January 2016

# Tools and Methods to Analyze Multimodal Data in Collaborative Design Ideation

Ramaswamy Senthil Kumaran Chandrasegaran
*Purdue University*

Follow this and additional works at: https://docs.lib.purdue.edu/open_access_dissertations

**PURDUE UNIVERSITY**
**GRADUATE SCHOOL**
**Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By  Ramaswamy Senthil Kumaran Chandrasegaran

Entitled
Tools and Methods to Analyze Multimodal Data in Collaborative Design Ideation

For the degree of  Doctor of Philosophy

Is approved by the final examining committee:

Karthik Ramani
Chair

Niklas Elmqvist

Lorraine Kisselburgh

Jitesh Panchal

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s):  Karthik Ramani

Approved by:  Jay P. Gore                                        1/26/2016

Head of the Departmental Graduate Program                Date

TOOLS AND METHODS TO ANALYZE MULTIMODAL DATA

IN COLLABORATIVE DESIGN IDEATION


A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ramaswamy Senthil Kumaran Chandrasegaran


In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy


May 2016

Purdue University

West Lafayette, Indiana

To Madhu, with all my love.

ACKNOWLEDGMENTS

I consider myself incredibly fortunate to have had the opportunity to work with, and learn from, an excellent set of teachers and friends. This thesis would not have been possible without them.

I am extremely grateful to Dr.Karthik Ramani, for giving me this opportunity to switch tracks from the industry back into academia, for supporting, challenging, and guiding me through my research, and for giving me the opportunity to experience and enjoy teaching here at Purdue. I cannot thank Dr.Niklas Elmqvist enough, not just for his excellent courses in Visualization and Visual Analytics that helped shape my research, but also for his attention to detail, compassion, and sense of humor, although I don't think he was joking when he said "BibTeX cleanliness is next to Godliness". I am also grateful to Dr.Lorraine Kisselburgh and Dr.Kylie Peppler whose mentorship and advice, especially on qualitative studies and analyses, have been invaluable.

Most of the work that you see in this thesis is a result of my close collaboration with Karthik Badam. Among other things, he taught me JavaScript, while I introduced him to Jeeves and Wooster. I'm not sure who got the better end of that bargain.

I also thank my colleagues Elkin Taborda, Zhenpeng Zhao, and Dr.Ninger Zhou, who were my other co-authors in the publications that have helped form this thesis.

My early experiences at Purdue were shaped by my interactions with Dr.Dave Anderson, Dr.Ram Sriram and Dr.Shreeram Abhyankar. Their depth of knowledge and meticulousness will always be a source of inspiration.

For their delightful company, *really* awful jokes, and their sage advice, I thank my close friends and current and previous labmates, especially Cecil, Bill, Dev, Sundar, Vinayak, and Woodie. I've learned more than I can remember from you.

My perspectives on engineering, design, and life in general would never have been the same without my friends and colleagues back home in India. Anil, Saravanan,

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

# ABSTRACT

Chandrasegaran, R. Senthil K. Ph.D., Purdue University, May 2016. Tools and Methods to Analyze Multimodal Data in Collaborative Design Ideation. Major Professor: Karthik Ramani, School of Mechanical Engineering.

Collaborative design ideation is typically characterized by informal acts of sketching, annotation, and discussion. Designers have always used the pencil-and-paper medium for this activity, partly because of the flexibility of the medium, and partly because the ambiguous and ill-defined nature of conceptual design cannot easily be supported by computers. However, recent computational tools for conceptual design have leveraged the availability of hand-held computing devices for creating and sharing ideas. In order to provide computer support for collaborative ideation in a way that augments traditional media rather than imitates it, it is necessary to study the affordances made available by digital media for this process, and to study designers' cognitive and collaborative processes when using such media. In this thesis, we present tools and methods to help make sense of unstructured verbal and sketch data generated during collaborative design, with a view to better understand these collaborative and cognitive processes. This thesis has three main contributions.

We first present use text visualization techniques to analyze verbal data generated during and about a brainstorming session, and use concept maps and lexical dispersion plots to provide an overview of the design process through these representations. We also use concordance views to disambiguate concepts of interest.

Secondly, we identify requirements for a digital framework that supports collaborative sketching in early design, and use skWiki, a framework designed with these requirements in mind, to understand the physical and cognitive processes involved in using such a framework. We identify the value of group awareness—to both the design team and the design researcher—afforded by skWiki's representation of collaborative

sketching in the form of "paths" of sketches. We identify, through skWiki's *branching* operation the associated cognitive processes of *lateral* and *vertical* transformations that the former involves. We also propose the cognitive processes of *reuse, refactor, repurpose,* and *reinterpret* associated with skWiki's *merging* operation.

Finally, we present a web-based visual analytics platform called VizScribe that supports the representation of the design process through the traditional medium of video and transcript, augmented by the text visualization and "paths" representation discussed earlier. We introduce interaction paradigms that help researchers query, filter, and code such data to make sense of collaborative design activity. We illustrate the extensibility of this framework to represent other temporal data forms, and to support collaborative coding.

PUBLICATIONS

This thesis includes the following published work by the author.

Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horvth, I., Bernard, A., Harik, R.F. and Gao, W. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, 45(2), pp.204-228, 2013.

Chandrasegaran, S., Kisselburgh, L. and Ramani, K. Understanding Brainstorming Through Text Visualization. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V001T04A020), 2013.

Zhao, Z., Badam, S.K., Chandrasegaran, S., Park, D.G., Elmqvist, N.L., Kisselburgh, L. and Ramani, K. skWiki: a multimedia sketching system for collaborative creativity. In *Proceedings of the ACM conference on Human Factors in Computing Systems* (pp. 1235-1244), 2014.

Chandrasegaran, S., Badam, S.K., Zhao, Z., Elmqvist, N., Kisselburgh, L. and Ramani, K. Collaborative Sketching With skWiki: A Case Study. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V007T07A041), 2014.

## 1. INTRODUCTION

### 1.1 Motivation and Overview

Product design and development are global ventures that require distributed teams of designers and engineers to work together [1, 2]. This has required designers to use digital tools and media, in order to express and communicate their ideas quickly and efficiently with stakeholders located across the world. Product design can be divided into five stages: requirements modeling, functional modeling, concept design, embodiment design, and detailed design [3]. Until recently, these tools have largely supported the later stages, especially detailed design. This is perhaps due to the fact that at this stage, the information pertaining to the product is well-defined and amenable to the kind of unambiguous, formal representations that can be supported digitally. This is true for most computer-supported collaboration tools, whether for socially constructed knowledge bases such as Wikipedia, collaborative text editors such as Google Docs, or in our case, computer-Aided design (CAD) tools such as PTC Creo and product lifecycle management (PLM) systems such as Enovia. These processes where the goal is selection, refinement and detail, and the result is a single product, whether a web page or a car, are termed analytical or "convergent" processes [4].

However, recent computational tools for conceptual design have leveraged the availability of hand-held computing devices and web-based collaborative platforms, thus providing a means to emulate the informal, messy, creative stages in design. These creative stages come under the category of "divergent" processes [4], where the goal is to generate alternatives and explore possibilities without the need to refine ideas. Part of the challenge in supporting divergent processes lies in the nature of creativity itself: a process of continuous inspiration, guesswork, and evaluation results in a stream of ideas, only some of which are developed further.

### 1.1.1 Design as a Creative Process

Design problems arise in a wide variety of domains, ranging from architectural design, to mechanical design, to web design. They also vary in complexity, from designing a doorstop to designing an aircraft wing, and in the tools required, from pencil and paper to mathematical modeling software. The design process, however, retains a degree of sameness across domains and applications. Ulrich [5] provides a generalization for this process, characterizing it through four information processing steps: *perceiving* a gap between what is and what can be, *defining* the problem of why this gap exists, *exploring* alternatives to help bridge this gap, and *evaluating and selecting* an appropriate plan to refine it. This notion of looking at design through the lens of information processing helps us understand why Rittel and Weber [6, p. 161] consider design a *wicked problem*: "the information needed to *understand* the problem depends upon one's idea for *solving* it". This means that the information required to solve the problem is unclear, incomplete, or undefined, and the designer has to rely on their knowledge, experience, and insight to make sense of the information.

This "sensemaking"—the act of integrating one's experiences into an understanding of one's surroundings—is a process of abductive reasoning, involving leaps of logic based on ambiguous information, seeking the best explanation given this information, and refining it in the light of new information [7]. A designer's creativity is thus linked to three main attributes: a comfort with ambiguity, the ability to change constraints, and the ability to form new, often tenuous connections. These connections are either broken or strengthened by new knowledge that is gained through the act of designing, which in turn provides further insight to solve this and other design problems.

### 1.1.2 Challenges to Computer Support for Creative Design

Owen and Horváth [8] classify representations of the designer's knowledge into pictorial, symbolic, linguistic, virtual, and algorithmic forms, all of which are seen through various stages in the design process (Figure 1.1). We can see that early design
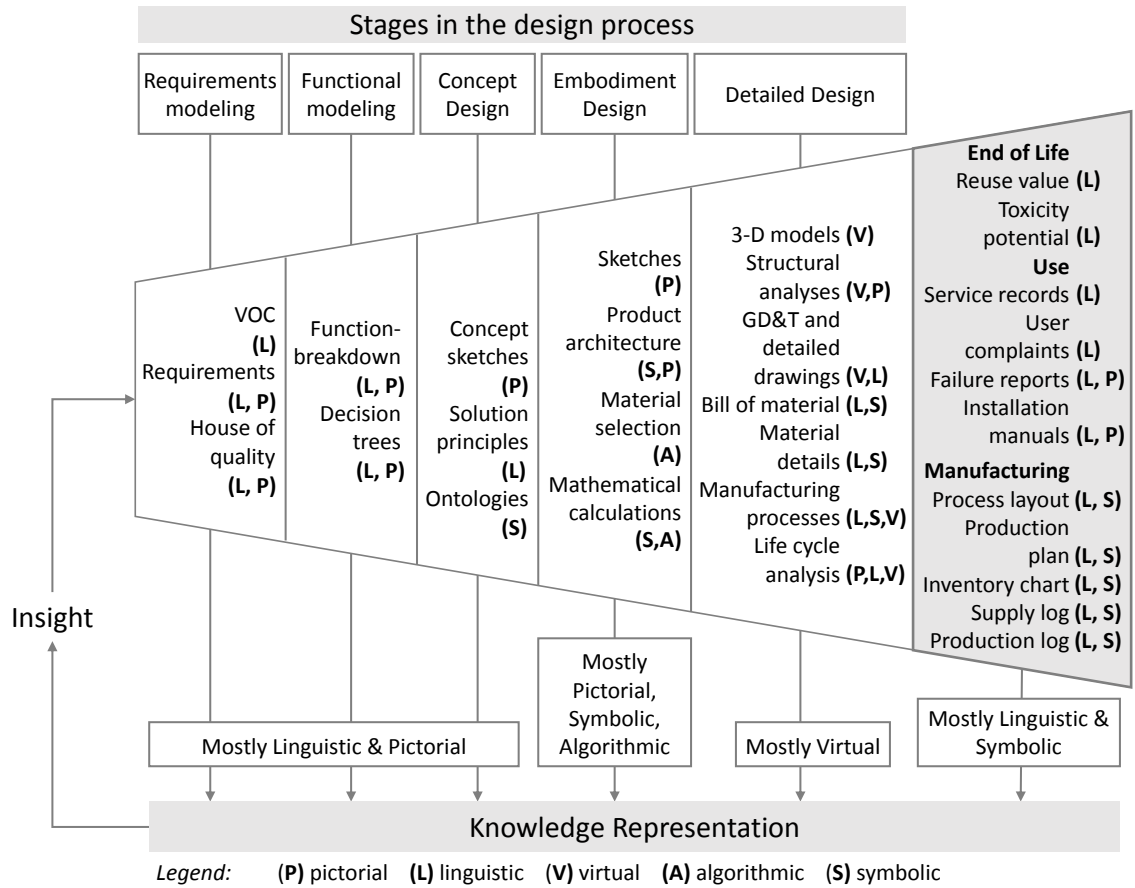
Figure 1.1. The various forms of representation through the design process, categorized based on Owen and Horváth's [8] classification. The stages in the design process shown on top follow Pahl and Beitz [3]. While the list of representations and their classification is more illustrative than exhaustive, we can see that early design chiefly involves linguistic and pictorial forms, while the more formal representations emerge as one moves towards detailed design. Computer support for knowledge representation, essential for the designer's insight, is currently skewed towards the later design stages.

is characterized chiefly by linguistic and pictorial representations (informal sketch and verbal data), while symbolic, virtual, and algorithmic representations (manufacturing drawings, simulations, life-cycle analyses) emerge as the problem space is better defined. A closer look at the list of representations under "detailed design" in Fig-

ure 1.1 reveals that every one of these representations is done today on the computer. This is not true of the preceding stages: as we move earlier in the design process, the computer support drops off. This is chiefly owing to the nature design itself: early design is characterized by exploration and ambiguity, while detailed design requires refinement and unambiguous communication between stakeholders.

Regardless of the sophistication of computer support tools, conceptual design has largely remained the same: collecting information, sketching, modifying, and iterating. Of these, sketching is of special interest, having universally been acknowledged as "the archetypal activity of design" [9, p. 97]. While it is prevalent throughout the design process, it is more dominant in the early conceptual design stages. Sketching is not just a method for documentation and communication, it is also a tool for thinking: the act of sketching provides a memory extension, a visual representation, as well as a baseline for abstraction to the designer [10]. These sketches and other visual representations provide a medium for negotiations and discussions, thus supporting the social and communicative processes in design [11].

The paper notebook, used for jotting down ideas and sketching concepts, is ideally suited to the informal nature of early design sketches [12]. Providing digital support for sketching has gained momentum recently owing to the recent developments in direct-touch tablets and active pens. However, we must keep in mind that computer support tools for ideation should augment instead of imitate [13]: to use the computational and interactive power to augment traditional design methods and tools, rather than imitate them. While the strengths of a digital medium, such as replication, composition, and persistent storage are understood, there remain unanswered questions about what more such digital media can afford teams of designers as well as researchers who study design. Collaboration adds another level of complexity to concept generation and progressive development: an effective communication and iteration mechanism is necessary to effectively utilize ideas generated by individual team members. Digital tools can render sketches persistent through storage, while at the same time make them mutable by enabling replication and editing. However,

developing such a tool needs to go hand-in-hand with observing how it is used, and understanding the potential impact of such tools on the designer's cognitive process.

### 1.1.3   Understanding Collaboration in Early Design

From a design researcher's point of view, understanding how designers think, communicate, and collaborate has typically fallen under the larger umbrella of protocol studies. The act of making connections between numerous and seemingly unrelated pieces of information is one that occurs in the designer's mind, invisible to the observer except for glimpses offered through artifacts such as the designer's notes and sketches. This poses a challenge to design researchers, who need to understand what goes on in a designer's mind when they are engaged in the design activity. Think-aloud protocol studies [14] are commonly practiced in such situations, where the designer is asked to continuously verbalize their thoughts during a design exercise.

While this works to an extent on individual design tasks, it does not work when studying how designers work together. In such situations, the researcher is forced to rely on sketches and records of verbal communication within a design team as the only forms of data. This involves recording the design session on video and audio, collecting their sketches and notes, and then transcribing, segmenting, and coding this data [15]. Recent developments in computational social science has brought in the use of sensors such as sociometric sensors [16], eye-tracking sensors, and psychophysiological measurement devices such as encephalographs and galvanic skin sensors. Today's design researcher needs to leverage the use of these sensors, as well as the data they are now able to collect through the aforementioned digital tools that support design ideation and collaboration. While there now exist computer-aided qualitative data analysis software (CAQDAS) for supporting protocol analysis, these tools do not support a meaningful representation of the multimodal data collected by researchers, nor do they give the researcher the freedom to create their own representations to suit their specific needs.

## 1.2    Organization and Contributions

This thesis presents tools and methods that help make sense of unstructured verbal and sketch data that are generated during collaborative design ideation, in order to better understand early design. Chapter 2 provides a background on existing visual representations of text data, existing computer support for sketching and early design, and existing tools and techniques to aid design protocol studies. The subsequent chapters detail the following contributions:

### 1.2.1    Text Visualization to Understand Collaborative Ideation

As mentioned previously, conceptual design is characterized by informal and ambiguous representations, chiefly sketches and verbal data. Chapter 3 discusses in detail how text visualization techniques such as concept maps, lexical dispersion plots, and text concordances can be used to represent this verbal data in order to provide overview and detailed representations of collaborative ideation. In this chapter, the verbal data generated as a result of team brainstorming exercises are categorized under transcribed speech, sketch annotations, and reports reflecting on the process followed and on the final design. These relatively unstructured datasets are then visualized using the mentioned techniques to show how the processes followed by two different sets of teams can be identified.

### 1.2.2    Cognitive Models to Understand Collaborative Sketching

Based on challenges faced by computer-supported collaboration in early design identified through literature review and preliminary studies chapter, we attempt to better understand the cognitive processes involved when designers use such a platform. Chapter 4 outlines design requirements to better support collaborative ideation, and use a web-based collaborative sketching platform called *skWiki* to create "branch-

ing" and "merging" operations that help designers to view, adapt, and modify each other's sketches to better explore the design space.

This chapter also uses Goel's [17] theory of transformations to understand the cognitive processes behind skWiki's "branch" operation. The chapter discusses how skWiki aids the sketching-related cognitive operations of *lateral* and *vertical* transformations, and how skWiki's representation design activity as a series of "paths" provides designers with an easily accessible representation of their design history. The "paths" model also provides design researchers with a means to observe team activity over longer durations when there is no means to record the team.

The "merge" operation afforded by skWiki is discussed in detail in Chapter 5. Here we identify the different cognitive operations involved in exploring the design space by merging existing sketches. We propose four operations, namely *reuse*, *refactor*, *repurpose*, and *re-interpret*, that aid the transformation of existing ideas into related or new ideas. We show how computer-supported collaborative sketching platforms can support these four operations to aid divergent thinking and delay fixation.

### 1.2.3 Understanding Design Team Behavior Using Visual Analytics

In order to better understand collaborative ideation, it is necessary for the design researcher to make sense of the informal verbal and sketch data generated during this activity. The preceding chapters discuss cognitive operations and representations of sketching and verbal behavior that can aid the researcher. Chapter 6 details how we use these representations along with traditional protocol analysis data such as video and transcript, presenting a web-based, extensible framework called *VizScribe* to analyze design activity. VizScribe uses an *orient-focus-code* interaction paradigm that helps the researcher *orient* themselves to the represented datasets, *focus* on specific data points to understand context, and *code* the data to gain further insights. We show the extensibility of this framework by integrating time-series data from sociometric sensors worn by a design team, and illustrate how such a representation

provides context and meaning to the design researcher. We also illustrate how the web-based framework can be used by multiple researchers to collaboratively code the same dataset, opening up possibilities for more reliable coding processes.

Finally in Chapter 7, we outline the conclusions of this work and its implications to the computer support and analysis of conceptual design, discuss limitations of the proposed tools and methods, and propose future research directions.

## 2. BACKGROUND

In order to understand the needs for and challenges in analyzing collaborative ideation, we must first gain an understanding of the processes involved, the challenges in providing computational support for these processes, and the opportunities provided by such computational support. This chapter thus provides an overview of approaches to collaborative ideation, and the importance of sketching in conceptual design. It then gives a brief overview of prior and existing work to support sketching and collaborative conceptual design. Finally, the chapter ends with a detailed discussion of existing methods, challenges, and opportunities in analyzing such computer-supported collaborative design activities.

### 2.1 Approaches to Collaborative Ideation

Most practical design work happens in teams consisting of multiple participants, often with different and complementary expertise [18, 19]. In such collaborative settings, externalizing the intermediate results of the creative process becomes particularly important so as to allow collaborators to be inspired by the work of each other [2]. Design exploration techniques such as mind-maps and SCAMPER [20] focus on forming connections between existing ideas, or on augmenting or modifying these ideas to generate new ones.

Brainstorming [21] is perhaps the most popular and extensively studied collaborative ideation technique. Other methods for group ideation include the Method 635 [22] and brain writing [23]. Designers use a combination of sketches and actions to communicate with each other [24]. Thus, graphical methods such as the Gallery method [25], brainsketching [26], and the more recent C-sketch [27] have also gained popularity. Evaluations of such techniques have yielded mixed results: studies on

brainstorming have shown that a higher number of ideas would lead to a higher quality of ideas, while others have observed that brainstorming groups end up with ideas that are lower in quantity and quality compared to those of individuals [1].

Collaborative design is progressive: it is as much about developing designs and alternatives as it is about expanding the problem space using concept generation methods. Resolving and combining ideas, concepts, and content from multiple designers is a canonical operation in any collaborative process involving the creation of new artifacts, such as documents, plans, or drawings [28, 29]. Blair and Hölttä-Otto [30] studied progressive idea generation methods such as 6-3-5 Brainwriting, and report that while the contribution to highly original initial ideas is not high, the group significantly increases the originality of previously unoriginal ideas. Each of these methods has its own strengths, and group ideation sessions often use a combination of these methods, or switch from one to another as the situation demands.

Put differently, the ongoing creative output of the design team is key in driving new creative output, which is one of the factors driving the development for computer support for collaborative ideation. Specifically, the focus lies on tools that combine the ease of sketching ideas on paper with the power of networked computers for sharing and iterating over these ideas.

