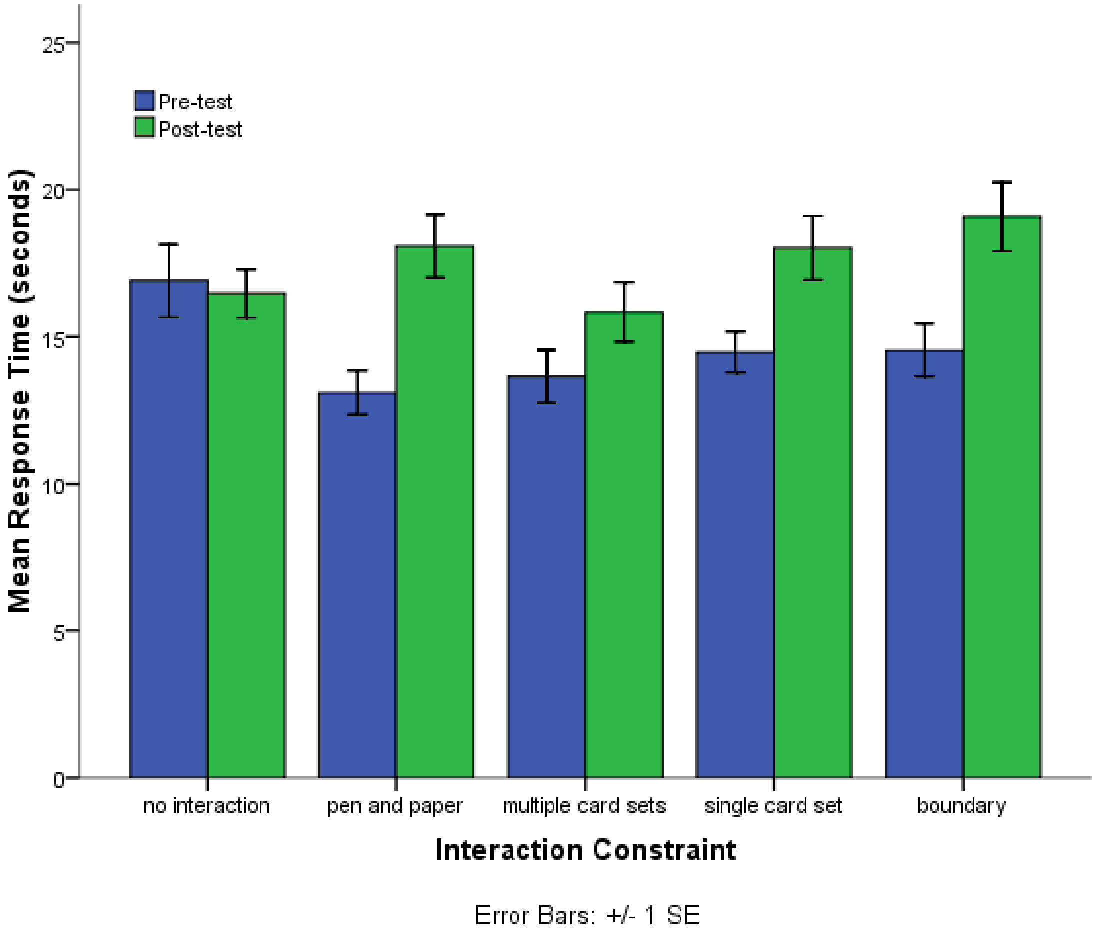
5.2.1 Response time

We expected participants to choose the next card faster during the post-test as the interaction constraints increased, since we hypothesized that they would be more likely to derive a better visual isomorph similar to the "magic square". However, we did not observe a significant main effect of improved response time ($F(4, 95) = 1.54, p = 0.097$). Figure 53 (top) shows the mean improved response time under different interaction constraints. However, interesting yet surprising findings emerged once we considered response time during pre-test and post-test separately. Figure 53 (bottom) shows the mean response time during both pre- and post-tests under the five interaction constraints. It should be noted that participants in the no interaction condition had an unusually slow average response time in the pre-test, which makes comparisons between that condition and the others problematic. In general, however, we found that most of our participants spent more time deciding which card to choose next during the post-test, and participants under the most confined constraints took the longest time, which ran counter to our expectations. We discuss possible reasons for this in Section 6.