## 2.2    A Brief Note on Sketching

Sketches play an important role in externalizing ideas during early design by providing a visual database of generated ideas that inspires new ones [31]. In fact, sketches in particular have been shown to be more effective than any other medium, including text and cross-representational techniques, for the early phases of ideation and creativity [32]. Designers have been observed to perform different levels of 'shape transformation' when sketching. These transformations can be categorized into general rules with which to interpret the changes that occur to a sketch [33]. A key

insight here is the role that sketches play in informing the design process, both for building on existing ideas as well as exploring new parts of the solution space.

Pen and paper has long been the gold standard for sketching [9, 10], but the emergence of high-quality pen-input devices is starting to make the digital medium more attractive in a wide range of domains, including automotive design [34], architecture [35], and software design [36]. Yang [37] reviewed concept generation and sketching among student designers, reporting that the quality of the design outcome is positively correlated with the number of dimensioned sketches in the design notebook, and makes an argument for electronic design notebooks aiding collaborative conceptual design. Several studies have explored how to replicate the strengths of paper—familiarity, precision, and accessibility—even on digital devices [38, 39] to harness their strengths of persistence, replication, and composition.

## 2.3 Computer Support for Collaborative Conceptual Design

Requirements of computer support tools for supporting conceptual design have been articulated by many. These include enabling easy entry of geometric, numeric, and text data, capturing ambiguity of early concepts with imprecise dimensions, allowing separate representations for conceptual designs (sketches) and detailed designs (CAD models), allowing selective development of details, and enabling easy reviewing of alternatives [40]. Early work in collaborative creative support for design was in the form of multi-user drawing support [41–43] for collaboration between geographically distributed participants. Wang et al. [44] review collaborative conceptual design tools, and report that most tools support the later part of conceptual design. They identify key challenges in the field, including but not limited to web-based collaborative conceptual design, tools for managing conflict resolution, knowledge management in collaborative environments, distributed design project management, and intelligent web-based interfaces. They observe that there were few tools that support early stages of conceptual design such as brainstorming and sketching.

Dorta [13] developed a Hybrid Ideation Space (HIS) which uses a spherical mirror model to provide an immersive virtual reality environment that incorporates a digital sketching interface as well as a 3D scanner for 2D and 3D exploration and manipulation of ideas. Clayphan et al. [45] devised a tabletop interface called "Firestorm" for text-based brainstorming, with a view to faster idea generation, high visibility of ideas, and easy selection and arrangement of ideas. Participants use keyboards to generate brief descriptions of ideas, without figures, and collaboratively use a tabletop interface for arranging and selecting ideas, which are color-coded to each participant. Support for multi-user sketching support was pioneered by the i-LAND environment [46] that uses personal sketching interfaces as well as collaborative display walls. TEAMSTORM [47] and GAMBIT [48] allow sketching on mobile devices, coupled with a large display for sharing and manipulating sketches. IdeaVis [49] recognizes the designer's preference for sketching on paper, and uses cameras and projectors to integrate paper into a collaborative environment that uses hybrid media. GAMBIT [48] is a web-based system for sketching user interfaces in co-located collaboration involving multiple devices and displays. Most of these systems require the use of an environment specifically designed for the purpose, and thus pose challenges in the context of design teams, especially student design teams, that are unstructured and mobile. Finger et al.'s Kiva Web [50], a system for group collaboration, circumvents this limitation by requiring web-enabled computers and a projector, available in most generic meeting rooms.

Naturally, there exists groupware that focus on the shared creation of other types of artifacts beyond sketches. Collaborative editors such as Google Docs and ShareLatex support concurrent editing, often using a form of operation transform (OT) [51, 52]. Chronicle [53] clusters, probes, and visualizes a document's workflow history, whereas MeshFlow [54] visualizes, clusters, annotates, and filters the history of operations on polygonal meshes. Finally, dating back to early offerings such as the Source Code Control System (SCCS) [55], version control systems manage content change and maintain a history of its evolution, and are now commonplace in the

software industry, e.g., git, SVN, and CVS. The recent Diffamation [56] method uses animation and color encoding to help developers understand the work done by other team members in such version control systems. While digital creativity platforms offer means to generate different alternatives to existing sketches, and groupware provides means to keep track of collaborative workflow history, there needs to be a better understanding of the various physical and cognitive processes that are involved collaborative conceptual design.

## 2.4    Analyzing Design Processes

Recent studies on design cognition has drawn significant attention towards the design processes and on how knowledge and understanding of the designer is affected by design processes and interaction among teams. In this section we discuss existing tools for protocol studies. We then motivate the need for two kinds of visualization techniques for such studies: event-based representations for temporal analysis, and text visualizations for inferring patterns in the structure and semantics of the transcribed text.

### 2.4.1    Methods to Analyze Design Processes

Stempfle and Badke-Schaub [57] propose a generic model of design activity in teams based on the four basic operations of generation, exploration, comparison and selection, and apply it in their study of design teams. They report that traditional focus of design methodology needs to focus not only on the solution concept, but also on the time and cognitive effort to generate the solution. Mumford et al. [58] in their extensive review of methods in creativity research posit that creativity is "a product of work on a particular type of problem" which is: (1) ill-defined, (2) novel, (3) demanding, (4) complex, and (5) exploitable. They recommend that studies of creativity and innovation require a multi-method, multi-measure approach. Gero et al. [59] through their protocol studies on engineering students following three concept

generation techniques of brainstorming, morphological analysis and TRIZ conclude that using structured methods tend to help students focus better on the structure of a solution. Jin and Chuslip [60] focus on the issue of "mental iteration" in engineering design, defined as "a goal-directed problem solving process", modeled as a "sequence of transition behaviors between information processing and decision-making". They studied the process of iterative thinking in designers and identify three distinctive "global iteration loops": problem definition loop, idea stimulation loop, and concept reuse loops. They further conclude that "creative design involves more iterations than routine design". This idea of iteration promoting more creative thinking has been empirically suggested, one could argue that this explains the design expert's emphasis on early design having alternating cycles of divergence and convergence.

### 2.4.2   Existing Tools for Protocol Studies

Early work on computational support for protocol studies included artificial intelligence based systems such as PAS-II [61], which incorporated linguistic processing on task verbalizations to generate graph representations of human knowledge, and KRITON [62], a knowledge-extraction system that infers knowledge elements and forms relations using propositional calculus. Other approaches to support such studies used generic software such as AQUAD [63] and SHAPA [64] for collecting and organizing gathered data and metadata, along with annotations.

Commercial tools for qualitative analysis include multimedia processing tools such as ATLAS.ti[1] and NVivo[2]. These allow visualization and annotation of video timelines as well as of transcribed text, preserving associations between the two. The Computational Analysis Toolkit [65] extends ATLAS.ti to a web-based framework while increasing coding flexibility and speed. StudioCode[3] is another popular tool for video and transcript analysis. Other tools, such as LINKOgrapher [66] are developed

---

[1]http://www.altasti.com
[2]http://http://www.qsrinternational.com/
[3]http://www.studiocodegroup.com

for more downstream applications: analysis of codes using a predefined ontology in the context of conceptual design. However, these tools cannot provide simultaneous views of multimodal data, nor can they adapt to diverse and evolving forms of data that protocol studies are beginning to entail. VizScribe allows the integration of such data, allowing a synchronized view of all recorded events accompanying the main audio/video and transcript, thus providing context to the user.

Additionally, while each of these tools has their specific advantages and disadvantages, there are issues such as forced workflows (imposing a specific style and sequence in coding), coding fetishism (using coding irrespective of whether or not it is appropriate), distancing the user from data, or, at the other end of the spectrum, difficulty in conceptual abstraction [67]. Complete automation of the coding process is another challenge. For example, automated coding tools need better approximations of linguistic features inherent to coding schemes [68]. While it is unclear as to whether a VA-based approach will mitigate some of these issues, a combination of visual representation and computational analysis would serve to augment the pattern-recognizing strengths of the human user and the data processing power of the computer.

### 2.4.3  Visualizing Event-Based Data

Analysis of data collected from protocol studies involves a temporal component, with a view to identifying and annotating co-occurring events. We restrict this section to related work in timeline visualizations of categorical data, which is more relevant to protocol studies than numerical data.

Lifelines [69] is a general representation of biographical data, visualizing discrete events and relationships, and allowing focus on specific parts on the timeline information. Wang et al. [70] use timeline-based interactive visualizations to align events in patient medical records in order to identify co-occurrences of other related events. Challenges in interacting with such representations are illustrated by Monroe et al. [71], who develop a visual query mechanism for these events.

CyberForensic Timelab [72] uses timeline displays for establishing history in cyberspace. It displays a timeline view of electronic records of personal, time-stamped events, to provide the investigator with a visual history of events. PortVis [73] uses an overview and detail-on-demand approach to display activity on a large set of TCP ports in a network over time, identifying traffic anomalies that signify possible attacks on the network.

Stab et al. [74] use timeline views for more general applications: they develop SemaTime, a temporal visualization tool that allows hierarchical categorization and filter of domain-specific data. It also incorporates semantic relationships between entities, similar to Continuum [75], which also provides histogram overview and timeline detail view of temporal events. Temporal relationships are represented as spans of bounding boxes, providing a visually pre-attentive visualization, with the level of detail controlled using a "dimension filter". Rubin et al. [76] present a set of tools for Navigation and editing between speech and transcript for high-level editing of interviews to create "audio stories". These tools, though not meant for protocol studies, use methods that are both relevant and useful for future iterations of VizScribe.

More relevant to our work are the Digital Replay System [77], an ethnographic tool that uses an ontology-based data representation for multimodal data analysis, and Chronoviz [78], a timeline-based annotation and visualization of multimedia data that uses timestamps to display video or audio data with related timeline data. VizScribe differs from these tools in two main ways: (a) it links multiple visualizations using Infovis techniques, and (b) it allows for extensibility of the timeline views to accommodate custom datasets, catering to changing data-acquisition technologies.

### 2.4.4 Text Visualization

Protocol analysis almost always include transcribed text as a main data format. These are thus uniquely suited for a combination of text visualization linked to associated timeline visualizations discussed earlier. Previous tools for in-situ studies

have used computational linguistics to draw inferences. However, there is no panacea for cross-domain linguistic analysis. Polysemy and sentence parsing issues make a completely automated text analysis tool a considerable challenge.

Developments in visualization techniques have opened up another dimension in text analysis: visualizing text data. Basic text visualizations include frequency-based word clouds such as Wordle[4], keyword in context representations [79], and lexical dispersion plots [80]. More sophisticated visualizations involve a degree of aggregate representation, or representation of metadata. The Word Tree [81] is an example of the former, with its aggregation of concordant terms to form a 'tree' of words or phrases, scaled by occurrence. Parallel Tag Clouds [82] is an example of the latter, with tag clouds in the form of parallel axes to show relationships between documents.

Representations for document content overview include Arc Diagrams [83], which represent document structure visually, or in semantic form as in Docuburst [84]. Such semantic bird's-eye visualizations work well when combined with the more detailed keyword visualizations for an effective combination of overview and detail.

---

[4]http://www.wordle.net

3. UNDERSTANDING BRAINSTORMING THROUGH TEXT VISUALIZATION

The balance between creative thinking and analytical thinking in design has been emphasized by many [9, 85, 86]. Over the last thirty years, several methods have been proposed that enable designers to achieve this balance, the most famous being Pugh's method of controlled convergence [4]. An early graphical model to this end was provided by Laseau [87, p. 91], who illustrated his model as a combination of two funnels, an expanding funnel of elaboration or "opportunity-seeking", and a contracting funnel of reduction or "decision-making". Recent years have seen developments in the study of cognitive aspects of the design process from the point of view of design education. Content analysis and protocol analysis have become widely-used methods in such studies. Studying the effects of design processes in an educational setting has two advantages: the larger body of student designers improves the robustness and repeatability of a study, and the results of the study could potentially have a more direct impact on design education. However, the large volume of textual data generated in such studies makes it a challenge to identifying critical information.

Automated content analysis software tools have significantly aided in the study of design processes in the recent past. However, they suffer from the lack of domain knowledge and insight that a human expert can provide. In this chapter, we report on the use of text visualization techniques that help in gaining insights and identifying relevant patterns from the results obtained through a content analysis software. We motivate our approach with the observation that examining overall patterns in data aids us significantly in identifying interesting and relevant details concerning specific contexts in the data. We use the proposed approach to study the effect of adopting Laseau's "design funnel" of alternating divergent and convergent design processes among student teams in a toy design course, and compare it to student teams that follow a free brainstorming process. We demonstrate the application of lexical dis-

persion plots and text concordances as a means to further examine the output of a conventional content analysis tool, and use these techniques to separate patterns from anomalies. We identify cases of concept consistency across teams using the dispersion plots, and identify cases of multiple word senses through text concordances. Finally, we present insights that were obtained through these visualizations and propose contexts for further studies of the data.

## 3.1   Comparing Brainstorming Strategies: An In-Class Study

Our study was conducted in the context of a toy design course offered as a senior elective to undergraduate engineering students. The course includes lectures that all students attend at the same time, and lab sessions, where the class is split into two batches. This batch assignment is based on a first-come, first-served basis at the time of enrollment. We used this division of batches for our between-groups study. The course includes a team project to design, model, and prototype an action toy. Students are split into teams of 4 for the project, for which we performed a random assignment to reduce the chances of performance outliers. Before they begin this project, the students are taught idea generation techniques like SCAMPER and "combining things" [88, p. 72, 332], trained on sketching as a way of visual thinking [89], and given an understanding of play value [90]. Equipped with these tools and techniques, the students are expected to start their project with a brainstorming session. We conducted this brainstorming session under a controlled environment in a classroom, where we gave one batch guidelines for "free" brainstorming, and guidelines for an alternating divergent-convergent brainstorming to the other. We then recorded the team discussions and the concepts generated in the session.

### 3.1.1 Participants

We recruited 70 participants (68 male, 2 female), who formed the entirety of the mentioned toy design course. All participants were undergraduate engineering students in their senior year.

### 3.1.2 Experiment Design

We used a lab session to conduct the brainstorming, so we could perform a between-groups study, with the groups comprising of the two lab batches. By conducting both sessions with a space of only ten minutes between them, we reduced the possibility of one batch influencing the other. We conducted the session in a room where the seating arrangement was modified to allow students in each team to sit facing their teammates, and each team was provided with a tack board for putting up their sketches or notes for discussion. Teams brought their own sketching instruments, and were provided with sheets of paper marked with their team numbers.

### 3.1.3 Tasks

We gave both batches the same problem statement: "Design a toy that exhibits a creative, non-trivial motion". Both batches had the same required outcome: a perspective sketch of their final concept, with annotations and notes describing the toy and the target customer, and added sketches of the concept's salient features. We gave both batches 90 minutes to brainstorm and arrive at this final concept. Each team was then provided with different procedures to follow:

**Batch 1** teams were provided with sheets of paper marked with their corresponding team numbers, and were allowed to brainstorm with relative freedom. They were asked to generate (and sketch) multiple ideas and put them up on their board. They were then to discuss and develop the ideas, shortlisting the promising ones and finally select one concept and develop it to detail.

**Batch 2** teams were provided with two kinds of sheets of paper: sheets marked with red dots, and sheets marked with blue dots. All sheets of paper were marked with the corresponding team numbers. The batch was told to follow a two cycles of alternating divergent and convergent brainstorming. They were given a slightly different directive: to brainstorm and generate concepts (10 more more was the preferred number) on the sheets marked with red dots, and, like Batch 1, put them up on their board. They then had to discuss and select the most promising ideas (less than 10), and mark them accordingly. They then were required to brainstorm further on these ideas, and use the concept of "crossing products" to take features from the selected ideas and develop them into more concepts. This round of concepts were sketched on the sheets with blue dots to distinguish them from the earlier ones. Finally, they were to select the concept that held the most promise.

After the session, all the teams were given three days to prepare a report that reflected on the process they followed and the product that they developed. This was to include the process they followed to generate ideas, how they balanced quantity and quality, and the rationale they used to select ideas that they found promising. They were also asked to include their feedback on the process they followed, and then discuss aspects that they found effective and those they found ineffective. The product section of the report was to include the number of initial concepts generated, a description of the final concept, its play value, and its features that customers would value. In addition, the teams were also to identify the underdeveloped features of their concept and explain what they would do to improve them. Finally, they were asked to list other concepts that they had liked, but had discarded in favor of their selected concept, and explain why.

### 3.1.4 Data Collection

Our intent was to study three main aspects of the brainstorming process, namely (1) concept discussion, which focuses on the conversations that occur within teams

during brainstorming, (2) concept representation, which focuses on the sketches and annotations that the teams generate during the session, and (3) concept reflection, which focuses on what the students think in retrospect about both the session and their concept. To this end, we made three corresponding categories of data:

- **Concept Discussion:** We recorded the brainstorming discussions of each team on individual audio recorders placed on their desks. Each individual audio recording was then transcribed, into text documents. A total of 41690 words were obtained as conversation records from the Free batch, and 40170 words from the D-C batch. In addition, two teams from each batch were arbitrarily chosen for video recording as well.

- **Concept Representation:** We collected and scanned all the sketches made by the teams, and further transcribed all sketch annotations into a text file for every team. A total of 1580 words were recorded as sketch annotations from the Free batch, while the D-C batch used 1470 words in total.

- **Concept Reflection:** We collected soft copies of the reports the teams prepared on their process and their product.

We tagged all the above information with the corresponding team names. Data from teams in batch 1 (free) were then combined to one text corpus each for concept discussion, concept representation, and concept reflection. A similar process was carried out for data from teams in batch 2 (D-C) as well.

### 3.1.5  Analysis

We subjected the text corpora from concept discussion and concept reflection to a (textual) conceptual analysis, is based on the frequency of certain word co-occurrences, and a relational analysis, based on relationship between these textual concepts in the structure of the text. We adopted a visualization pipeline as shown in Figure 3.1 that consisted of three main representations:
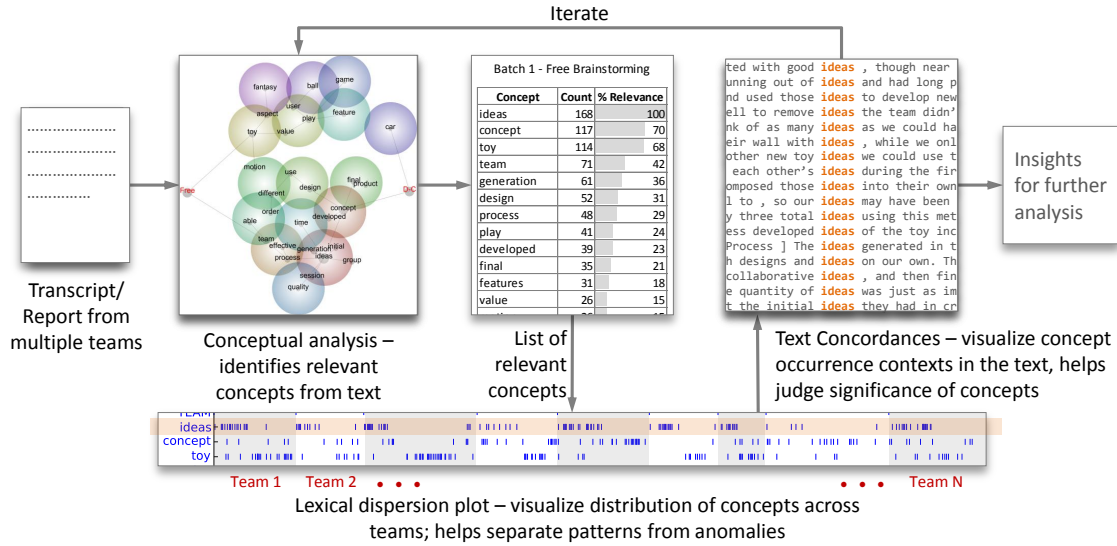
Figure 3.1. Chart showing the analysis performed on text from the discussions, representations, and reports from brainstorming sessions.