5.2.2 Score

If we consider mean scores on the pre-test and the post-test separately (Figure 54 (bottom)), it is clear that in general our participants scored higher after the strategizing session under all five interaction constraints ($F(1, 1190) = 57.7, \eta_p^2 = 0.046, p < .001$). More importantly, the subjects in the more constrained interaction groups tend to score higher than those in the less restrictive interaction groups.


Mean Improved Response Time (seconds)
40
30
20
10
0
-10
-20
no interaction
pen & paper
multiple card sets
single card set
boundary
Interaction Constraint
Error Bars: +/- 1 SE


🟦 Pre-test
🟩 Post-test
Mean Response Time (seconds)
25
20
15
10
5
0
no interaction
pen and paper
multiple card sets
single card set
boundary
Interaction Constraint
Error Bars: +/- 1 SE

For improved score (Figure 54 (top)), we observed a significant main effect of interaction constraint type ($F(4, 95) = 6.5, \eta_p^2 = 0.215, p < .001$). Post-hoc tests showed that the improved scores are significantly different between numerous pairs of interaction constraints. To elaborate, the improved score for participants assigned to interaction constraint #5 (boundary) is significantly larger than that for participants assigned to interaction constraint #1 (no interaction), $p = .001$, constraint #2 (pen and paper) with $p < .01$, and constraint #4 (one set of cards) with $p < .01$. Although the result of other pairwise comparisons were not significant, we can see a clear trend (Figure 54 (top)) that as the interaction constraints become more restrictive, the improvement of score increases except in the case of constraint #4. We further analyze this unexpected "dip" in the discussion section.

5.3 The effect of isomorph on improvement of score

Overall, the main effect of types of derived isomorph is significant ($F(4, 95) = 8.495, \eta_p^2 = 0.263, p < .001$) on improved score (figure 55). Post-hoc tests showed that the improved scores for participants who derived the magic square isomorph is significantly higher than for participants who derived partial magic squares at $p < .05$, and significantly higher than those of all other participants at $p < .01$. The result supports our hypothesis that the optimal solution does lead to much better performance in terms of accuracy. Although the other pairs are not significantly different on mean improved score, we can see a trend that as the isomorphs are further from the optimal magic square, the mean improved score decreases. We further performed a linear contrast between visual isomorphs (1, 2, 3) and non-visual