**Concept Maps:** We used Leximancer [91] to conduct the conceptual and relational analysis, the results of which are then displayed as a "concept map". Leximancer uses computational linguistics and machine learning to extract "textual concepts" based on word co-occurrence statistics. These concepts are ranked based on their relevance to the most occurring concept, as shown in the tables on the left and the right in Figure 3.2. The most occurring concept thus has a relevance of 100%. The relational analysis uses a concept-mapping algorithm based on a variant of Chalmers & Chiston's spring-force model [92] to identify relationships between these textual concepts. The output is the concept map, which consists of a graph with nodes as concepts and edges as relations between concepts. The concepts are further clustered into groups called "themes". The concept map represents (1) *concept frequency* through brightness of a cluster of concepts, (2) inter-concept *relative co-occurrence frequency* through the intensity of edges connecting related concepts, (3) total concept connectedness through hierarchical order of appearance on the map, and (4) a representation of direct and indirect inter-concept co-occurrence through proximity

| Batch 1 - Free Brainstorming | | | | Proximity Analysis (between batches) | | | | Batch 2 - D-C Brainstorming | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Concept | Count | % Relevance | | Concept | Count | % Relevance | | Concept | Count | % Relevance |
| idea | 308 | 100 | | idea | 597 | 6 | | idea | 287 | 100 |
| toy | 228 | 74 | | toy | 372 | 3 | | car | 214 | 75 |
| ball | 154 | 50 | | car | 362 | 3 | | ball | 194 | 68 |
| water | 132 | 43 | | ball | 347 | 3 | | cool | 176 | 61 |
| draw | 127 | 41 | | cool | 299 | 3 | | draw | 143 | 50 |
| car | 123 | 40 | | draw | 272 | 3 | | water | 130 | 45 |
| things | 109 | 35 | | water | 262 | 2 | | things | 129 | 45 |
| time | 104 | 34 | | use | 247 | 2 | | use | 125 | 44 |
| motion | 101 | 33 | | shoot | 245 | 2 | | toy | 121 | 42 |
| goes | 98 | 32 | | things | 238 | 2 | | wheels | 105 | 37 |
| different | 91 | 30 | | wheels | 218 | 2 | | game | 103 | 36 |
| use | 91 | 30 | | game | 191 | 2 | | play | 99 | 34 |
| stuff | 88 | 29 | | different | 185 | 2 | | different | 92 | 32 |
| top | 82 | 27 | | play | 184 | 2 | | guess | 91 | 32 |
| probably | 78 | 25 | | motion | 183 | 2 | | thinking | 90 | 31 |
| game | 75 | 24 | | goes | 173 | 2 | | work | 86 | 30 |
| spring | 72 | 23 | | work | 172 | 2 | | stuff | 83 | 29 |
| thinking | 70 | 23 | | stuff | 171 | 2 | | motion | 79 | 28 |
| play | 67 | 22 | | time | 169 | 2 | | take | 77 | 27 |
| guess | 67 | 22 | | thinking | 160 | 2 | | guy | 76 | 26 |
| boat | 66 | 21 | | guess | 158 | 1 | | goes | 75 | 26 |
| hit | 65 | 21 | | take | 131 | 1 | | time | 65 | 23 |
| air | 61 | 20 | | top | 129 | 1 | | people | 59 | 21 |
| inside | 60 | 19 | | hit | 128 | 1 | | air | 54 | 19 |
| ones | 55 | 18 | | probably | 128 | 1 | | | | |
| whole | 41 | 13 | | air | 116 | 1 | | | | |
| cause | 41 | 13 | | guy | 116 | 1 | | | | |
| | | | | ones | 112 | 1 | | | | |
| | | | | trying | 108 | 1 | | | | |
| | | | | inside | 102 | 1 | | | | |
| | | | | spring | 102 | 1 | | | | |
| | | | | feel | 99 | 1 | | | | |
| | | | | whole | 72 | 1 | | | | |

**Legend**
| Common concepts |
| Unique to D-C |
| Unique to Free |
| Unique to common corpus |

Figure 3.2. Concepts extracted from batch-wise brainstorming transcripts, with concepts unique to each batch highlighted. The proximity map on top shows concepts clustered nearer the groups they occur more in. Warmer hues of concept clusters indicate higher frequency of occurrence.

of the concepts on the map. Further, Leximancer allows manual tagging of a text corpus, which is then represented on the concept maps as a category [93]. The final concept map then clusters the discovered textual concepts around the specified tag categories, as shown in Figure 3.3.

Figure 3.3. Proximity diagram for concept representation. The grey circular sectors are centered on the free and d-c categories, and are of the same radius. The higher clustering around free indicates a higher number of concepts from the corresponding teams.

However, in this representation, the discovered concepts do not characterize the whole text: they are specific to the tag categories and do not cover the major themes of the whole data set. Relevance here is then calculated as the probability of a concept occurring in a specified category, i.e., one of the two batches in this study. Thus the concept with the highest occurrence in both categories does not have a relevance of 100%, as shown in the central table marked "Proximity Analysis" in Figure 3.2. This visualization of the concept map in relation to the tagged categories is then useful for a comparative analysis, in our case, between batch 1 and batch 2.

**Lexical Dispersion Plots:** The text corpus for each batch contains content from nine teams. Thus, it becomes important to examine the distribution of the most occurring concepts. A concept cannot be said to characterize a batch unless it is fairly uniformly distributed across most teams in the batch. While it is possible to construct proximity maps with each team as a category to identify if some of the concepts are skewed toward certain teams, the human readability of such maps becomes difficult with increasing number of teams. A *lexical dispersion plot* [80, p. 120] offers a more visually preattentive way to represent concept distribution. The plot indicates locations in the text corpus where a particular concept occurs, and therefore can offer the following visual representations:

1. In analysis of transcribed meetings and discussions, it can show temporal patterns in the occurrences of concepts within and among teams.

2. It provides a visual representation of document structure with respect to the concepts of interest.

3. It provides a means of visually comparing concept occurrence counts, which can be used to identify common or unique ideas among teams.

The lexical dispersion plot in our study was generated using the Python Natural Language ToolKit (NLTK) [94].

**Concordances:** In addition to consistent distribution of a concept's occurrences across teams, its significance is also determined by its usage in context. For example, frequent occurrences of the word "feel" in a design discussion could indicate that participants are discussing user experience, or are using the word to voice their opinion. A *Key Word In Context* (KWIC) concordancing program [79, p. 31] is used to display all occurrences of a word of interest, along with their surrounding words to establish context. The word in question is aligned vertically to provide a visual reference for the user to identify patterns in the preceding and succeeding words. Figure 3.5 shows
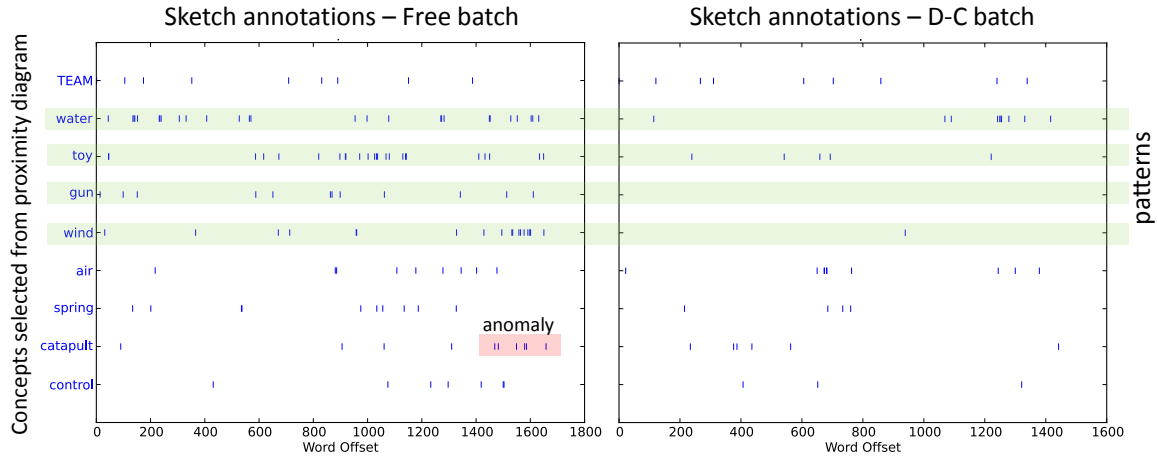
Figure 3.4.   Text dispersion plots of selected concepts from sketch annotations aid the identification of patterns and anomalies.

an output of such a concordancing program. In addition to providing a means to visually determine the context of a concept, concordances are also useful in identifying compound concepts: groups of words that carry a meaning different from that of the words that make it up, such as "remote control". Concordances were thus used to (1) further vet the concepts for significance, and gain insight into the sense in which the concept was used (2) To disambiguate word senses visually, and (3) to identify compound concepts and further iterate over the conceptual analysis with Leximancer.

## 3.2   Results and Discussions

We sought to study the effects of two different approaches to brainstorming among student teams, and to identify any patterns in their discussion, representation, and reflection that may be attributed to the process they followed. Our results are thus structured under the three heads of Concept Discussion, Concept Representation, and Concept Reflection. Under each of these heads, we discuss our findings from the content analysis, and explain the visual representations generated using the lexical

dispersion plots and KWIC concordances, which allowed us to drill down to details and gain further insights.

### 3.2.1  Concept Discussion

The conceptual analysis of each batch revealed 27 concepts for the Free batch, and 24 concepts for the D-C batch. The proximity analysis of both batches revealed 33 concepts, 7 of which were unique to the Free batch and 5 to the D-C batch. A complete list of the concepts color coded according to uniqueness to the two batches is shown in Figure 3.2.

Concepts closest to the "Free" and "D-C" categories on the proximity map were then considered as potentially significant concepts. Dispersion plots of these selected concepts in their corresponding text corpora revealed that "top", "probably", "time", and "motion" were both relevant to and consistent within the Free batch. Similarly, "guy", "cool", "game", "play", "take", and "draw" were selected as potentially significant concepts for the D-C batch.

Text concordance of these concepts revealed no unique use of "probably", "time", and "motion". However, there were 8 instances of the mention of "top" as a toy by four teams in the Free batch, but none in the D-C batch. The D-C batch showed relatively higher incidences of the word "guy" in the specific context of an action figure, and lower occurrences of the term "play value" than the Free batch.

The analysis of concept discussion does not reveal much information, and one of the main reasons can be seen in the ranked concept list from the proximity analysis, shown in the central table of Figure 3.2. The percentage relevance of the top concept, "idea", is 6%. As mentioned earlier, this number indicates the percentage incidence of the concept in the text corpora from either of the batches. A cursory reading of the transcripts shows a significant amount of banter among team members, which dilutes the occurrences of discussions meaningful to the brainstorming exercise. A possible solution could be to manually excise non-relevant discussions from the transcripts,

but this may result in the loss of information that could potentially contain records of "chance encounters" [95]. A more challenging but fruitful approach would be to manually identify the presence of such instances of creativity that come from chance encounters with new content, stories, or recall of past experiences. The visualization pipeline used in this work may not be sufficient for this task, but a more hierarchical visualization of broad-level concepts from text [84] could perhaps aid discovery.

### 3.2.2  Concept Representation

As mentioned in section 3.1.5, the text corpora from concept representation consisted of annotations made on sketches. These were isolated words and phrases, and the proximity of these words on the transcribed text file might have meaning only if they were from the same sketch. Thus the results of the conceptual analysis had to be interpreted with care. The concept clusters were unlikely to have any thematic meaning, but the proximity of the concepts to the manually-assigned categories of "Free" and "D-C" was still meaningful. Figure 3.3 shows the proximity diagram for concept representation, with circular sectors of the same radius drawn centered at the "Free" and "D-C" categories to show the variation in clustering density. It was immediately evident from the diagram that concepts were more heavily clustered around the Free batch, and those clustered around the D-C batch occurred less frequently. This was also evident in an examination of the ranked concepts from individual analyses of text from each batch: concepts unique to the Free batch were more frequent in the the combined proximity analysis than those unique to the D-C batch. Lexical dispersion plots of selected concepts helped discard the concepts "catapult" and "parts" from both batches, while "water", "toy", "gun", "wind" were selected for examination in the Free batch. "Motor" and "body" were concepts relevant to the D-C batch that were selected for further analysis. Some of these plots are shown in Figure 3.4.

A concordance analysis showed that "water" continued to be of significant importance: eight teams from the Free batch discussed ideas for water toys or games, while
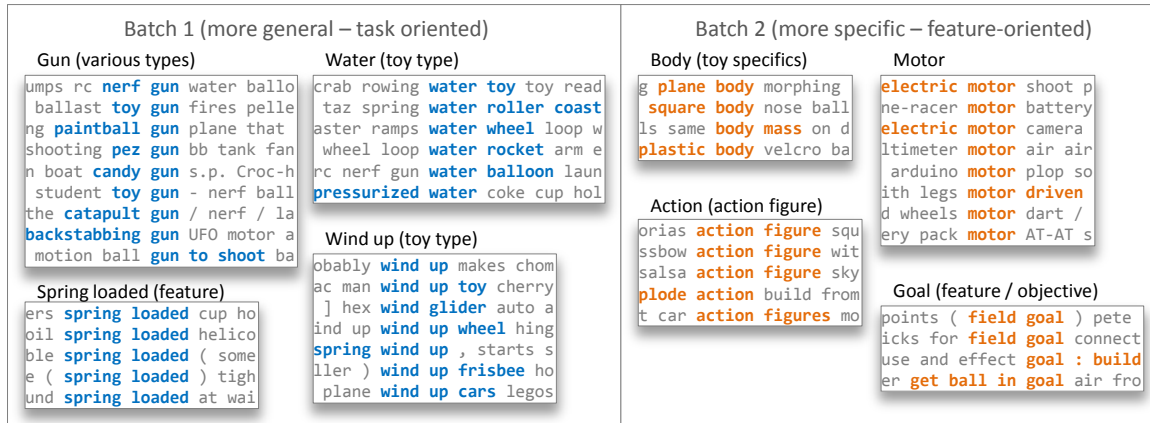
Figure 3.5. Text concordances from concept representations. Examining the usage of the identified concepts helps understand their significance.

only three teams discussed water-based ideas for toys. The word "wind" was used in the sense of "wind up" (wind-up toys) in almost every occurrence in the Free batch, while it appeared only once in the D-C batch, in the context of "wind powered". The word "toy" was five times more frequent in the Free batch than the D-C batch. In terms of sketch annotation, this could be significant: it could mean that the Free batch concentrated more on types of toys, without getting too much into detail, while the D-C batch showed a higher tendency to get into detail. This inference is further supported by the higher frequency of use of the terms "motor" and "body" in the D-C batch. Text concordances of both concepts revealed that they referred to components of the toys, thus showing a greater focus on how they worked. Some representative concordances are shown in Figure 3.5.

The inferences from the sketch annotations section were more or less what was expected: the alternating divergent-convergent process requires, in the second cycle, to concentrate on features of the design, and helps the designers focus on detail. The next question that came up was, which approach is better? The Free brainstorming batch came up with more kinds of toys, which is beneficial for initial idea generation,

while the D-C batch focused on details, which is beneficial for better quality and evaluation of ideas. The question of duration also comes into the picture: did the D-C batch focus on details "too quickly"? Is it better to use an alternating divergent-convergent brainstorming session in one go, or is it better to have a gap between each cycle to allow for reflection? These could form the topic of future research.

### 3.2.3 Concept Reflection

The proximity diagram for concept reflection, shown in Figure 3.6 indicates a higher incidence of shared concepts between the two batches, shown in the higher clustering in the middle of the "Free" and "D-C" categories. As with concept reflection, circular sectors of equal radii centered on the category nodes and overlaid on the diagram helped isolate concepts that were closest to each category. A lexical dispersion plot of the selected concepts helped eliminate "car", which was concentrated on two teams in the D-C batch, and also helped draw attention to "team", which had a higher occurrence in the Free batch, and "group" and "feature", which seemed to have a higher occurrence in the D-C batch. While "toy" had high occurrences in both batches, it seemed relatively higher in the D-C batch. An interesting observation here was that the occurrences of "toy" were concentrated in the latter half of each team's report. This was consistent with our requirement which stated that the reports should have a separate sections for the process and the product: "toy" would occur more in the section discussing the product. This structure being revealed by the lexical dispersion plot highlights its characteristics discussed in section 3.1.5.

Text concordances of the selected concepts revealed that the D-C teams discussed more about the features of the toy than the Free batch, which is consistent with the results of the concept representation. The use of "group" and "team" were revealed to be synonymous to each other, and combining the two revealed that reports from the Free batch discussed more about their teams. Closer examination of the occurrence patterns of "group" and "team" did not show a clear concentration of occurrences
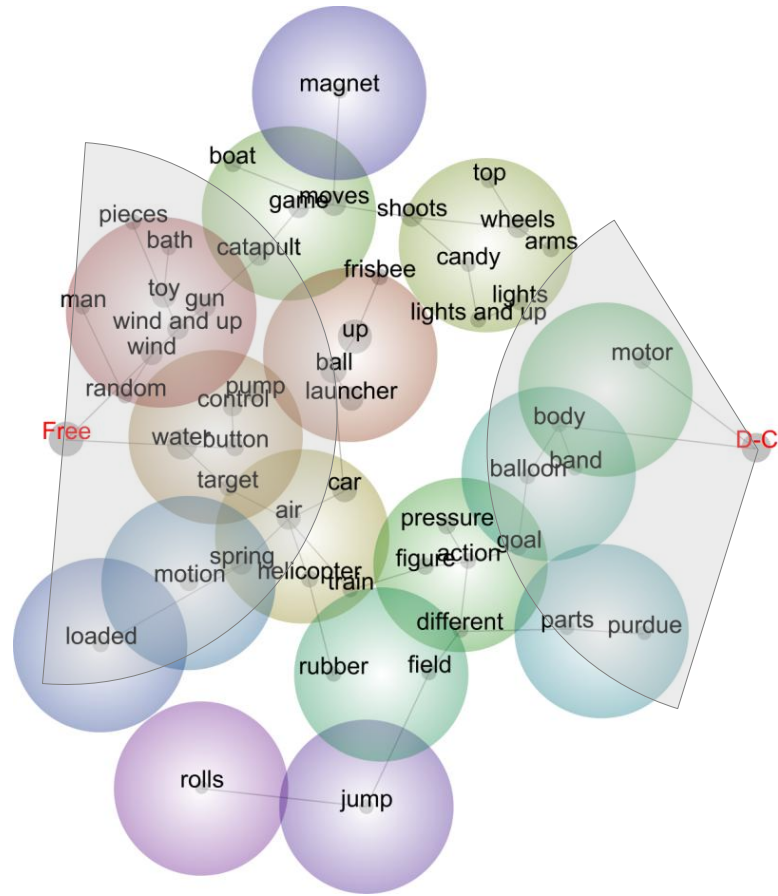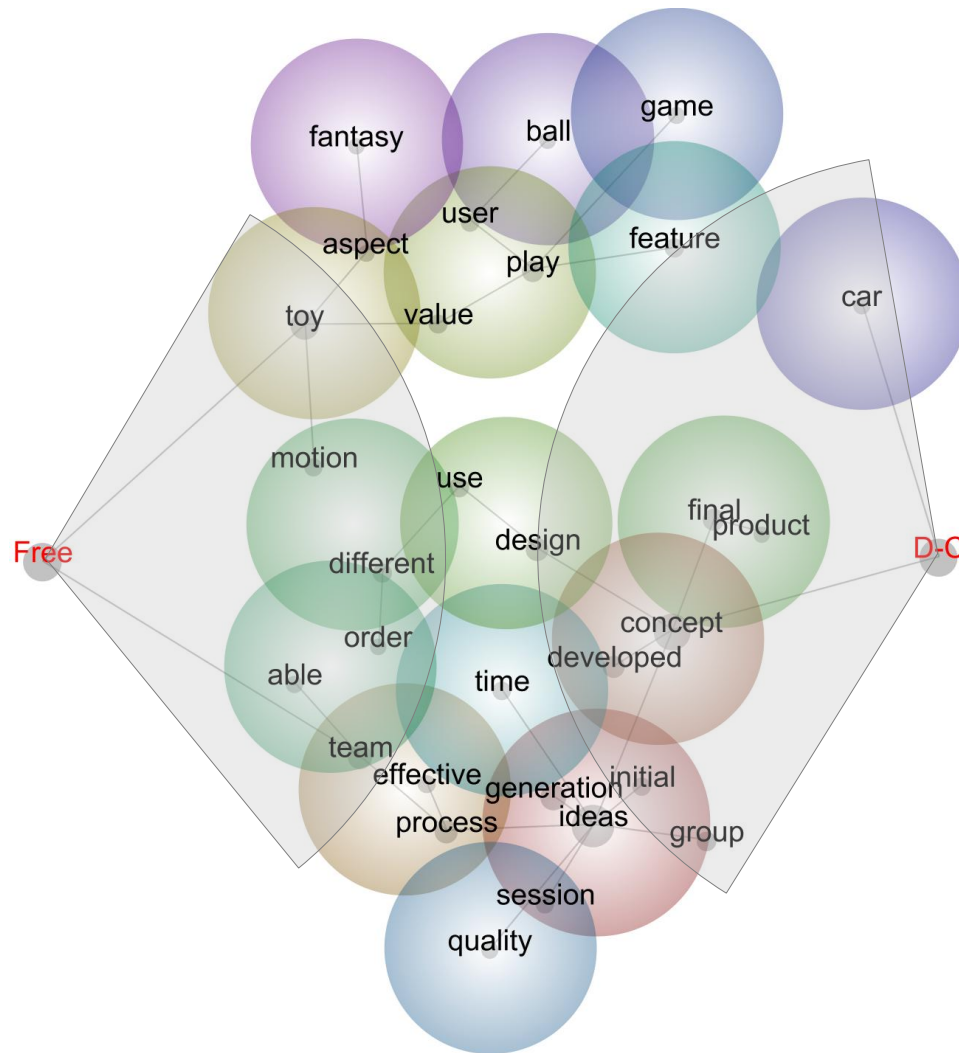
Figure 3.6. Proximity diagram for concept reflection. the grey circular sectors are centered on the free and d-c categories, and are of the same radius.

in either section of the report. The D-C batch also had higher instances of the term "final" which was used in the context of "final concept" or "final design" in both batches. This could be due to the emphasis on the selection process that is placed due to the alternating divergent-convergent processes, and merits further study.

A concordance of all concepts from the proximity diagram also revealed that the D-C batch showed 25 instances of the term "game", while the Free batch showed 5. A

preliminary inference that could be drawn from this is that the D-C batch tended to reflect more on the use of their toy, and how it would be played with. However, more teams from the Free batch reported on the play value [90] of their toy, while the term "customers will value" was used uniquely (but not very frequently) in the reports from the D-C batch. This could reinforce the inference of the D-C teams reflecting more on the user experience. A further study on concept discussion focusing more on terms related to the experience of using the toy could reveal more about how the two processes might influence consideration towards the user experience.

## 3.3  Conclusions and Future Work

In this chapter, our key contribution is a visualization-driven approach to augment automated content analysis tools in order to enable a domain expert to observe patterns, identify anomalies from an overview of data, while also enabling a closer examination of data in context to gain deeper insights. Our primary motivation was to enable the analysis of large bodies of unstructured text obtained from brainstorming sessions of student groups. Our primary goal was to understand the difference between the effects of two distinct modes of brainstorming, embedded in the context of a toy design course. To this end we concretely investigated our approach through the study of discussion, representation, and reflection of concepts generated during two independently conducted sessions: free brainstorming, and alternating divergent-convergent brainstorming. The visualization techniques used were lexical dispersion plots, which helped evaluate the consistency of concepts identified by the content analysis tool, and text concordances, which helped observe the use of these concepts in the original text body. These visualizations helped separate the relevant concepts from anomalies, and verify the significance of the usage of these concepts, leading to insights that would guide further studies.

We observe that the study of brainstorming design processes can be as subjective as the processes themselves due to the inevitable involvement of several modes of

communication and expression distributed across several individuals in a group. This poses a very unique challenge in terms of both the data acquired from the brainstorming sessions in different forms as well as the final concepts generated which cannot be objectively evaluated due to absence of details. With this in view, we appreciate the importance of more in-depth studies using our proposed approach towards conclusive and insightful results. Our next steps would involve a more focused analysis of team orientation towards trends observed in this study such as user experience and reflection on team processes. We posit that a protocol analysis of the video recordings would reveal behavioral aspects of participants and their relation to tendencies observed in this study. We believe that a visualization-driven approach has immense potential to address many of the issues related to content analysis in design studies and must be investigated and developed further.

## 4. COMPUTER SUPPORT FOR COLLABORATIVE CONCEPTUAL DESIGN

Computer support tools for design have made inroads into most design stages, from requirements gathering to detailed design. However, the (paper) design notebook remains the primary tool of the designer for sketching, recording observations, and making notes. This is partly due to the user's training—most people have sketched on paper since childhood—and the affordances of the medium—a notebook is easily carried around, and can store sketches as well as photos or clippings pasted to it [9]. Hong et al. [96] observe that "Where paper notebooks are used by designers to capture work for personal reference, an electronic version has the potential of being accessed by team members across space and time". Moreover, digital support for sketching has the potential to aid design collaboration, including duplication, modification, and archival of concepts. The value of digitizing the design notebook was recognized early on [97], but it was not until the rise of direct-touch smartphones and tablets that the digital design notebook became a realizable vision.

Dorta [13] argues that computer support tools for ideation should "augment instead of imitate": they should use the computational and interactive power to augment traditional design methods and tools, rather than imitate them. Collaboration adds a level of complexity to concept generation and progressive development: an effective communication and iteration mechanism is necessary to effectively utilize ideas generated by individual team members. Digital support for sketching should thus augment the intuitiveness of sketching on paper with the capabilities of storage and communication offered by computers.

With this in mind, we used skWiki [98] a web-based application framework for collaborative creativity that would couple the interactions afforded by hand-held personal computing devices, with the collaborative power of a web-based platform. skWiki allows users to sketch their ideas, add text annotations and images, and share

them with collaborators (Figure 4.5). The system also features a representation of a team's collaboration: a "path viewer" (Figure 4.1) that graphically and transparently represents all team activity, tracking all edits, modifications, and duplications of sketches. skWiki was developed to support digital creativity in collaborative teams, both co-located as well as distributed.

In this chapter, we briefly outline the requirements of a collaborative sketching tool, and describe how skWiki addresses these requirements. We then present two evaluations of skWiki conducted through user studies of conceptual design processes. The first study uses a constrained brainstorming process to identify the operations that are possible using the "path viewer", and the second uses a longitudinal study of a collaborative conceptual design process to provide a more higher-level understanding of how skWiki provides higher potential for creativity in collaborative design.

## 4.1 Collaborative Sketching Requirements

The paper-pen paradigm enjoys its position as the medium of choice among designers owing to its flexibility and ease of use. However, digital media is gaining ground through its advantages of better storage and sharing models, with its interfaces rapidly evolving to bridge the gap between the two. In this section, we define a sharing model on a digital platform in terms of four aspects required for designers, and explain the skWiki interface on the basis of these aspects.

A1 **Storage and retrieval:** This aspect represents the variety of sharing operations possible using a digital platform. The fundamental operations of storage and retrieval include:

**Save**: Saving a new concept (or sketch) to a storage space. This be a private space (handheld device) or a shared space (network/ cloud storage).

**Modify**: The modify operation extends the *save* operation, and involves updating (edit + rewrite) a saved sketch on the storage space.

**Branch**: A branch operation occurs when a saved sketch (representing one or more concepts) is modified and stored as a new sketch in the storage space, commonly seen in code versioning systems.

**Merge**: In most concept design processes, coming up with one or more final ideas (convergence) is as important as the divergent processes. Also, concepts can span over multiple sketches and thus requiring a merge operation that can merge the sketches together into one.

Some of these operations are possible with pen and paper. However, they require a lot of manual effort and use of scanners or copy machines, and lack the level of ease and control possible in digital platforms.

A2 **Atomicity of the data:** In our context, atomicity refers to the smallest identifiable entity in the data that can stored through one transaction with the storage space. While an image can be stored on a file system, structured data in the form of text or vector representations are more atomic, and can be stored on a database. This allows for higher retrieval speeds and richer, more controllable encoding of the data. For sketches, this translates to the capture and storage of its history, at the level of individual strokes.

A3 **Collaboration representation:** Visual representations of authoring and collaboration can provide necessary group and situation awareness to the user. A graphical representation of user activity and of user contribution to a shared repository helps provide an overview of the evolution of the designs.

A4 **Capture of user intent:** This includes capturing user comments and explanations of the concept for the understanding of other users and/ or administrators. Types of capture may include video, audio, and textual methods.

### 4.1.1   Sketching Framework Used

Based on the above requirements for collaborative sketching, a web-based system called *skWiki* was developed by Zhao et al. [98]. skWiki, short for "sketch Wiki", helps manage concepts that are generated in the form of different media types such as text and sketches. It captures individual user operations such as the creation of sketch strokes or text input, and stores the operations rather than graphical entities on a database. This allows for a higher level of atomicity (A2) in the collected data, and enables flexible storage and retrieval of sketches (A1). By combining these two aspects, skWiki enables history support for stored media, allowing save, branch, and merge operations. These operations are implicit, with the user performing normal file-like operations such as save/commit, checkout, and edit.

The collaboration representations (A3) in skWiki include a path viewer and a thumbnail viewer. The path viewer is a node-link diagram representing the saved ideas as nodes, and connections between them such as branch and merge as links. The path viewer provides the ability to examine and use design sketches created by other users, modify them, and save them as new versions. Capture of user intent (A4) in skWiki happens through the implicit way of annotating the sketch. Figure 4.1 shows the user-level operations using skWiki, and its representation on the path viewer.

skWiki was built as a web application using the Google Web Toolkit[1] and runs in a web browser, requiring no additional software installation. Data in skWiki are stored as sets of operations. For example, a sketch is treated as a set of strokes, and adding a stroke becomes an atomic operation for the sketch. The operation sequence is stored in a database and it allows version management of the ideas generated, thus reducing the stored data.

To evaluate our prototype, we performed two studies. The first, a study of co-located teams on a creative task, provides a low-level understanding of using skWiki's persistent storage and the possibilities opened up by its branching operation. The

---

[1]http://www.gwtproject.org/
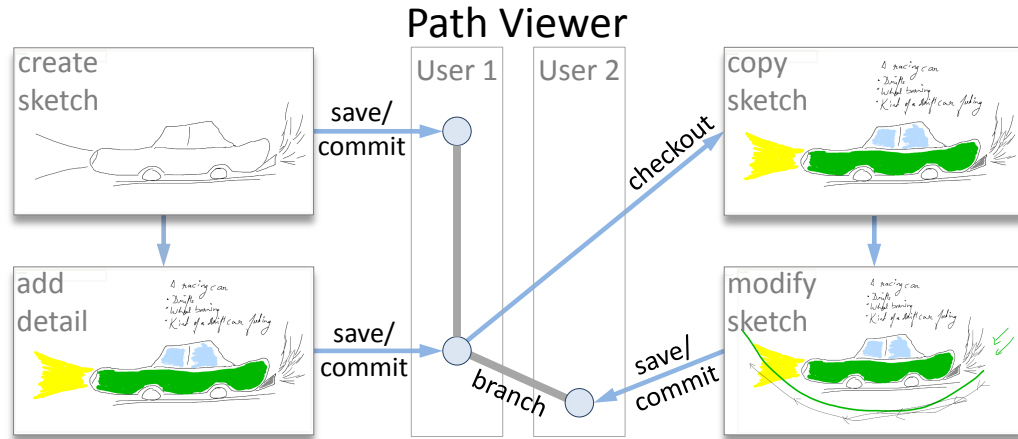
Figure 4.1. A basic representation of what the path viewer depicts. A sketch created by user 1, once saved or committed, is represented on the path viewer as a node (shown in light blue). Every time the user makes any modifications and commits the sketch, that particular version is added to the path viewer as another node. Any of these versions can be checked out by User 2 and modified. Upon committing, User 2's version appears under his ID on the path viewer, with a link to the original checked out version, called a "branch". Branching can thus be seen as checkout + modify + commit.

second is a longitudinal study of one team in their conceptual design process, as part of their course design project.

## 4.2   Web-Based Collaborative Sketching: A Preliminary Study

The first study aims to understand the value offered by skWiki's "paths" model in a collaborative conceptual design scenario. Since free-form creativity not only takes training but is also difficult to quantify, we used a form of controlled brainstorming called C-sketch [27]. C-sketch is a sketch-oriented adaptation of the more generic method of 6-3-5 brainwriting [22], contextualized in a design environment. In the C-sketch method, designers first spend 6-10 minutes sketching out an idea, and then pass it on to the next person. Each designer then spends the next 6-10 minutes working on editing or extending the design idea given by their teammate. This

process continues for 2-3 iterations, at the end of which all designs sketched initially have undergone iterative development by at least 3 people. The C-sketch workflow is shown in Figure 4.2.
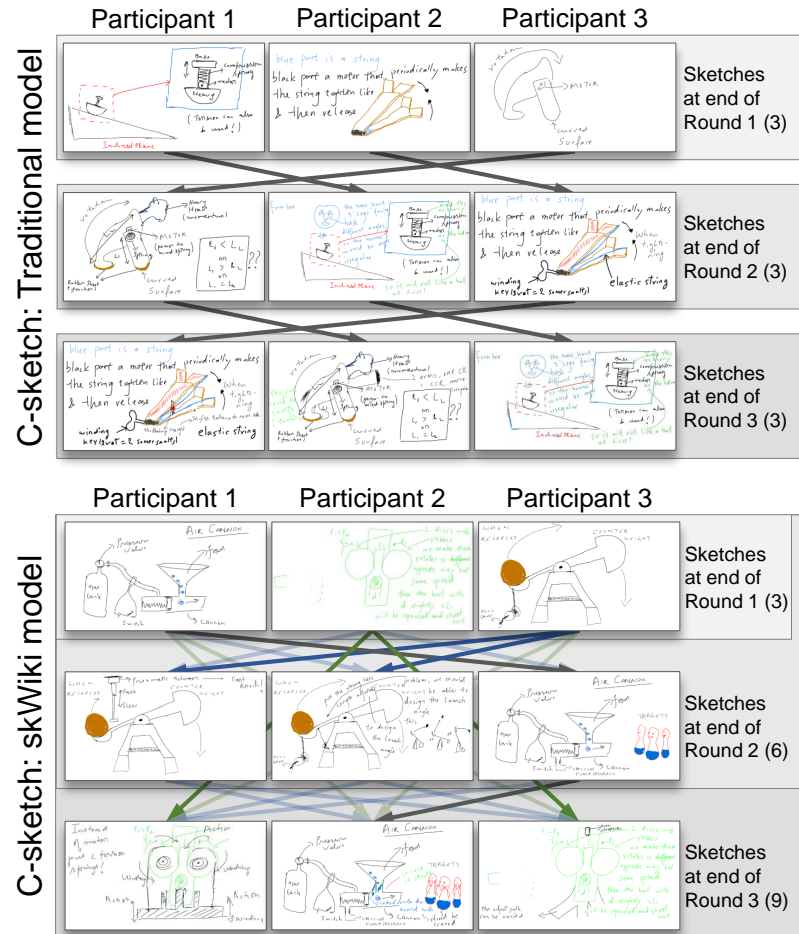


Figure 4.2. Schematic diagram of the traditional C-Sketch model (top) versus its skWiki version (bottom) used in the study. Each column shows sketches made or modified by each participant. The grey arrows show the "sketch movement" paths of the traditional C-sketch method, and colored arrows show the paths enabled by skWiki. Potential paths are shown as translucent, while actual paths taken are shown with solid arrows. The skWiki model allows duplication and multiple copies at each round (aquamarine arrows), as well as for branching from earlier states (orange arrows). The team thus has more choices at the end of each round, can access all sketches at the end of the session.

### 4.2.1 Experimental Conditions

One of the main drawbacks of methods such as C-sketch is that some promising ideas can be lost in the series of iterative edits. Additionally, at the end of the session, there are only as many ideas as there are designers, with intermediate—and potentially promising—ideas being "lost". We hypothesized that the history support afforded by skWiki, along with the option to branch out and create multiple versions of the same sketch, will provide designers with more potential sources of inspiration and development for the design problem at hand.

In order to study the effects of the affordances offered by skWiki, we involved each team in two different conditions:

- **Traditional:** The team used a traditional three-round C-sketch workflow as described above, with only sketch movement (no copying) between participants, and with only the latest sketch version available for each round.

- **Full skWiki:** Here the team used the same three-round method as above, with two main differences: (1) at the beginning of the second and third rounds, they could choose any sketch to work on except the sketch they had created in the previous round, and (2) when selecting a final design (end of round 3), they could choose sketches from rounds earlier than the immediately preceding one.

While the C-sketch method typically involves sketching on paper. our intent was not to compare paper vs. digital media. We thus used skWiki for both versions: a version without branching and history "rollback" for the C-sketch condition, and a version with both these features for the full skWiki condition. The aquamarine and orange arrows in Figure 4.2 show the branching and rollback operations respectively.

### 4.2.2 Participants

We recruited 4 teams of 3 paid participants each (11 mechanical engineering graduate students and 1 post-doctoral researcher, all male). Participants were aged between

21 and 33 years. 10 participants were comfortable with sketching, and 5 considered themselves proficient. 4 participants had prior knowledge of the C-sketch method. Participants were randomly assigned to teams based on available time slots.

### 4.2.3 Apparatus

All participants used Microsoft Surface Pro tablets, equipped with the Surface Pen for sketching and annotations. Both conditions used different versions of the skWiki interface, running on the Google Chrome browser. For the traditional C-sketch condition, the skWiki interface was provided with curbed features based on the C-sketch method, such as adopting a sketch and editing it, with no copy or history support. The skWiki C-sketch condition also required a level of feature curbing: history support was minimized to what was available at the end of each round, but not to a stage between them. These constraints helped control the experiment conditions, in addition to allowing the participants to concentrate on the method rather than spend their time on learning and remembering commands.

### 4.2.4 Tasks

The teams were assigned two tasks, one for each condition: (1) design a toy catapult with an innovative launching mechanism, and (2) design a new kind of somersaulting toy. The order of conditions and tasks were varied among the team to balance out learning effects as well as testing bias.

Each task was split into three rounds of 6 minutes each. In the first round, participants were asked to sketch one idea each for the toy, and annotate it so that their team could understand the idea without additional explanation. No verbal communication between team members was allowed during the three sketch rounds. In the second and third rounds, each participant was asked to develop or edit the sketch of another participant, without completely erasing it. In the case of the traditional C-sketch method, participants were asked to circulate their sketches clockwise to their

adjacent teammate. For the full skWiki condition, participants could choose any of their teammates' sketches from any stage, but not their own. This restriction was imposed to prevent participants from continuously working on their own idea for the duration of the session. At the end of the session, participants were asked to spend 5 minutes to discuss and select the most promising concept.

Participants were asked to respond to questions pertaining to the usefulness and ease of the methods on a Likert scale. A log of participant choices in rounds two and three in the skWiki condition was also recorded to identify cases of departure from the traditional method afforded due to the branching (creating copies) and history support (choosing a sketch from the first round during the third). Finally, each team's selection of the "most promising idea", and its corresponding round, was noted.

### 4.2.5   Results and Discussion

The branching operations performed by participants in each team for both the traditional C-sketch and the full skWiki conditions are shown in Figure 4.3. Below we discuss how features that are unique to skWiki were used by the teams.

**Cloning and History Rollback** The C-sketch flow of design follows a linear sharing model through a "passing the paper around" paradigm prescribed by the method that the participants were required to follow. For the skWiki method, participant activity shows instances of multiple copies of a sketch in every round, for every single team. Of these instances, three teams branched out (cloned) from earlier versions of their team members' sketches, made possible through the "history rollback" support. Three out of four teams selected their final design from the last set of iterations, shown by the "starred" nodes in Figure 4.3. However, it is noteworthy that one team selected a design from their second round, which would have been lost had it not been for the history rollback support.

Participant responses to survey questions support the usefulness of the branching and history rollback afforded by skWiki: of the 12 participants, 11 preferred the full

skWiki model of C-sketch. Reasons cited for the preference ranged from the ability to choose a more promising idea, the availability of a larger variety of ideas to choose from, and the ability to see more popular or "growing ideas", as one participant put it. Participants also cited the ease of collaboration as another reason for preferring skWiki. The one participant who preferred the traditional C-sketch model of sharing cited his reason as the full skWiki system allowing the designer to stick to a limited set of designs if he so chooses, as opposed to C-sketch, which ensured that everyone worked on everyone else's designs. However, all 12 participants reported that they found the option to select ideas useful.

**Path Viewer with Preview:** Recall that the traditional C-sketch path viewer for the user study was configured to show only the latest sketches of all the users at any point of time, while for the skWiki model it showed the revisions of all users that were uploaded at the end of each 6-minute session. The post-survey responses suggest that a majority of the users (average of 75%) found it easy to decide on a version to download for the next round during full skWiki model even though more versions are shown than in the traditional C-sketch. For the full skWiki sharing model, we anticipated a decrease in the ease of browsing and choosing ideas in later rounds owing to an increase in the number of ideas to choose from. Participant responses, however, were mixed: 7 out of 12 participants mentioned that it was easy for them to choose an idea to work on in the second round, while 1 participant reported finding it difficult. 4 participants were undecided. Surprisingly, the number of participants who found it easy to choose ideas increased for the third round to 9 participants, while 3 participants found it difficult. Interestingly, this increase was accompanied by a mix of transitions: all participants who found it difficult to choose in the previous round found it easier to choose in the last round, whereas 3 participants who found choosing in the second round easy, had the opposite experience in the third round. The increase in participant ease could be explained by a greater familiarity with their team member's designs by the third round, assuming changes are clear in the thumbnail view. A more complex design change, however would entail checking out
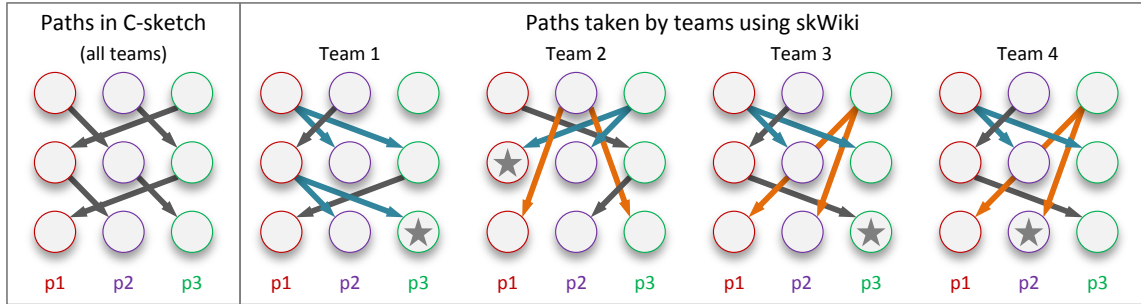
Figure 4.3. Comparison between paths taken in the C-sketch model (left) and skWiki (right) in the user study. Each gray node represents a sketch by a participant (labeled as p1, p2, p3) at the end of every round. Standard "passing on a sketch" operations are shown as gray arrows, branching to create multiple copies is aquamarine, and branching from history is orange. Stars indicate a sketch was selected as the best design.

the sketch and examining it closely, a process that becomes more tedious the more choice one has. This is echoed by the participants: some suggested using larger previews, or larger thumbnails with the facility to flip through them easily.