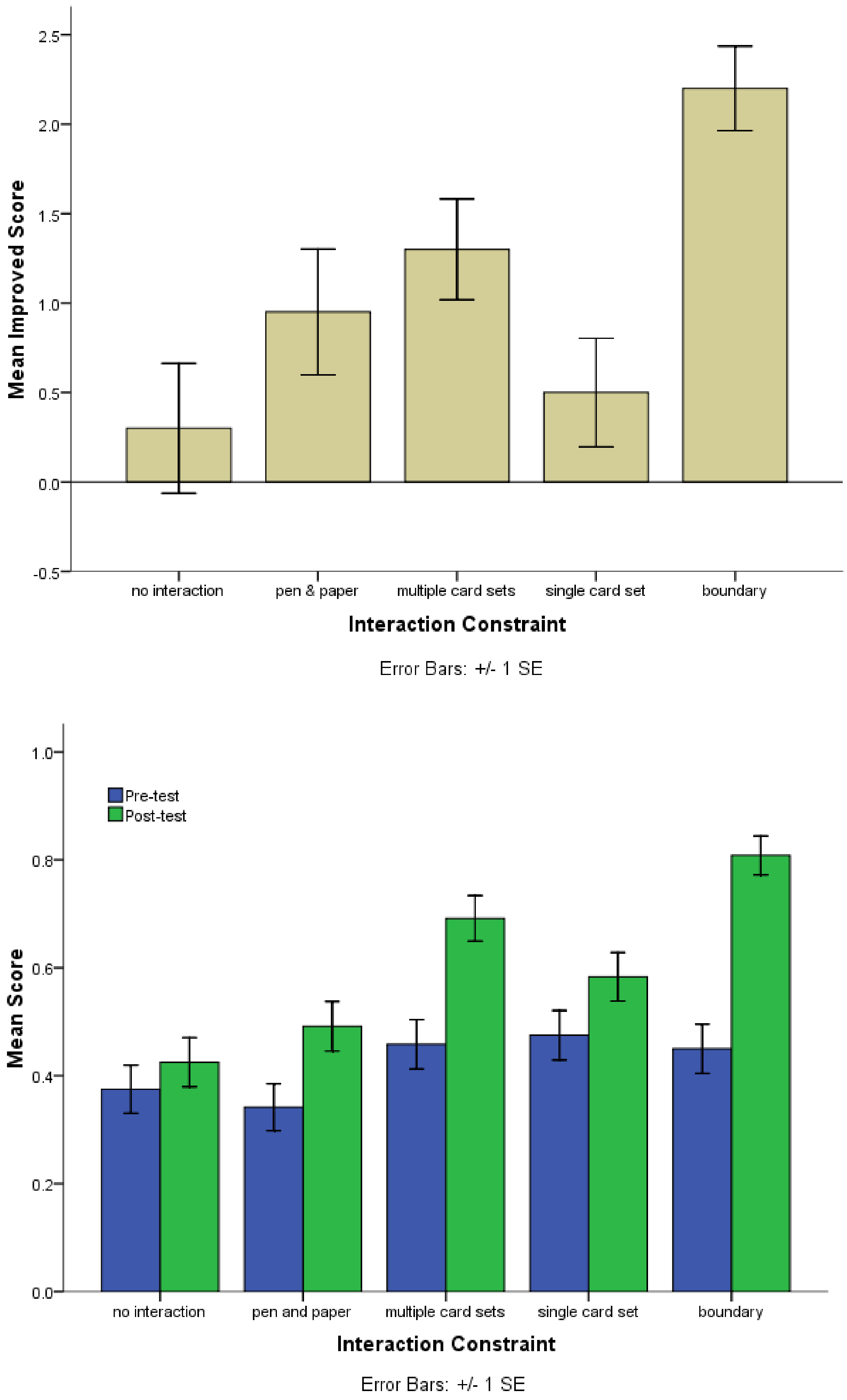


Figure 54: (top) Mean improved score; (bottom) mean score (pre vs. post test)

isomorphs (4, 5) on improved score. The result shows that the mean improved score for participants using visual isomorphs is significantly larger than for those using non-visual isomorphs ($t(95) = 3.822, p < .001$).