From a methodology point of view, it is premature, based on this study alone, to conclude that more choice for the designer is better. In fact, allowing the designer, especially a design engineer, to freely choose a design could lead to fixation, as engineers tend to favor previously encountered designs or designs they developed themselves [99]. However, the purpose of skWiki is not to merely provide choice, but to preserve every stage of work as well as to allow for potential branching (cloning) of ideas at every such stage. In the context of the C-sketch method, Shah et al. state saturation—participants feeling "that they could no longer contribute to the idea generation process" [27, p. 191]—as one of the issues of their method. With skWiki, this saturation can be delayed since each designer can have the opportunity to work on every other designer's *initial* design, without incremental additions or modifications of features done by other designers, thus geometrically increasing the number of potential iterations. Additionally, designers can return to the problem days later

and pick up where they left off, owing to the persistence of every state of their design on the server. Finally, as seen with Team 2 in Figure 4.3, skWiki preserves promising ideas that would otherwise be lost to further iterations.

## 4.3   Web-Based Collaborative Sketching: Longitudinal Study

We performed a follow-up longitudinal study with skWiki in a classroom scenario, where it was used by a team in an undergraduate toy design course over a period of 19 days. With this study, we sought to answer the following research questions:

Q1 How does the availability of "shared design notebooks" aid the concept generation and development process? In other words, does skWiki aid lateral and vertical transformations [17] of ideas between designers in a team?

Q2 As a research tool, what insights does the path viewer representation offer about the design process itself?

Note that the study was meant to examine how digital creativity platforms affect the design process, and not directly compare the performance of skWiki with that of a traditional ideation process using pen-paper as a medium. Such a comparison is premature as sketching performance on direct-touch tablets is yet to meet that of paper [38], and comparing design processes between the two media would require conflating the effects of sketching performance as well.

### 4.3.1   Sketches as Idea Transformations

Ferguson [100] identifies three kinds of sketches that designers make: the *thinking sketch*, for nonverbal articulation of their ideas, the *prescriptive sketch*, for detailing an idea to a draftsman or a CAD modeler, and the *talking sketch*, as a means of collaboration and communication with other designers. Of these, the thinking sketch and the talking sketch have a fluidity and ambiguity to them that renders them
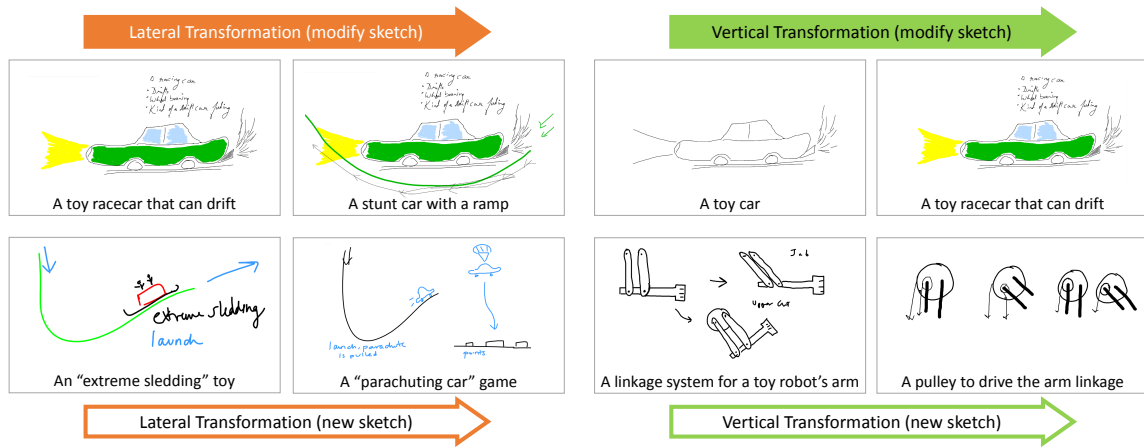
Figure 4.4. Lateral and vertical transformations observed in the study. Lateral transformations can occur through modification of an existing sketch, or through the creation of a new sketch inspired from an existing sketch, to create a new idea. Vertical transformations can occur through adding details such as dimensions, annotations, or features to an existing sketch, or through a new sketch showing a different detail, to develop an existing idea.

mutable—an important attribute for divergent thinking. They are thus of particular importance to those studying the conceptual design process. Specifically with respect to design sketches, Goel [17] identifies two main kinds of transformations:

- A **Lateral Transformation** involves a change or movement from one idea to another idea that is related to, but distinct from, the original. Lateral transformations are essential for divergent thinking: generating ideas, exploring alternatives, and expanding the problem space.

- A **Vertical Transformation** involves detailing an existing idea by adding annotations, dimensions, or features. Vertical transformations can be seen as a form of convergent thinking, to help refine an idea and develop its details.

Lateral and vertical transformations are thus signs of divergent and convergent thinking, which in turn enhance the creativity and quality of the final design [9].

Each of these transformations can take two forms (see Figure 4.4). A lateral transformation can be performed by either (a) referring to, or being inspired by an existing sketch, and creating a new version that denotes a design alternative, or (b) modifying an existing sketch to create an alternative. Similarly, a vertical transformation can be performed by either (a) creating a different view of the object denoted in the original sketch to show more detail, or (b) embellishing the existing sketch with dimensions, annotations, and additional features to provide detail.

In both transformations, option (b) while more convenient than (a), would result in the original idea being lost to some degree. A collaborative design setting increases the potential for lateral and vertical transformations, but the sketching medium—traditionally pen/ pencil and paper—inhibits most instances of modification, especially when it comes to others' sketches.

Among collaborative, graphical ideation techniques, the C-sketch method [27] used in our preliminary study, prescribes the use of modification to effect lateral and vertical transformation, while the gallery method [25] prescribes the use of creation (of a new sketch) to do the same. The C-sketch method reduces the inhibition against modification by incorporating it into its prescriptive method, but suffers from the loss of earlier ideas in the bargain. The gallery sketch, while less prescriptive, is time-consuming and suffers from the limitation of the (paper) medium on which it is often performed. Digital tools for collaborative sketching have the potential to enable lateral and vertical transformations both by creation and modification. Such a system would need to make sketches persistent, by preserving older versions, and shareable, through duplication.

### 4.3.2 Participants

We recruited one team of 4 participants (3 male, 1 female), of ages 21-22 from an undergraduate senior elective course on toy design. All four were undergraduate students of Mechanical Engineering. Since the timing of the study was chosen to coincide
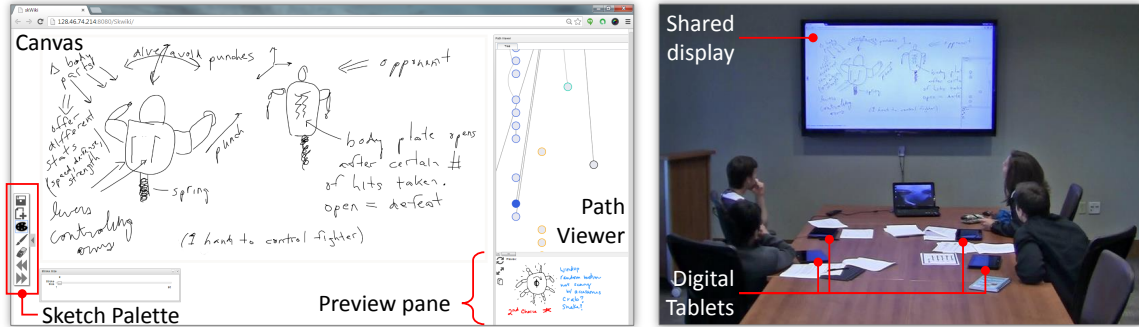
Figure 4.5. The experimental setup and apparatus are shown in the photograph on the right, with a screenshot of the skWiki interface on the left. skWiki runs in an ordinary web browser, and is thus platform-independent. The team is shown using skWiki on Microsoft Surface Pro devices, while a laptop running skWiki serves as a shared display for discussing ideas.

with the concept generation stage for the course project, the participants had been trained in the fundamentals of freehand sketching as detailed in Taborda et al. [89], understanding play value [90], as well as ideation techniques such as brainstorming and SCAMPER techniques [88]. As an incentive, participants were granted personal use of the tablet computers used for the duration of the study.

### 4.3.3 Apparatus

Each participant was given a Microsoft Surface Pro equipped with its dedicated stylus, the Surface Pen. skWiki was run on the Google Chrome browser in all the devices. In addition, a laptop, also running skWiki on the Chrome browser, was connected to a large display for participants to view and discuss each other's sketches whenever required. The experimental setup is shown in Figure 4.5. Note that in the normal working of the course, the participants are required to work with their teams in a classroom with paper and a board for each team to pin up their ideas.

### 4.3.4 Context of the Study

The course structure requires students to spend the most part of a laboratory session in concept generation using the above techniques, after which they continue working on the concepts outside class hours. They submit their final concept 18 days from the in-class concept generation session, and start work on detailed design using a CAD application. We designed our study to follow the same process. The study began from this concept generation session, and ended when the students moved on to detailed design and started generating CAD models.

The longitudinal study was split into two main stages:

1. **Concept generation session:** This was the part of the study when participants were more closely monitored. An observer was present at all times in the same room as the participants, monitoring them through the given tasks, and helping the team with any problems they faced with skWiki. Participants were video recorded with their consent.

2. **Concept development:** Here participants were required to use the system whenever they worked on the design. Their activity was not physically monitored, with server logs later used to study the team activity on skWiki.

At the end of stage 1, participants filled out a survey, reporting on their demographics and an immediate feedback of the process they followed and their experience with skWiki. At the end of stage 2, we conducted a short interview with the team, where they provided further feedback of their use with the system.

### 4.3.5 Tasks

The team was given a user interface guide, to show them the different features and operations possible with skWiki. The process of sketching, adding more canvases, and using the path viewer was explained, and the team was asked to practice creating and

saving sketches, as well as branching and merging operations. The team was allowed to practice using the interface until they were comfortable with it.

We then asked them to generate concepts for their project, an action toy. The concept design process was split into three tasks, with five-minute breaks in between:

**Task 1** (25 minutes): The team was asked to generate ideas for the toy based on the requirements given. Participants were instructed to sketch out no more than one idea per layout. However, they could create multiple sketches of the *same idea* on one layout. Participants were also asked to add brief descriptions and annotations wherever possible, and use the shared display whenever necessary to explain your ideas to the team.

**Task 2** (25 minutes): Participants were asked to discuss all generated ideas and select the most promising ideas. They were to extend these ideas, combine them with others, and generate new concepts, either by creating new sketches or by branching out from existing sketches using the path viewer.

**Task 3** (15 minutes): The team was asked to select a final idea and refine it.

At the end of the in-class session, the team was asked to continue using skWiki on the tablets, to elaborate the final concept and develop it over the following 18 days, at which point they would move to detailed design using a CAD application.

### 4.3.6 Results and Discussion

We used skWiki's interface to obtain all versions of the participants' sketches. We also used skWiki's path viewer to determine instances of branching, sketch development, and merging, as well as a rough overview of the sequence in which sketches were generated. The team activity on skWiki was matched up to the video recordings determine which sketches were created during each of the three tasks. We classified all sketching activity into the following categories:

S1 The creation of a new sketch from scratch, representing a new idea, with no reference to pre-existing ideas.

S2 The creation of a new sketch, representing a new idea that is inspired from or based on an earlier sketch, created by the same or by a different participant.

S3 The creation of a new sketch, showing additional detail of an existing idea.

S4 The modification of an existing sketch to create a new idea that is distinct from the original idea, but uses elements from the sketch.

S5 The modification of an existing sketch to add further detail to the existing idea, through dimensions, annotations, or additional features.

In the path viewer, sketches of types S1, S4, and S5 were immediately visible: S1 appeared as a new node, while S4 and S5 were visible as either branches or successive commits with significant changes. Studying the actual sketches further helped determine if they were of type S4 or S5. S1 and S3 were less straightforward: apart from studying the content of the sketch, noting the sequence in which the sketches were created was necessary to determine if a sketch belonged in either category. For the in-class concept generation session, viewing the video of the team discussion further helped categorize these sketches. The sequence of sketches by each participant, along with the identified source for inspiration or modification are shown in Figure 4.6 for the in-class concept generation session, and in Figure 4.7 for the out-of-class concept development sessions.

We immediately see that sketches of type S2 and S4 fall under lateral transformations, while sketches of type S3 and S5 fall under vertical transformations.

**In-Class Concept Generation:** Concept generation is essentially a process of exploring and expanding the problem space. We thus expected to see more sketches of types S1, S2, and S4 in this session, with a higher concentration of new ideas from scratch (S1) during task 1, and more lateral transformations, such as new ideas inspired by existing ones (S2) or modifications of existing ideas (S4), during task 2. This was indeed the case, as we can see in Figure 4.6. While vertical transformations (S3 and S5) are rarer in such situations, it is not unexpected to see them.
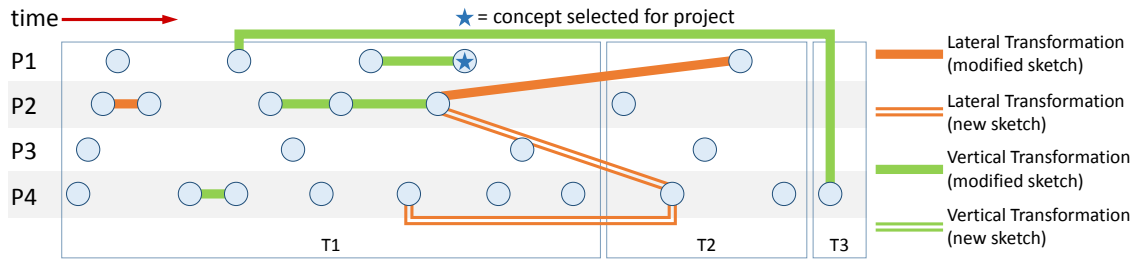
Figure 4.6. Lateral and vertical transformations observed in the in-class concept generation session with skWiki. P1–P4 are members of the design team, and their sketches are denoted by the blue nodes, arranged in the order of creation from left to right. Lateral transformations by branching, lateral transformations by visual reference, and vertical transformations (see Figure 4.4) that occur during this session are shown.

From the point of view of skWiki, the inter-participant transformations were of particular interest: evidence of branching and/or merging of ideas using skWiki, or sketches of types (S4 and S5). This occurred twice in this session. The first occurrence was between P2 and P1 as a lateral transformation: P1's idea of a toy sports car that "drifts" was transformed by P2 into a toy stunt car with a take-off ramp. The second transformation occurred between P1 and P4 as a vertical transformation: P1's idea of a mechanical spider was detailed by P4 through the addition of annotations detailing it as a wind-up, non-scary spider capable of making "random motions". Note that while the instructions of Task 3 were to select a concept and develop it further, it was not a mandatory requirement, and so the team merely short-listed promising concepts. The other lateral transformation seen was by participant 4, who used the preview and checkout options in the path viewer to draw inspiration from existing ideas and create a new sketch (type S2). There were no sketches of type S4—vertical transformations created by drawing a new sketch detailing an existing idea—which is normal for a concept generation stage. The final concept selected by the team,
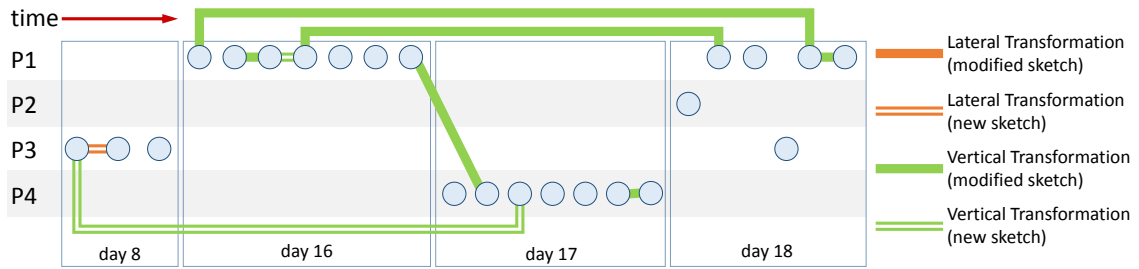
Figure 4.7. Lateral and vertical transformations observed in the out-of-class concept development session with skWiki. P1–P4 refer to the same participants as in Figure 4.6. Since this phase involves concept development, one would expect addition of detail to the selected concept, which translates to a higher number of vertical transformations as compared to the concept generation phase.

a robot sparring game proposed by P1, is represented by the node with a star in Figure 4.6 and its sketch is shown in Figure 4.5.

**Out-of-Class Concept Development:** The team was given possession of the tablets and styli for the next 18 days to develop the selected concept. We thus expected to see more vertical transformations of types S3 and S5, and these are indeed evident in Figure 4.7. While individual or team sessions during this stage were not monitored physically, monitoring the server activity helped determine that there were 3 sessions of individual participants developing concepts, with no-one else being online at the time, and one group session with all participants online at the same time. Our final interview with the participants revealed that they were also collocated during this session.

All the work done during this stage was in developing the selected concept. Technically, this meant that *every single node* in Figure 4.7 was a vertical transformation of the selected concept. This is not explicitly shown in the figure, so as to emphasize the transformations that occur within these sketches. Note that all instances of skWiki's branching and continuous commits here take the form of vertical transformations, with only one inter-participant transformation between P1 and P4, where

Table 4.1.  Number of sketches in each type (S1–S5) by each participant (P1–P4).

| | | Number of Sketches | | | | |
|---|---|---|---|---|---|---|
| | | S1 | S2 | S3 | S4 | S5 |
| Concept Gen. | P1 | 3 | 0 | 0 | 1 | 1 |
| | P2 | 3 | 0 | 0 | 1 | 2 |
| | P3 | 4 | 0 | 0 | 0 | 0 |
| | P4 | 7 | 1 | 0 | 0 | 2 |
| Concept Dev. | P1 | 6 | 0 | 1 | 0 | 4 |
| | P2 | 1 | 0 | 0 | 0 | 0 |
| | P3 | 3 | 1 | 0 | 0 | 0 |
| | P4 | 4 | 0 | 1 | 0 | 2 |

P4 adds further detail to annotations created by P1 on a mechanism description. Another interesting observation here is the emergence of participant profiles, notably that of P1 and P4: P1 commits fewer sketches in the concept generation session, but after his concept is chosen for the project, there is a marked increase in the number of commits he makes. P4's commits remain high in both sessions. Table 4.1 shows the number of sketches under each category (S1–S5) by each participant for the concept generation and concept development sessions.

Feedback from the team was positive overall: 3 out of 4 participants found skWiki to be an effective and fun way to collaborate. The fourth participant reported server lag as the main reason why he did not find the system effective. All team members reported that they felt less burdened due to skWiki, as they did not need to carry their sketches around, and that even without the tablets, they could open a browser on any computer and access skWiki to look up their sketches. They also liked the idea that saved sketches become immediately accessible to the entire team, and they no longer needed to be collocated.

With respect to the sketching experience, responses were mixed, with two participants finding the tablet hard to sketch on, while the other two found it easier to adapt to. All team members used the stylus to annotate the sketches, in spite of a keyboard being available, and the team was unanimous in their dislike of the application's relative insensitivity to handwriting. They suggested the embedding of audio files to sketches so they could speak out details while annotating key points, which would make communicating easier in a distributed collaboration setting. They also reported feeling impatient when waiting for a teammate to finish a sketch and share it, before they could view and discuss it; they desired a model that would update in real time. Finally, the team mentioned that they would like a "shared whiteboard" emulation, where multiple people could sketch on the same space simultaneously, which would improve their collaboration.

## 4.4   Implications

Our skWiki implementation uses the paths concept to support effortless collaborative creating, sharing, and merging for multimedia. A key component for managing these meandering paths is the path explorer, which not only tracks paths in space but also in time and across multiple users. Through these studies, we attempted to answer our research questions of how skWiki functions as a tool for collaboration, and whether it aids the expansion of the design space and the development of design solutions. We observed the different transformations at different stages of concept design, and found how it fits in Goel's categories of transformations.

Our second research question concerned the effectiveness of skWiki as a research tool. We found that skWiki enabled us to see the progression of ideas and the collaboration between team members, as well as distinguish between the different forms of lateral and vertical transformations. We will discuss both these answers in detail.

### 4.4.1   Interpreting skWiki as a Collaboration Tool

Recall that skWiki was designed with the four aspects of storage and retrieval (A1), data atomicity (A2), collaboration representation (A3), and capture of user intent (A4). Aspects A1 and A2 in a sketch translates to history support, similar to feature-level history support for a CAD model. Just as a CAD model is more than just a boundary representation of a geometric entity, a sketch is more than just a collection of marks on paper. A CAD model represents a sequence of logical operations that captures user intent. Playing back this history allows both understanding of how the model was designed, and opportunity to modify the model from an earlier version of it. Similarly, skWiki's data representation allows an understanding of the designer's thought process, especially for early conceptual sketches. Additionally, it allows for a semantic grouping of chronological operations on a particular sketch. A sketch can thus be "rolled back" to an earlier version, and can be modified to take the idea in a different direction, thus increasing the potential for both lateral and vertical transformations. This potential is increased by the path viewer, through which a designer has access to every version of every sketch created by their teammates.

Designers have been described as "both doodlers, by virtue of their sketches, as well as hunters and gatherers...made evident by the types of paraphernalia that they collect for inspiration and reference" [9, p.186]. Designers take inspiration from the world around them, from analogies in nature to in mechanisms and products in different domains. The design notebook is the main implement of such a hunter-gatherer. In today's networked world, skWiki, running on a digital design notebook such as the tablet used in this study, becomes a powerful tool for accumulating and sharing new ideas through images, text, and sketches.

Further, the path viewer provides a representation of collaboration (aspect A3), or an overview of the team's activity. Group awareness has been a key research area in computer-supported collaborative work (CSCW), and skWiki's path viewer provides a non-intrusive, effective way for each team member to view their team's ideas as well

as progress. By assigning roles in skWiki, this awareness can be tweaked to either a WYSIWYS (what you see is what I see) or a WYSINWIS (what you see is not what I see) system, based on the demands of any given project [101].

### 4.4.2 Interpreting skWiki as a Research Tool

In this study, we were able to identify the sequence of the sketches and classify the different transformations that occurred, chiefly by playing back the sequence of activities and observing the connections between sketches and participants using the path viewer. The path viewer explicitly indicates the connections between sketches through its branching and continuous commit connections, and implicitly indicates the less tangible connections by allowing the researcher to view the sequence of sketches and compare the contents to determine possible origins of an idea. Additionally, through skWiki's aspects of storage and retrieval (A1), coupled with its data atomicity (A2), and capture of user intent (A4), a researcher can look deeper into the process of sketching to understand the designer's thought process

Schmidt [102] discusses research questions for CSCW to focus on, such as identifying the signals and cues an actor relies on when observing team activity, and the aspects of group awareness that are monitored actively in order to make sense of the surrounding (team) activity. skWiki thus doubles as a powerful research tool, allowing the researcher to collect data unobtrusively, and observe forms of collaboration among teams, whether they are synchronous, asynchronous, collocated or distributed.

### 4.5  Conclusions

In this chapter, we presented a brief overview of skWiki, and two studies for evaluating the operations enabled by the tool. In the lower-level study using the C-sketch framework, we saw that skWiki's paths model allowed participants access to all sketches at the end of each round, increasing the potential to create new variations of existing designs, without the danger of feature saturation setting in.

We embedded our longitudinal case study of skWiki in a design project in an undergraduate toy design course. Operations performed by the team were classified into five categories. We used Goel's categories of lateral and vertical transformation as a theoretical basis to understand these operations, and showed how skWiki increased the potential of such transformations, in turn increasing the potential for creative collaboration. We further illustrated the use of skWiki as a research tool for observing team collaboration, and for understanding explicit and implicit transfer of ideas between team members.

We obtained positive feedback from the team for skWiki being an effective and fun tool for collaboration. Suggestions from the team for further improvement included a more heterogeneous communication medium such as audio and video, in addition to text and sketches, and for a "digital whiteboard" allowing multiple designers to work simultaneously on a sketch. This is in line with findings by Purcell et al. [24], who showed that designers communicate by means of sketches, speech and actions, seamlessly switching between media to communicate with other designers. This form of "talking sketches" [100] can be enabled for distributed, or even asynchronous teams by incorporating such multimedia options for communication, akin to tagging sketches with gestures and speech. Observations from such studies will be used to augment the collaborative concept design process to enable higher-level operations such as enabling consensus among team members, collaborative tagging [103], and tag-based selection and filtering of paths.

We also discussed how skWiki's path viewer can be used as a research tool. However, to be an effective research tool, the path viewer visualization provided by skWiki needs to be embedded in context with other data collected, such as video recordings of the team and server log data of team activity.

It is worth noting that while skWiki allows both branching and merging, we observed that participants in both studies detailed in this chapter used only the branch operation. This is chiefly because the merging operation required a level of planning from the user: the current implementation of skWiki allows the merging of

canvases (similar to layers in CAD and image editing tools) and not the sketches or strokes themselves. Thus to be able to select parts of sketches to merge, they have to have been drawn on a separate canvas. This form of prescriptive interaction was seen as intrusive, leading to participants avoiding the merge operation altogether. In the next chapter, we study, using a simplified interface, the physical and cognitive aspects of selection and transformation operations that are required to merge sketches.

## 5. A CLOSER LOOK AT MERGING SKETCHES

In the preceding chapter, we have seen how collaborative design commonly involves multiple designers concurrently working on new sketches in *convergent* and *divergent* processes [4, 89]. In divergent phases, designers use creative thinking to create entirely new concepts, consequently expanding the solution space, whereas in convergent phases, designers use analytical thinking to evaluate and select the most promising ideas. Both these design processes are often associated with close coupling between participants [28, 104]: designers draw upon their own (and each other's) sketches in the divergent phases, and evaluate these sketches in the convergent phase.

Fixation—"the inability to see alternative solutions" [105, p. 231]—is often seen as a common roadblock in divergent thinking. There are several techniques to help delay fixation. Zahner et al.'s *abstraction and re-representation* technique [105], for instance, involves designers who are given specific solutions, and are asked to first abstract them to a more general solution, and then apply them to a different domain. More common techniques include concept maps [7], which help designers form new connections between ideas, or SCAMPER [20], which provides the designer with an array of transformations that can be applied to a given idea. Such techniques are intended to help with reframing the design problem and use existing designs to generate new ones. We also saw in the preceding chapter how these techniques have been adapted to sketching with methods such as C-Sketch [27] or the gallery method [3], where designers sketch on existing ideas to transform, extend, or enhance them. The extension is natural, as sketches serve as the designer's thinking medium: they externalize the designer's thought [106, 107], provide a way for the designer to discover spatial features and relations [108], and provide underlying shapes for the designer to change and transform [33]. Sketching also aids transformations of ideas at a cognitive level, what Goel [17] calls *lateral transformations*—changing the idea into

a different, but related idea—or *vertical transformations*—creating a newer version of the same idea.

Reinterpreting a sketch by regrouping its constituent components is another way to battle fixation. Suwa et al. [108], for instance, show through their studies with architects how perceptual reorganization of components of sketches led to new interpretations. This chapter focuses on studying the *physical* act of regrouping these sketches by taking their components and merging them to represent alternative or new ideas, as well as the *cognitive* aspects of changing the function and/or context of these components.

In a traditional paper-based design scenario, the merge operation may be performed in any number of ways, ranging from drawing a new sketch from scratch while incorporating themes from both source sketches, to to physically cutting and pasting together the new sketch from its source materials using scissors and glue. In a digital scenario, on the other hand, merging simply involves selection and composition of objects using digital "cut", "copy", and "paste" operations. In other words, digital settings do not require completely redrawing or physically destroying the source materials, and is thus significantly more efficient in terms of time and resources. Despite this, the merge operation has received scant attention in digital design tools, and little is known of best practices for designing the interface, interaction, and methods for merging two source sketches into a new sketch. Furthermore, few digital design tools provide support for merging by re-purposing design elements from the source sketches into entirely new ideas and concepts in the resulting new sketch.

We remedy this gap in the literature by presenting results from an in-depth qualitative user study involving 19 designers combining sequences of two pre-drawn source sketches into a single output sketch. The study involved the divergent process in two forms; one where participants were asked to combine the source sketches into an output sketch that incorporated ideas from both, and the other where they were asked to use elements from both source sketches to create an output sketch with entirely new

ideas. We compared the qualitative performance for this merge operation in both phases for paper as well as digital settings.

We tested two digital selection techniques: free-form selection where users could 'cut' and use parts of a sketch as they do with paper, and an object-selection technique that used sketch clustering [109] to support selecting composite sketch elements using a simple click. Apart from the selection options, the digital setting differed from the paper setting in two important ways: (a) selecting a part of a sketch did not *remove* it from the original sketch, so it was available for selection again, and (b) the selected part of the sketch could be non-uniformly scaled to both re-interpret and re-purpose it into the 'merged sketch'. To better elicit the effect of these differences from participants, we also made them perform one merge task with paper.

In the same way that sketching, a physically expressed process, acts as an extension for the designer's memory, reducing their cognitive load, we sought to understand the cognitive operations involved in *merging* these sketches. To this end, we identify four cognitive operations of reuse, refactor, repurpose, and reinterpret, and suggest ways in which they support design ideation. We use the merged sketches from our user study to understand the relation between these cognitive operations and the extent of exploration of the design space by the participants. Our results give a deeper insight into how these operations aid design exploration, and makes the case for digital design tools as strong alternatives to traditional paper-based design methods.

## 5.1   Design Framework: Merging Sketches

We call the operation to combine two[1] source sketches $A$ and $B$ into a new resulting sketch $C$ a *merge* (or "sketch merge"). To understand what happens in the sketch merge operation, we need to separate the physical aspects of the process from its cognitive aspects. By physical aspects, we mean the parts of the operation that are expressed externally by the designer: operations such as select, cut/copy, paste,

---

[1]We limit our treatment here to two source sketches, but these ideas should generalize to a higher number of source sketches.

rearrange, and compose. By cognitive aspects, we mean the parts of the operations that occur internally, in the designer's mind. This involves the designer's internal representation of what the sketches denote: the ideas expressed in the source sketches, or newer ideas that are formed in the designer's mind when parts of these source sketches are isolated or combined. These two aspects of merging are intricately connected: sketches are external representations of the designer's evolving thought process [106], and identifying these aspects may help understand how they come together.

### 5.1.1 Physical Operations in Merging

The physical operations that come into play when merging sketches can be categorized into two kinds of processes:

- **Selection:** This is act pf selecting a part or parts of source sketches.

- **Transformation:** Translating, rotating, and scaling the selected parts to form the merged sketch.

#### 5.1.1.1 Selection

At a physical level, selection involves identifying and demarcating areas of interest from the source sketches. We identify two forms of sketch selection:

- **Freeform selection:** Commonly known as "lasso select" in existing drawing and photo manipulation editors, this requires the marking of a freeform shape depicting the area of interest, and selecting everything that lies in this area.

- **Object selection:** While freeform selection is a natural choice for most image-based media, sketches are also collections of strokes, and it makes sense to have a selection mechanism that is aware of this. This idea is not new: drafting applications allow the selection and manipulation of lines or groups of lines, which the user can select directly, or by chaining connected lines/curves. This allows selecting coherent "objects" inferred in the sketch.
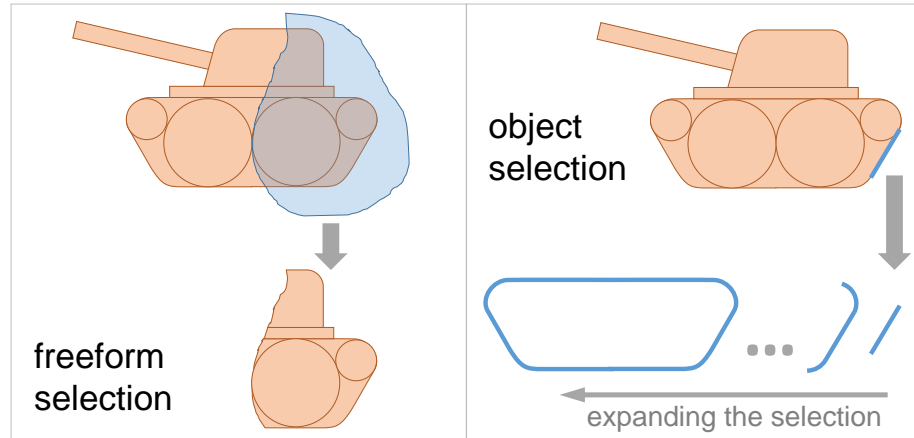
Figure 5.1.   The two selection types considered for this study.  The freeform selection type is an area-based selection commonly seen in image editors and other graphical applications.  The object selection type is a stroke-based selection that extends the initial selection of one stroke to include other temporally adjacent strokes.  Both methods have their advantages: freeform selection frees the user from having to select complete shapes.  Object selection allows the selection of a shape from nested or interlocked shapes as long as the shape consists of temporally adjacent strokes.  Figure 5.2 details this mechanism.

These two forms are shown in Figure 5.1. To implement object selection, we need to overcome the problem that sketches consist of unorganized strokes, not discrete objects. Contrast this with traditional vector editors, where illustrations are created using geometric shapes and paths. To identify higher-level entities, we use an agglomerative stroke clustering method based on the time and order in which the strokes were created [109]. The strokes in a sketch are thus combined into clusters based on a proximity measure and temporal order, and these clusters (a group of strokes) are further combined with other strokes or clusters using the same approach. This recursively builds a tree-like structure (a dendrogram) representing the clustered sketch (stroke) histories (as illustrated in Figure 5.2. Note that the root cluster on this tree represents the entire sketch, while the leaves represent individual strokes. Selecting a particular stroke allows us to expand the selection using the cluster hierarchy (by
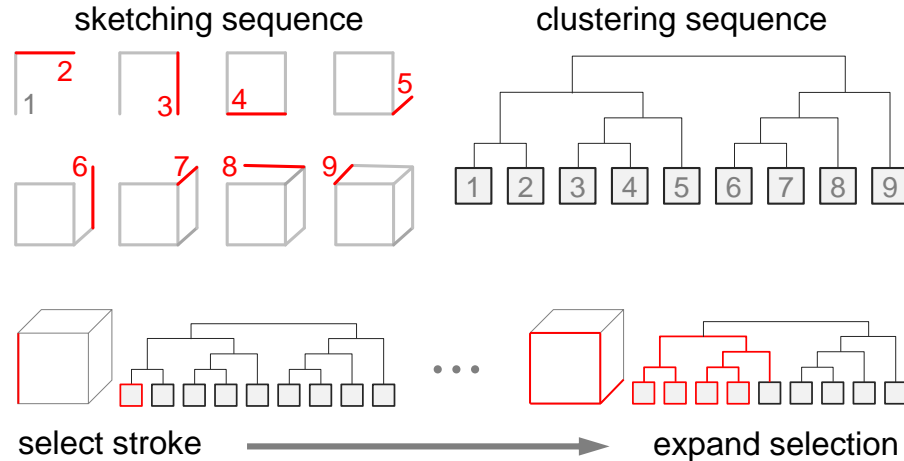
Figure 5.2. An illustration of the sketch history-based clustering technique for grouping strokes into objects, and the expansion of selection by traversing to the parent nodes of a stroke (cluster) in the tree layout.

moving to the closest parent in the tree); by giving users control over this, they can smoothly expand or shrink their selection with little effort.

This is just one automated clustering algorithm that can used for this purpose; there are many other alternatives from the data analytics domain including k-means and k-d trees. Our approach uses agglomerative clustering as it creates a hierarchy of stroke clusters, which is amenable to the process of "expanding" a stroke selection. Additionally, while we use temporal order to cluster, the technique can be changed to use spatial proximity, or a combination of both: selection and expansion will still be possible. Overall, selecting a specific sketch part that conceptually belongs together— such as a wheel on a car, a person in a room, or a part in a schematic—becomes significantly more efficient than freeform selection.

### 5.1.1.2 Transformation

Once the designer has selected a visual element to transfer from one of the source sketches to the resulting sketch, they may want to transform the element in the final
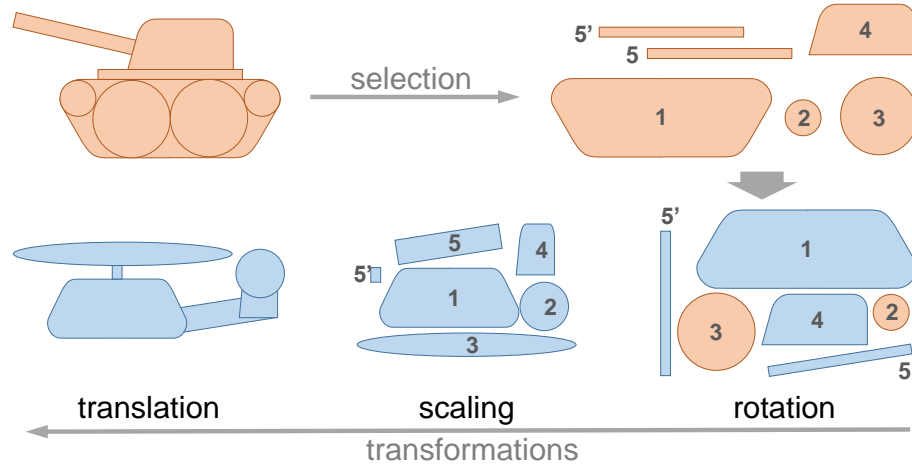
Figure 5.3. An illustration of the different transformations and their role in changing what the selected components represent. The components, selected from the sketch on the top left, are numbered to to keep track of them through the transformations. Parts of sketches that undergo at least one transformation at a given stage are colored blue. The final composed sketch is shown on the bottom left.

result. Examples include changing the location, size, or orientation of a selected element in the composition. To accommodate this, the merge operation incorporates translation, freeform scaling (with non-fixed aspect ratios), and rotation for each individual element. An example of the combined use of these transformations is shown in Figure 5.3.

In divergent design phases, *selection* is used to physically embody how a concept is distanced from its surrounding context, while *transformation* is used to increase this distance.

## 5.1.2 Cognitive Operations in Merging

Cognitive techniques such as *abstraction and re-representation* [105] or SCAM-PER [20] provide the designer with the means to overcome or delay fixation. The act of merging components from two sketches to represent a new idea has a cognitive side that is aided by the above physical aspects of selection and transformation. We
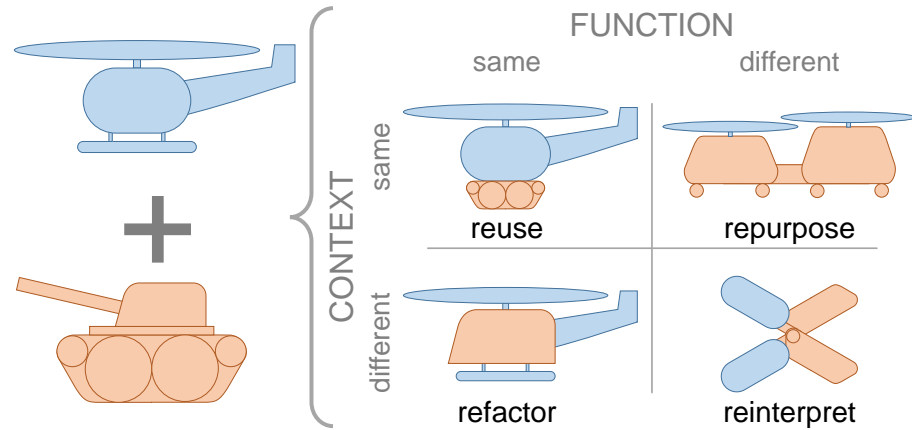
Figure 5.4. Merging as different cognitive processes. The illustration shows the ways in which the change or constancy of context and/or function creates different results in the merged sketch. In "reuse" operations, both context and function of merged components are similar to what they were in the source sketches. They result in ideas very similar to the source sketches. "Reinterpret" operations, where both the context and functions of merged components are changed, result in ideas that are very different from the source sketches. The "repurpose" and "refactor" operations result in ideas that are somewhere in between. The components used from the source sketches on the left are indicated by color in the four merges on the right.

generalize these cognitive aspects to two: *function* and *context*. Function is the role a component plays in the original or source sketch, whereas context is the domain in which the component or its parent sketch exists. For example, in a sketch of a car, the tire performs the function of supporting the car, and providing traction; its context is the automotive, or, more generally, the transportation domain. When a designer performs a merge, components that go into the merged sketch can either have the same or different function as in the source sketch, and exists in the same or different context as in the source.

Based on these possibilities, we identify four cognitive operations that can occur when a sketch $C$ is created by merging elements from sketches $A$ and $B$.

- **Reuse:** The use of a component from $A$ or $B$ in $C$ where both the function and the context of this component in $C$ is the same as in the source sketch.

This is akin to the use of morphological matrices [110, p. 204] in conceptual design, where a product's functions are broken down into sub-functions, multiple concepts generated for each sub-function, and then new product designs are generated through various combinations of these concepts. In each design, a particular concept, if selected, always performs the same function. Thus, the *reuse* operation is often seen when generating alternatives for a design problem by choosing parts of existing design solutions. This is illustrated in Figure 5.4 as the use of the tank tread on the undercarriage of the helicopter. The tread performs the same function as it did for the tank. Cognitively, this can be seen as a representation of Goel's lateral transformation [17], where a sketch undergoes a modification to a version that is similar to, yet distinct from, its earlier version.

- **Refactor:** In computer science, *refactoring* is the process of reorganizing a program's internal structure without altering its external behavior. In our context, we interpret the term slightly differently, as the use of a component from $A$ or $B$ such that it performs the *same function* in $C$ as it did in the source sketch, but in a *different context*. The sketch component thus performs the same function, but can be seen as having a different structure. This is illustrated in Figure 5.4 by the use of the tank's armored body as the fuselage of the helicopter. It performs a similar function as it did in the tank (protect and house the pilot/driver), but in this case, it is now an armor-plated fuselage. This operation can thus help in exploring the design space further by abstracting the function performed by a part of the sketch.

- **Repurpose:** Here a component from sketches $A$ or $B$ is used in $C$ in a way that it occupies the *same context* or domain as the original sketch, but performs a *different function*. This operation often uses the form of a component of a sketch to replace a different component. For example, in Figure 5.4, the outline of the tank tread is repurposed to now play the role of the fuselage of the helicopter.

- **Reinterpret:** Finally, when an element from $A$ or $B$ is used in the merged sketch $C$ such that it performs a *different function* in a *different context* from the source sketches, we term the operation a reinterpretation of that component. This operation could be particularly useful in the early stages of ideation, as it creates results that are the most different from the source sketches, thus helping to explore the design space better. In the example shown in Figure 5.4, components of the tank and the helicopter are completely reinterpreted to create a pair of scissors. Our definition of reinterpretation is derived from Suwa et al.'s [108] perceptual reorganization, where regrouping parts of sketches provides new interpretations.