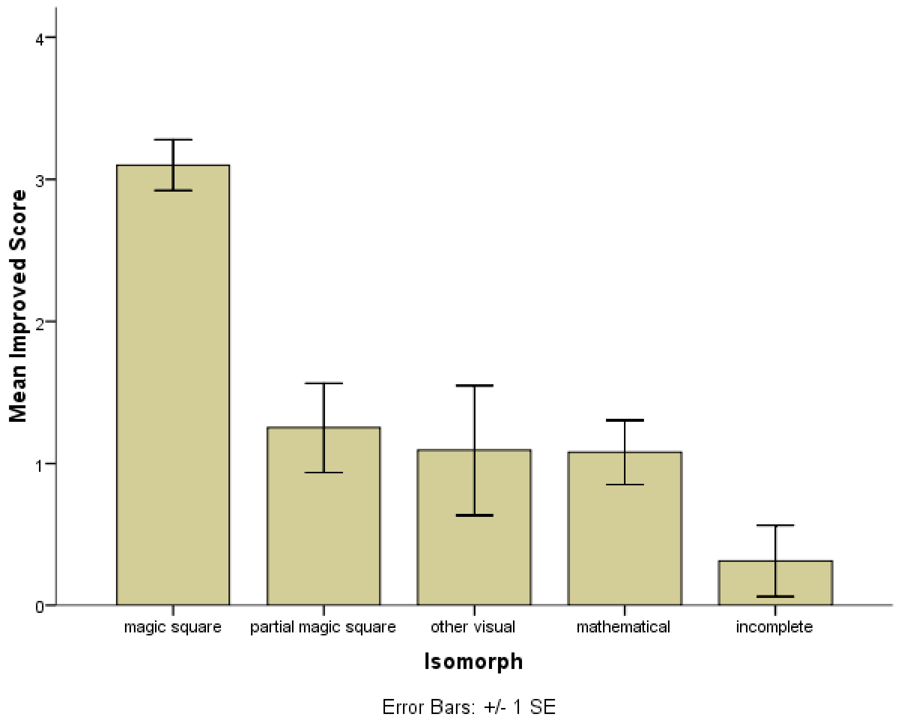


Figure 55: Mean improved score vs. Isomorph

6 Discussion

We start our discussion by addressing the key questions based on our hypothesis:

Do more confined interaction constraints yield a better chance of deriving a visual isomorph?

Yes, based on figure 52 and the chi-square analysis (section 5.1), we observe that as the interaction constraints are increasingly restricted, larger number of visual isomorphs are developed. In addition, the strictest interaction constraints led to the highest number of the optimal visual isomorphs discovered. Nine out of 20 participants under constraint #5(boundary) discovered the magic square isomorph during the strategizing session and seven participants out of the remaining 11 discovered a partial magic square isomorph. Based on further analysis of feedback about the interaction constraints, most participants under this condition found constraint #5 very helpful in their discovery of the visual isomorphs. Many of them left comments such as, "It helped me visualize the problem and make competitive moves." Similarly, most subjects under interaction constraints #3 (multiple sets of cards) and #4 (one set of cards) felt that being able to manipulate the cards freely was helpful. Thus both statistics and user feedbacks support the hypothesis that interaction constraints significantly affect the types of isomorphs users are able to derive by altering the way participants approach the same problem. In other words, the manipulation of the isomorphs could be embodied in the interaction.

Does a more advanced visual isomorph outperform a non-visual isomorph in terms of score?

Yes. We consider an isomorph as more advanced if it is more similar to the optimal visual isomorph (the magic square). Thus our results summarized in Section 5.3 confirm that visual isomorphs lead a greater increase in score compared to non-visual isomorphs. What's more, within the group of visual isomorphs, the optimal visual isomorph outperforms the other two significantly.

Does more confined interaction constraint always yield larger improvements on score?

The short answer is: not always. As seen in Figure 54, the general trend shows that as the interaction constraints become more restricted, the improved score tends to rise, with the exception of constraint #4 (one set of cards). The low improved score in this condition can be explained by considering Figure 52, which shows that none of the participants under this condition derived a magic square (red) or partial magic square (orange) isomorph. Without more efficient visual isomorphs, it made sense that the subjects did not do much better in their post-test compared to the pre-test. However, when we designed the five interaction constraints, we considered one set of cards as a highly restrictive constraint, thus we expected better scores and more derivation of the optimal isomorph. Based on the comments they left, many participants in this condition felt limited by only being able to interact with one set of cards and wished they were given paper to write down combinations of numbers they found to offload the burden of having to memorize them. After the experiment, when we present the magic square isomorph to participants, most in this condition thought they were close to discovering the optimal isomorph at some point during the experiment. But without the extra boundary to further constrain their interaction, it was hard for them to find the bridge between one set of cards and the magic square. This finding highlights the fact that more restrictive interaction constraints are not necessarily helpful unless they meaningfully encode information about the problem. The single set of cards constrained interaction, but without the boundary this constraint did not by itself tell participants anything about the nature of the problem.