### 5.1.3  Paper vs. Digital Media

In traditional design settings conducted using paper, merging two source sketches often happens in many different ways, at different levels of abstraction:

- **Recreate from scratch:** Here the designer merely uses the source sketches as inspiration when creating a new sketch that combines the original ideas;

- **Trace:** The designer reproduces elements from the source sketches by tracing over them using tracing paper; and

- **Cut and paste:** Source sketches (or copies) are used directly in producing the resulting sketch by physically cutting out elements and pasting them together.

Note that when sketching on paper, transforming the selected parts of the sketch is not trivial. Paper allows for only rotation and translation, but for scale, it would need additional mechanisms such as pantographs (for tracing a larger shape), or photocopiers (for creating a larger sketch to cut out). The *recreate from scratch* option does allow for all transformations, but it depends on the sketching skill of the designer. Combinations of these methods, as well as entirely different methods than
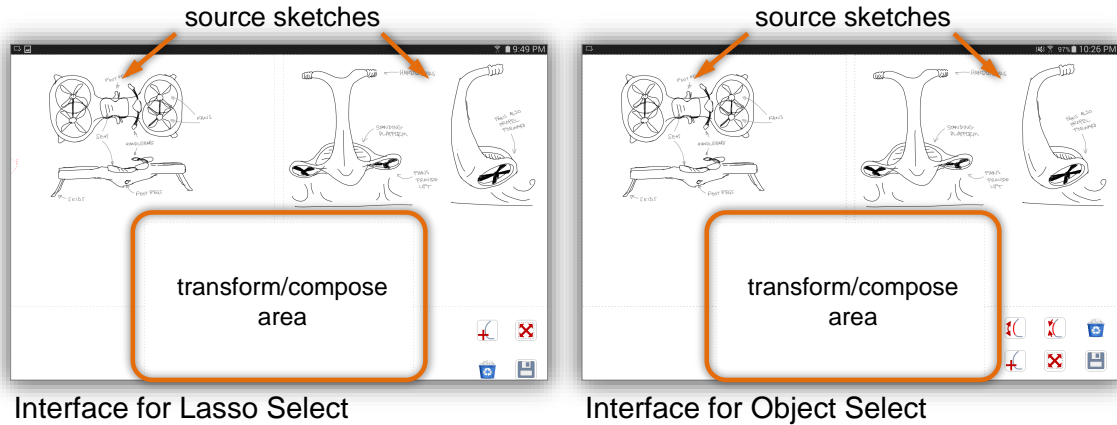
Figure 5.5. The interfaces of the applications used for the freeform and object select methods. The two source sketches are placed in juxtaposition to the "merge area", where the selected sketch segments can be placed, transformed, and composed for the merged sketch.

these, are certainly possible, but at a conceptual level would still involve the same physical and cognitive operations as detailed earlier in this section.

Digital sketching platforms have both strengths and weaknesses compared to traditional paper-based sketching [38,98]; one of the oft-cited strengths is the possibility to effortlessly reproduce a sketch without consuming the original. Despite this, few existing collaborative sketching platforms provide explicit support for sketch merging beyond allowing designers to recreate existing sketches into a new sketch.

While recreating a sketch may sometimes be exactly what the designer wants, the digital medium allows us to better support situations where parts of the original sketches are reproduced in the resulting sketch. We performed a study of such merging operations under two different design situations, in order to understand the physical and cognitive operations involved, and the role of the medium in supporting these operations. This study is detailed in the following section.

## 5.2 Study Methods

In the previous section, we outlined the physical and cognitive operations of merging sketches, with possible implications on the outcome. To understand how these operations affect the results, we conducted our study in three stages:

- **User Study:** We examined how designers merged two sketches under conditions that required them to use all the physical and cognitive operations discussed in the design section. The goal of this study was to understand how designers perform merging under different design scenarios when using different tools, and to obtain a set of merged sketches under controlled conditions.

- **Crowdsourced Survey:** We then performed a crowdsourced survey where respondents were shown a set of sketches created from the same source sketches by participants using different techniques. They were asked to choose sketches that were (a) the most different from the source sketches, and (b) represented the most original idea.

- **Coding of Cognitive Operations:** Having defined the four cognitive operations involved in merging, we identified, in each sketch, the number of times each operation occurred uniquely in each sketch.

### 5.2.1 User Study

In the previous section, we had identified the cognitive operations involved in merging, namely *reuse* to explore the design space close to the source sketches, *reinterpret* to explore the design space furthest from the source sketches, and *refactor* and *repurpose* to explore the space that lies in between. We split the user study into two stages to address these explorations: (1) a "functional" merge, where the source sketches are possible solutions to the same design problem, and (2) an "exploratory" merge, where the source sketches are unrelated to each other. In the case of the functional merge, the design task was to generate a merged sketch that addresses

the *same design problem* as the source sketches, while in the case of the exploratory merge, the design task was to generate a merged sketch that is *completely different* from the source sketches.

### 5.2.1.1 Participants

We recruited 19 paid participants (4 female, 15 male), 16 of whom were students of a senior course on toy design, and 3 who were graduate mechanical engineering design majors. Participants were aged between 21 and 28 years, had normal or corrected-to-normal vision, and 17 were right-handed, and 13 participants were proficient in using tablets and styli. All participants had novice to intermediate level skill in sketching on paper. All demographics were self-reported.

### 5.2.1.2 Apparatus

We conducted the tablet portions of the experiment on an Samsung Galaxy Note 10.1 2014 edition using the S-Pen with a < 1 mm tip for input. For paper, the participants were provided with sketches printed on two different colored letter (8.5 × 11 in.) papers along with scissors, glue, and a blank sheet of white letter paper for the merged version.

### 5.2.1.3 Tasks

For both stages of this study that were performed on the digital medium— functional merge and exploratory merge—we asked participants to use both techniques: freeform selection and object selection. We developed an Android application for each of these techniques. The visual space on the Android device display was divided into two main areas corresponding to the physical operations discussed earlier: a *selection view* where two source sketches were displayed from which the participants could select elements, and a *transformation view* where they could apply

transformation operations to the selected elements. Participants were free to orient the tablet or paper to their comfort. The outcome of each merge operation was stored on the tablet before the participant moved on to the next task. At the end of each task, the participants' brief description of their creation was recorded.

In both functional merge and exploratory merge stages, participants performed tasks using two different merge techniques on digital media (a direct-touch tablet): freeform selection and object selection. All transformations discussed earlier were enabled for all the above conditions with standard tools for scaling, translating, and rotating the selections provided to the participants.

After each stage, the participants answered a series of questions about their experience with the merge technique for that particular stage. Questions were answered on a 1–5 Likert scale on aspects such as ease of selection, successful expression of selection intent, and open-ended questions explored the selection technique and transformation techniques that were liked or disliked. The participants also conducted one of the two merge tasks (assigned randomly) on paper using the cut-and-paste with scissors, paper cutter, and glue.

### 5.2.1.4 Dataset

We used a total of eight pairs of annotated sketches created by two expert designers for these tasks. Of these, four pairs were created such that each pair showed two solutions to the same design problem, for use in the functional merge stage. The remaining four pairs were created such that each pair showed two sketches that were completely different and addressed different design problems. These pairs were used in the exploratory merge stage. These sketches were processed to have similar visual appearance in terms of the stroke width, colors used, and the view (the sketches are all perspective views) to avoid the effects of sketch quality on the merge outcome.

### 5.2.1.5 Factors

We studied the following factors in the experiment:

- **Situation (S)** represents the design scenario or stage as described earlier: functional merge and exploratory merge.

- **Technique (T)** represents the selection technique used in the task: freeform selection and object selection.

We chose to not study medium (digital vs. paper) as a factor since the goal of this study is to identify and compare merge techniques for different design tasks on the digital medium. In other words, the two media are simply not comparable due to the different workflows involved. However, having a paper-based task expands our understanding of the benefits of the digital merge techniques compared to the paper-based merge, which is already known and familiar to participants.

### 5.2.1.6 Experimental Design

We used a full-factorial within-participants design:

|   |     |                                               |
|---|-----|-----------------------------------------------|
|   | 19  | participants                                  |
| × | 2   | **Situation** $S$ (functional/exploratory merge) |
| × | 2   | **Technique** $T$ (freeform/object select)    |
| × | 2   | repetitions (with different source sketches)  |
|   | 152 | total trials (8 per participant, training excluded) |

This does not include the paper task, which had only one trial. The order of both situation $S$ and technique $T$ was randomly chosen. For each trial, we collected the following metadata:

- **Selection areas:** The proportion of each of the source sketches retained in the merged outcome.

- **Duration:** While each participant was required to spend 5 minutes on each task, this was not strictly implemented, to allow the participant to finish a task if they were close to completion. The actual time taken was recorded.

- **Number of selection operations:** The number of discrete selections that made it from the selection area to the transformation area of the interface.

### 5.2.1.7 Procedure

The participants were provided with an introduction to the purpose of the study and the data to be recorded. They filled out an initial survey on their background and their experience with sketching on tablets and design. We then provided a set of instructions for the first stage of tasks to be performed (functional or exploratory merge) and asked them to practice using the assigned selection technique and the transformation operations before the tasks began. Participant training time ranged from 3 to 5 minutes per technique.

This was followed by a set of four merge tasks with two merge techniques for the assigned stage of the task (functional or exploratory merge). The participants were given 5 minutes to complete each task. We determined this time limit by analyzing the level of detail in each set of sketches through a discussion with the expert designers who created the original sketches; the purpose was to pick a short period of time amenable to quick ideation.

Prior to each task, the source sketches were briefly explained and then the participants were asked to infer more detail from the annotations. The participants were told to ask the administrator any question about the sketches during the session if necessary. At the end of each task, the participants were asked to explain the purpose and idea behind their merged outcome. After completing the survey forms for this stage, the participants moved to the next stage, repeating the above procedure

Following this, the participants moved to the paper-based task, during which the procedure for merging sketches on paper using cutting and pasting was explained to

them. They were given five minutes to finish their composition. The administrator was present throughout the session to monitor and answer any questions regarding the sketches if the annotations are not clear enough. The sessions were also video recorded. A typical study session lasted 70 minutes.

### 5.2.2 Crowdsourced Evaluation

Keeping in mind our original goal of understanding the effect of the selection technique on the final outcome of the merge, we decided to evaluate the participant-generated ideas on their originality. Additionally, to determine the extent of how different the outcomes from the exploratory merge tasks were from the source sketches, we also decided to evaluate the merged sketches on their dissimilarity from the source.

We collected the eight sets of merged sketches and performed a crowdsourced survey. Each set contained 21 merged sketches from 19 participants (19 digital merges and 2 paper merges for every pair of source sketches). Respondents to the crowdsourced survey were presented with a total of 12 questions. Eight questions related to the originality of the merges from all eight pairs of source sketches the participants used, and the remaining four questions were specific to the four pairs used in the exploratory merge sketches.

Each question had the same format: the two source sketches were shown, along with their descriptions, and 4 options were presented, requiring the respondent to select the most original sketch (or the one most different from the source sketch, as the case may be) from the given choices. These 4 options were generated randomly for each respondent from the pool of 21 merged sketches generated by participants of the user study. The limit of 4 choices were imposed to make the task of choosing easier for the respondent. With enough participants, the randomization would ensure that each merged sketch received sufficient representation. The options shown to each respondent and their choice were both recorded.

Respondents were also required to briefly explain the rationale for their choice for each question in a text field provided. This also served as a means for weeding out spurious responses: if a rationale was not provided, the corresponding response was removed from our evaluation. We received an average of 53 responses for each question. Based on these responses, we calculated the following score for each sketch:

$$M_{score} = \frac{(\#sel) - (\#nsel)}{(\#sel) + (\#nsel)} \tag{5.1}$$

Here $\#sel$ is the number of times a particular sketch is selected by the crowd, while $\#nsel$ is the number of times a particular sketch is shown as an option, but not selected. This normalizes the score for each sketch based on the number of times it is shown to a respondent.

### 5.2.3  Coding of Cognitive Operations

As discussed in the design section, we identified four cognitive operations that are possible when a component from a source sketch is incorporated into the merged sketch: reuse, refactor, repurpose, and reinterpret. We analyzed each merged sketch, and used the corresponding participant descriptions to interpret the role of each component used from the source sketches. Once the role was identified, we coded this element as an instance of one of the above four cognitive operations. Each merged sketch could thus have zero or more instances of these four operations. We did this only for unique roles: e.g., if a component was reused as a wheel 3 times, it still counted as one instance of reuse. We recorded the number and kind of each cognitive operation used in each merged sketch. This coding was performed by three of three of the investigators of the experiment, whose inter-rater reliability was calculated to be 0.82 (Cronbach's alpha test).

## 5.3 Results

We analyzed participant feedback on their experiences with the different selection techniques for significant differences, and the scores from the crowdsourced study for correlations with the number of cognitive operations identified.

### 5.3.1 Participant Feedback

The participant responses to questions on ease of selection and success of selection for the selection technique did not follow a normal distribution. We thus conducted a non-parametric Wilcoxon signed rank test, which showed that for the composite design merging tasks, participants reported that selecting parts of the sketch using the freeform select technique was significantly easier ($Z = 3.69$, $p < .001$) and more successful ($Z = 2.95$, $p = .002$) than when using the object select technique. The test also showed that for the "new design" merge tasks, participants considered the freeform select technique significantly easier to use than the object select technique in selecting parts of the sketch ($Z = 3.01$, $p = .002$). However, participants did not rate the freeform select technique as significantly more successful in selecting parts of a sketch than the object select technique ($Z = 1.33$, $p = .19$). A comparison with paper was not performed as each participant used paper only once, and for only one of the two merge tasks. These results are shown in Figure 5.6.

### 5.3.2 Crowdsourcing Results and Cognitive Operations

We performed correlation analyses between the scores described in Equation (5.1) and the number of cognitive operations (reuse, refactor, repurpose, and reinterpret). Figure 5.7 shows a sample of sketches annotated with these codes. The analyses revealed no significant correlation between the scores of the sketches created for the functional merge tasks. Recall that only originality scores were obtained for this set of sketches. For the sketches created in the exploratory merge tasks, there was
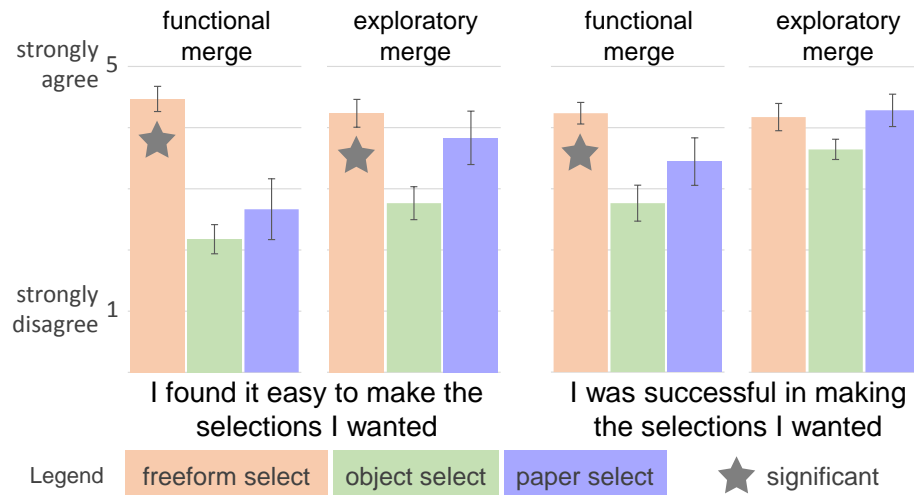
Figure 5.6.   Participant feedback on ease and success for each selection technique, for each stage of the user study.  Freeform performed significantly higher in all cases except for success in exploratory merge.

a significant negative correlation between the dissimilarity score and the number of instances of reuse (Pearson's $r = -0.53$, $p < .001$), and a significant positive correlation between the dissimilarity score and the number of instances of reinterpretations (Pearson's $r = 0.41$, $p < .001$). For these same sketches, there was a negative correlation between the originality score and the number of instances of reuse (Pearson's $r = -0.23$, $p = .033$), and a significant positive correlation between the originality score and the number of instances of reinterpretation (Pearson's $r = 0.28$, $p = .011$).

We also conducted a multivariate test to examine participants' application of reuse, refactor, repurpose, and reinterpret operations while working on functional and exploratory merge tasks. The results showed that participants used significantly greater number of reuse operations ($F(1, 157) = 4.28$, $p = .04$) and repurpose operations ($F(1, 157) = 13.71$, $p < .001$) in functional merge tasks than in exploratory merge tasks. The participants used significantly greater number of reinterpretation operations for exploratory tasks than function tasks ($F(1, 157) = 46.69$, $p < .001$).

Finally, we conducted a correlation test for selection methods and cognitive operations.  The results showed that there is significant correlation between selec-
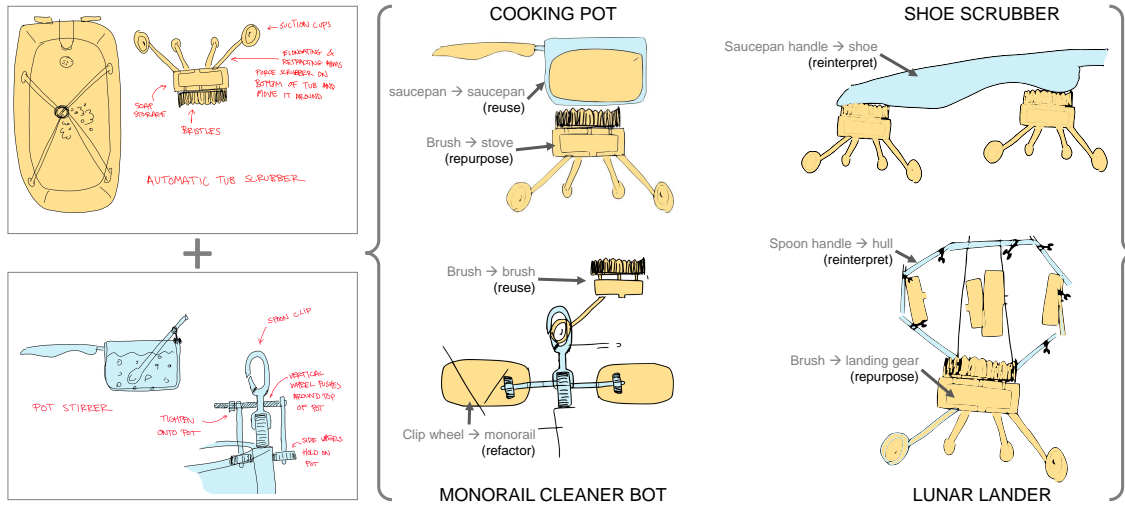
Figure 5.7. A sample set of participant responses to an "exploratory merge" task. The two sketches to the left were the inputs to the participants, and the four sketches within the curly braces were created by merging components of these two source sketches. The source and merge sketches are colored post-hoc to better illustrate the components that were used in the merge. The labelling of reuse, refactor, repurpose, and reinterpret were created by the coders based on the classification detailed in the design section.

tion methods (freeform select, object select) and the repurpose operation (Kendall $\tau = -0.17, p = 0.02$), suggesting that participants who used the cluster method are more likely to use repurpose. There was no significant correlation between selection methods and other types of cognitive operations.

## 5.4   Discussion

In this section, we first discuss the implications of the physical and cognitive operations in merging, and then discuss the larger implications of our findings.

### 5.4.1    Selection Operations

From the participant feedback summarized in Figure 5.6, it is clear that participants preferred the freeform select over object select and paper-based cut-and-paste technique. This is partly due familiarity: the freeform selection technique is standard in most graphical editing tools. However, 15 participants did report that there were portions of the sketches they were *unable to avoid* when they were trying to select something else. This reflected an inherent nature of the freeform selection technique: it requires the formation of a closed area, and everything within that area is selected. As one participant observed, "*It was sometimes difficult to exclude internal shapes; strange lasso patterns were necessary to avoid them.*" In most graphical editing applications, this issue is side-stepped by using layers or selection filters.

Our intent in introducing the object select technique was to study the effect of a possible alternative that seemed well-suited to sketching applications, where collections of strokes could be grouped. One participant appreciated the feature: "*Sometimes the object select mode has better results. I can select the contour of a sketch and avoid the things inside for once.*" However, most had issues with the technique: an average of 12 participants reported finding it difficult to select specific parts of the sketches, while avoiding other parts of the sketches. One participant remarked that the object-select technique gave him "*limited control over what primitives and shapes I was selecting.*" Another participant felt that he was operating blind: "*I had no way of knowing what would be selected next, so I spent most of the time blindly adding and selecting new starting points to avoid what I did not want.*"

These problems stemmed from the distance metric used for the agglomerative clustering of the strokes. In our case, we used the temporal difference between strokes as the metric, which makes it sensitive to sketching styles. For example, when sketching out a design, the designer's tendency is to first sketch out the overall form of the object, and then adding details to its features iteratively. If a novice designer were to use a version of the object select method, they may want to select and reuse an

entire feature, which they will find difficult to do. An expert designer may want to just select the overall form and then add their own detail to it.