Why is improved response time not a good measure?

Unexpectedly, we did not observe a significant result of isomorph type in terms of post-test response time. In fact, response times in the post-test were generally longer than in the pre-test, and participants who discovered the optimal isomorph tend to take an especially long time responding during the post-test. We contacted them afterwards about why they made decisions more slowly during the post-test and found out that instead of playing defensively using the magic square, they spent more time thinking about how to beat the computer. Thus we can infer the bar this particular group of participants set was higher than just “not to lose.” Overall, it may have been the case that participants in the post-test took a longer time because they were consulting their cheat sheets or otherwise thinking harder about their strategy, as we encouraged them to do in the strategizing session.

Another reason we did not observe a significant result of different types of isomorphs on improved score is that the search time for each of the visual isomorphs our subjects derived to decide the next card might vary drastically. For example, searching through a partial magic square should yield a faster decision than searching through a 9x9 matrix, while searching through a 9x9 matrix leads to a faster decision than going through all possible combinations of three numbers adding to 15. Overall, since there are many other factors involved in the improved response time (such as search time and self-expectation of performance), we did not observe a strong causal relationship between types of isomorph and improved response time.

7 A note on the variety of visual isomorphs

In Table 24, we roughly categorized all the isomorphs our subjects developed during the study into five categories including three visual and two non-visual isomorph types. In this section we mainly focus on the visual isomorphs discovered by the participants. It is interesting to see that eight participants across interaction constraint #3 (multiple set of cards) and #5 (boundary) developed a partial magic square isomorph, and that 11 participants discovered other forms of visual isomorph across interaction constraints #1, 2, 3 and 4. Within the partial magic square isomorph, there are many variations. Figure 51(a) illustrates a few of them, and we can see that the variations are mainly caused by ordering. There are even more variations under the "Other visual isomorph" category. One type of variation was a decision tree, such as the examples in Figure51(b); additionally, a few participants built a 9x9 matrix (Figure 56).

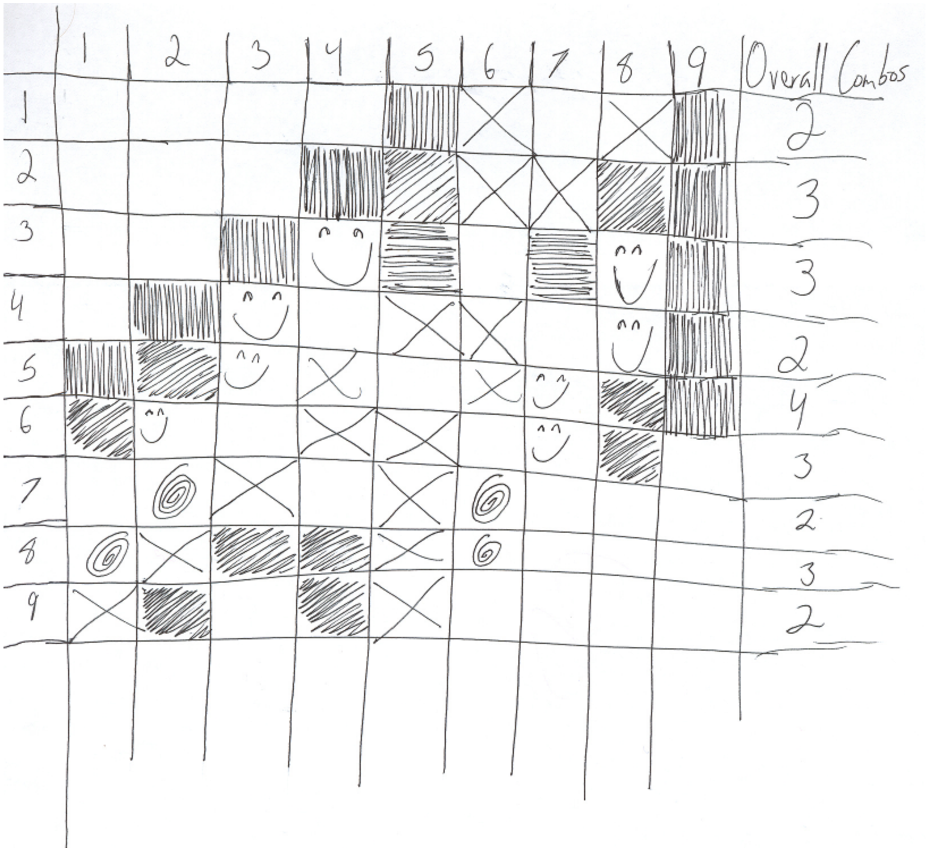


Figure 56: A matrix-like visual isomorph

In Figure 52 we can see a strong contrast between the types of visual isomorphs the participants came up with. Most participants under interaction constraint #5 (boundary) developed magic square-like visual isomorphs during the strategizing session, while there are a relatively larger number of participants under both constraints #3 and #4 who discovered more creative visual isomorphs (such as different forms of decision trees and node-link diagrams). Thus, there seems to be a trade off between interaction constraint and the creativity of the resulting visual isomorph.

8 Implications and Future Work

Our findings suggest that there is a clear connection between the nature of interactions available in a visual representation and the types of strategies users tend to develop when working with the representation. While we have demonstrated this in the context of a specific problem-solving scenario, we argue that our results have significant implications for the more general area of interaction with visual representations with which visual analytics concerns itself.

In particular, this research suggests that degree of constraint is an important dimension to consider when designing interactions for visual analytics systems, although this is not a common way of talking about interaction design in visualization. In cases where a task has an optimal solution path—for example, when there is a standardized procedure that analysts are expected to follow—highly constrained interaction is likely to be a good way to guide a user towards this procedure without the need for extensive training. In situations where the designer needs to encourage creative solutions to a problem, some middle ground between constrained and unconstrained interaction is likely to be more helpful. One strong implication of our findings, however, is that complete freedom of interaction may make problem-solving more difficult; encoding some degree of boundaries into the interaction will likely help users to understand the task in a more intuitive fashion.

As demonstrated by our results, the optimal visual isomorph indeed makes the Number Scrabble problem easier to solve. But as mentioned in Section 2.2, efficiency is not the only measure of interest in visualization; our goal is to make information not just accessible, but understandable. In this context, it is worth mentioning that we had one participant who discovered the magic square visual isomorph but failed to realize that the nature of the game is just like tic-tac-toe given the optimal isomorph. While one incident does not warrant enough evidence to confirm or counter any existing theory, it is an interesting phenomenon to consider.

Since the problem we considered has a known and clearly defined optimal visual isomorph, our designed interaction constraints were geared towards this isomorph. Realizing the limitations of our task, we certainly hope that this proof-of-concept could be generalized to more complex problems. The obvious next step is to examine how to design interaction constraints for problems that might not have known optimal visual isomorphs.

9 Conclusion

We have demonstrated that constraining user interactions indeed affects problem-solving through exploring the relationship between interaction constraints, visual isomorphs, and problem-solving performance as measured by response time and score. Our results showed that more confined constraints lead to better visual isomorphs, and better visual isomorphs result in large improvements in scores on the Number Scrabble game. Our hypothesis is further confirmed by a significant effect of interaction constraints on improved score. Overall, our results indicate that the manipulation of isomorphs can be embodied in user interaction by imposing different constraints, and that certain interaction constraints can lead to a higher chance of deriving a better visual isomorph for a problem. With better visual isomorphs yielding higher performance, our results demonstrate that we can indeed improve the effectiveness of problem solving activities by embodying information in user interaction.