It was clear that the stroke clustering algorithm for the most part did not cluster the strokes the way the participants seemed to mentally "chunk" the strokes. Waldron and Waldron's experiment on recall of technical drawings from short-term memory [111] showed that novice designers recalled drawings in terms of line segments, intermediate designers recalled in terms of machine components, and experts in terms of higher-level functional components. This brings forth the more complex question of whether or not it is possible to cluster the strokes of a sketch in a way that makes sense to most viewers. Further analysis is needed with different clustering techniques to determine which approach closely mimics the way designers mentally chunk parts of sketches.

Recall that participants had only one attempt on using paper-based merge to perform one of the functional merge or exploratory merge tasks. While this was not enough for a statistical comparison, doing this task at the end provided them perspective to comment on the experience. As seen in Figure 5.6, participants liked the paper-based selection technique more than object select, chiefly because it was a medium they had grown up to be comfortable with. However, they realized that the process of selecting (by cutting out) one element of the sketch from paper, other elements become partly or completely unavailable to them. They also remarked on the irreversibility of selection: "*I did destroy some parts getting to the pieces I wanted*", "*I really wish I could duplicate some of the parts like in the tablet version*", and "*no undo*" were comments we received from multiple participants.

## 5.4.2 Transformation Operations

The transformation operations of translation, rotation, and scaling, were enabled for the tablet interface using multi-touch gestures. As almost all the participants were used to these gestures through their familiarity with smartphones and tablets, they
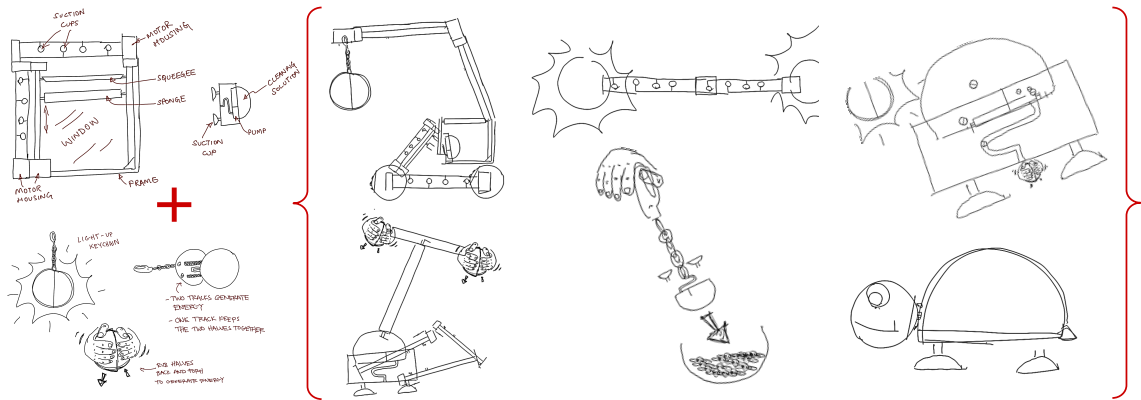
Figure 5.8. A sample of merged sketches that were developed from the same two sketches (left) by participants in our study of merging techniques.

found it easy to perform most of these transformations. However, some had issues with rotating selected components to specific orientations, especially with smaller selections. While transformation operations were not an explicitly controlled part of the study, the paper-based task illustrated the difference: while participants could translate and rotate paper cutouts with the same ease as their digital counterparts, they could not scale the parts. One participant noted "...it wasn't as easy as (the) digital version because I couldn't copy and scale". Another participant remarked, "one issue here is that I cannot scale any of the cut-outs, which resulted in a cartoonish-looking final piece."

The role of these transformation operations can be equated to the role of sketching in ideation: while the activity may be expressed physically, it represents an externalization of the designer's working memory [10]. A small sample of the variety made possible through transformations is shown in Figure 5.8, where exploratory merge operations of components from a sketch of an automated window-washer and a light-up keychain were used to generate a wrecking ball, a mortar and pestle, a light-up flute, and a turtle, among others.

### 5.4.3   Cognitive Operations

We had posited while defining the cognitive operations of reuse, refactor, repurpose, and reinterpret, that reuse represented an exploration of the space close to existing designs, and as such is a means to generate alternatives, rather than come up with original ideas. This is evidenced by the negative correlation between the number of reuse operations in a sketch and the originality and dissimilarity scores. The correlation between reuse operations and the functional merge task further supports this argument. This correlation was observed only for the exploratory merge tasks, possibly because the nature of the functional merge task itself was not amenable to a high level of originality: participants were forced to use parts of existing sketches to create a composite sketch that addressed the same design problem as the original sketches. Similarly, we had posited that the reinterpret operation represented a reorganization of the designer's perception of the existing sketch [108]. This is evidenced by the positive correlation between the dissimilarity and originality scores and the number of reinterpret operations. Additional support is provided by the significant correlation between the number of reinterpretations to the exploratory design task.

While significant correlations were not observed for the refactor and repurpose tasks, we are confident that a more nuanced selection of merge tasks designed to elicit more varied instances of "lateral transformations" [17] would throw further light on the relevance of these tasks to design ideation.

## 5.5   Limitations and Future Work

Our qualitative user study by no means represent an exhaustive exploration of the design space of merging sketches for creative design. For example, we restricted the scope of our work to stroke-based sketches as the only form of design output. While the literature suggests such sketches as the primary communication medium in collaborative design [32], there are plenty of other media such as text, annotations, and scrapbooks where our work cannot be applied.

Our interaction design was guided by discussions with our professional designer collaborators, and in no way is an attempt to reach an optimal layout, visual representation, and workflow. As we mention in our discussion on selection operations, the object-select technique has the potential to better tune selection operations to how designers "chunk" sketches. Our next step is to perform a more exhaustive study of clustering techniques to achieve a grouping that mimics this chunking. For instance, we are interested in pursuing hybrid sketch summarization methods in our future work that attempts to combine both temporal and spatial clustering. Finally, given that sketching is a meaningful way for a designer to engage in a dialog with their ideas, we intend to study the effects of selection and transformation operations that suggest new interpretations to the designer. This is perhaps best summed up by a participant in our study who said, "*Although sometime the selection wasn't your original idea, it may inspire you with other ideas that would also work.*"

## 5.6 Conclusions

In this chapter, we studied the physical and cognitive operations involved in selecting parts of existing sketches and merging them into a new sketch. In addition to the physical operations of selection and transformation, we identified the cognitive operations of reuse, refactor, repurpose and reinterpret. With a view to aid digital support for early collaborative design, we determined to study the above operations on a digital medium. We studied selection operations through a freeform selection tool and an object-select tool that uses a stroke-level aggregation method that combines strokes based on the temporal order of sketch strokes. To study the different scenarios of use, we conducted a user study where participants were given a set of design tasks that involved functional merges where the result addresses the same design problem as the original sketches, and exploratory merges where the result is completely different in form and function from the source sketches. Though the freeform selection technique was preferred by participants, we identified problems that can potentially

be addressed by object selection, with an appropriate clustering metric. We scored the merged sketches through a crowdsourced survey, and coded each sketch with the cognitive operations used. The scores from the survey for sketches produced in exploratory merge tasks were negatively correlated with the number of reuse operations, and positively correlated with the number of reinterpret operations. This supports our argument that these cognitive operations determine the extent of exploration of the design space. Finally, we closed with suggestions on further analyses to study the selection, transformation, and cognitive operations for merging.

# 6. A VISUAL ANALYTICS APPROACH TO UNDERSTAND DESIGN COLLABORATION

Design as an activity can be interpreted in different ways: as a social practice that requires interaction between designers [112], as a cognitive process in the form of a dialogue between the designer and their sketch [106], or as an elicitation of tacit knowledge by designer [113]. Action, interaction, and communication are essential components of the designer's thought process. To understand designer thinking and the design process, it is necessary to study how designers interact with each other and their environment.

Such studies, often falling under the broader category of *protocol studies* involve recording the design session on video and audio, followed by transcribing, segmenting, and coding this data. Dinar et al. [15] cover such analyses in their detailed review of empirical studies of design, including traditional protocol analyses that involve coding designer utterances and actions. Typically verbal data forms the basis of such analyses, such as think-aloud protocol analysis where a designer verbalizes her activities, giving insight into her thought process [14], or latent semantic analyses of a design team discussion to throw light on coherence between team members [114].

In addition to audio and video data, protocol studies often also involve analysis of sketches, notes, and other "marks-on-paper" [10, p.269] that are integral to designer behavior. To designers, sketches function as external memory, or as a medium to think. To analysts, sketches provide additional information about designer thinking— as an externalization of thought, and as a log of their thought sequences. Qualitative analyses of protocol studies—both in-situ and in-vivo—often require making sense of multimodal data, often with a temporal component. This data can include, but is not restricted to, audio/video recordings, artifacts created by subjects, or electronic records of user activity. Qualitative studies of this nature often involve the use of

Computer-Aided Qualitative Data Analysis Software (CAQDAS). Such software are suitable for analyzing and coding traditional qualitative analysis data such as video, transcripts, and documents, but are not flexible enough to accommodate newer forms of data. These include position and movement data from smartphones or sociometric sensors, or psychophysiological measurement devices such as electroencephalography.

Analysis of verbal data requires manual transcriptions of participant verbalizations by transcribers—often lacking domain knowledge—provided with a relevant glossary of terms. These transcriptions and segmentations then need to be verified by employing multiple coders and this verification by itself needs to be consistent across coders. The reliability of a coding process is further compromised and complicated when one is analyzing, say, a design discussion within a team [115], which involves audio/video records of the discussion, sketches generated by the team, and other user log data that is collected. Furthermore, the design researcher may not be the one coding the data, although they may need to expore the data to identify questions that need answering. Dinar et al. [15], in their review of 25 years of research in design theory and methodology, suggest the use of automated data collection and analyses to process the increasing volume and heterogeneity of data collected in such studies.

In our attempt to address these challenges, we look to the field of visual analytics for inspiration. Visual analytics (VA) is "the science of analytical reasoning facilitated by interactive visual interfaces" [116, p. 4]. Specifically, it focuses on interactive visual representations that exploit the human ability to spot patterns and anomalies, while allowing the computer to process large datasets. Information visualization (Infovis), defined as a "graphical representation of data or concepts" [117, p.2], naturally forms an essential part of VA. Keeping these needs and possibilities in mind, we present VizScribe, a web-based framework that supports the representation of multimodal data, including traditional as well as newer forms discussed earlier. The framework allows analysts to generate appropriate visual representations of any specific temporal datasets, and link them to the existing video and transcript displays. VizScribe supports interactive, customizable brushing and linking between different

data representations, allowing analysts to easily and intuitively explore the data and interactively code it. Our goal is twofold: (a) to enable the analyst to explore design study data in context and find patterns and anomalies, and (b) allowing the analyst to query the data to identify further clues behind these patterns and anomalies.

We performed formative studies with both open and prescribed coding tasks in order to refine the framework. We then performed a summative study with 10 participants who were tasked with analyzing video, transcript, sketch, and sensor data from a 60-minute design session. Our studies showed that timeline and text views of the transcript, and the timeline view of sketching activity were the most versatile visualizations, while word cloud visualizations were used as a filter to identify salient parts of the transcript. VizScribe allowed analysts to orient themselves with a medium that they found most comfortable—sketches, transcript, or the video of the design session shown—with little difference in the task outcome or user feedback. Our studies showed that VizScribe allowed analysts both with and without training in CAQDAS tools to make sense of given data, enabling them to perform closed- and open-ended tasks with ease.

In the following sections, we review related work, motivate the need for such a tool, and detail its design and implementation. We then describe the user studies, and participant feedback, finally highlighting challenges and future work.

## 6.1   Design Rationale: Visualizing Protocols

Newell [118, p.1] defines a protocol as "a record of the time sequence of events" that also includes "continuous verbal behavior of the subject" in the context of problem solving in a think-aloud setting. These transcript of verbal behaviors are studied, annotated, and coded by design researchers, in order to form and test hypotheses, and to answer research questions [66]. Such annotations and codes are created by first identifying the act, and if possible, identifying the *time* of occurrence of the act. The analyst's challenge is to make sense of multiple time series data or observations,

by linking them to transcripts of designer verbalizations, video captures of designer actions, and artifacts such as sketches or prototypes created by the designer.

Our layout design and the interactions afforded by our framework are informed by these aspects of protocol studies. With the increasing prevalence in the use of software tools for design and the use of sensors such as accelerometers, gyroscopes, and bluetooth devices to track collaboration dynamics in teams [119], protocol data have become richer and more varied. Making sense of these multiple forms of data thus depends on two major factors: the way in which these datasets are displayed to the user—the analyst performing the protocol analysis—and the way in which the user is able to interact with these datasets. Both of these requirements are commonly encountered when designing Infovis and VA tools and will thus benefit from principles derived from these fields.

VizScribe is designed essentially as a cohesive collection of *coordinated multiple views* [120, p.12]—linked views of related datasets that enable the viewing of the data from multiple perspectives. The rationale and requirements behind this framework are detailed in this section.

**R1** *Spatial Coherence:* Visually representing time sequences of events, including verbal behavior, needs a spatially coherent alignment to make sense of these sequences. Position is a widely accepted encoding of quantities occurring on a scale [121], the scale in this case being time. All timeline data pertinent to a video-recorded protocol study session must thus be presented coherently with the video timeline.

**R2** *Multiscale Representation:* The substantial data obtained from protocol studies can benefit from multiple scales of representation: of time, content, and detail. These allow the user to identify patterns and form meaningful connections between data. This "overview+detail" approach [122] is a fundamental requirement of interactive visualizations.

**R3** *Context Awareness:* Connections between different representations of data need to be made evident to the user as needed. For data from protocol studies, this is especially important in order to provide context. This is also useful when the analyst is trying to understand if there are patterns of behavior evident in the coded sets of data.

**R4** *Interactivity:* Making sense of multiple datasets requires fluid interactions with their representations. These interactions should allow the user to *orient* themselves to data visualizations, *focus* on specific elements of the visualizations, and then *code* the data. Elmqvist et al. [123, p.336] emphasize the importance of "providing immediate visual feedback on interaction" for both major and minor interaction events. They also recommend allowing the user to interact directly with the visual elements.

**R5** *Extensibility:* Given the ever-changing nature of data collected in protocol studies, such visualizations need to be extendable to accept newer forms of data, mapped to appropriate visual representations.

## 6.2    The VizScribe Framework

VizScribe's design addresses the previously-established requirements of spatial coherence, multiscale representations, context awareness, interactivity, and extensibility. The two main aspects of this design are the *layout*—determining how the data is represented to the user— and *interactions*—determining how the user explores the data. We detail both these aspects here.

### 6.2.1    Layout

VizScribe's layout is divided into two main sections, shown in Figure 6.1:

- A *temporal view*, where all time-series data is shown in congruence with the video of the design session being studied, and
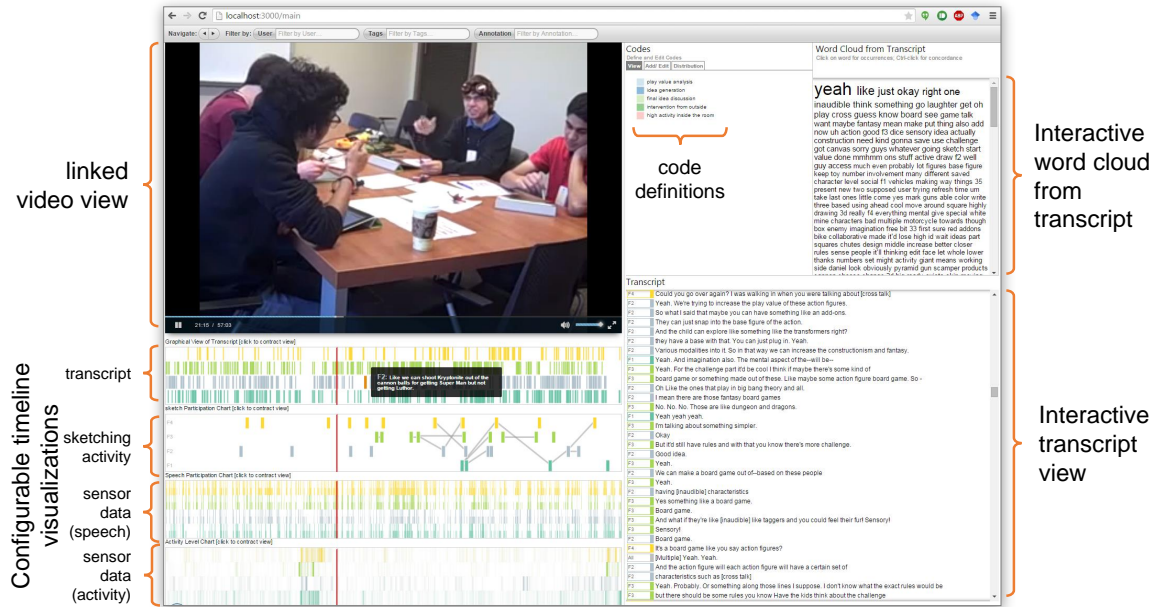
Figure 6.1. A screenshot of VizScribe. The application runs on a web browser, with video, timestamped transcript, user log file, and associated images uploaded to a server. The interface displays this data in the form of timeline-based views, such as video progress, transcript visualization, and sketch timeline. Interactive views such as a word cloud and the transcript provides the user with an "anchor" for navigation. The user can jump to a specific part of the video by clicking the transcript, and can select a line to assign it a specific code. These codes are interactively defined, and the "coded timeline" updates dynamically as the user continues coding.

- A *transcript view* where the transcript and other transcript-related data including word frequencies and codes are shown.

### 6.2.1.1 Temporal View

Video data forms the central reference of this view, providing context to any of the other forms of temporal data. Visual representations of time-sequenced data should be designed to answer the question of "what was happening when...?", by providing

a means to align all temporal data such that it provides context to the analyst. The temporal view pane thus includes a video playback interface, with time encoded as a progress bar spanning the width of the pane. All timeline data visualizations are stacked under the video interface. They are scaled to, and aligned with the video progress bar, providing spatial coherence (requirement **R1**).

The temporal view features an event-level interpretation of protocols, including a timeline representation of the transcript. Our visual encoding of these events is inspired partly from temporal sequencing visualizations by Isenberg et al. [124]. The transcript is encoded as a series of thin vertical marks as shown in Figure 6.1, annotated as the "transcript" timeline visualizations. Each horizontal strip of marks represent utterances of one speaker, identified in the transcript file. Multiple speakers or subjects are color-coded accordingly across visualizations based on Harrower's [125] qualitative scheme. This color coding is consistent across timelines, as seen in the figure. In the data used for this figure, sketching is performed on digital tablets and shared among users, allowing sketching activity to be logged. The visualization of this sketching activity is similar to the transcript timeline, with added lines denoting the modification or extension of previous sketches, inspired from Robinson et al. [126] and Zhao et al. [98].

This timeline visualization is also designed to be extensible, wherein the design researcher, with some programming background, can adapt existing new timeline visualization panels, or add new panels of their own to suit their requirements (**R5**). For example, the speech participation data and accelerometer data collected from personal sociometric devices [16] on each subject is visualized in the interface shown in Figure 6.1. This extensibility is explained in detail in Section 6.3. Finally, any coding performed on the transcript is also reflected in the timeline view, color-coded according to the user-defined codes. All of the above timeline visualizations are overlaid with a video progress indicator (Figure 6.1), visually linking them to the current time in the video.

### 6.2.1.2 Transcript View

The transcript view shows an interactive view of the transcript with corresponding text visualizations and annotations. Color-coded speaker identifiers provide a visual link to the corresponding timeline visualizations. In addition, within the transcript view, a *word cloud* view is generated from the complete text of the transcript after stop word removal. The words are scaled in proportion to the frequency of their occurrence in the text, helping identify recurring terms in the transcript. The transcript timeline view, text view, and word cloud together present multiple resolutions to view participant utterances within the transcript view (**R2**). Finally, a "code view" section within the transcript view allows the researcher to define and edit codes. A hierarchical code definition is enabled through tabbed entries in the field, thus supporting open coding. The code view is color-coded as well, and any transcript assigned to a particular code can be identified through these colors (**R3**).

Each visualization embedded in these views is linked to related visualizations, shown implicitly through color and position, and identified explicitly through user interactions. These interactions are described in detail in the following section.

### 6.2.2 User Interactions

Fluidity of interaction and directness with which the user can manipulate the data representations separate an effective VA system from an ineffective one. Guidelines for designing such interactions have been determined with respect to mapping user intent [127] as well as fluidity of interactions [123]. The interactions designed into VizScribe follow requirement **R4**, described in detail below:

- *Orient:* This is the act of the user familiarizing themselves with the data representations without explicitly changing its state. We map such actions to *hover* events: the dwelling of the mouse pointer on any element of the visualization to "raise the attention of the computer as a dialog partner" [128, p.561].

- *Focus:* This is the act of the user focusing their attention on one element on the data representations. We map such actions to "selection" events (mouse click and associated interactions) that change the state of the visualization.

- *Code:* This is the act of the user defining a code—a category of behavior or utterance—and/or assigning that code to a part of the transcript.

| Orient | Focus | Code |
|---|---|---|
| Hover over visualized items to identify connections | Investigate connections by filters or detail views | Categorize items of interest that follow a pattern |
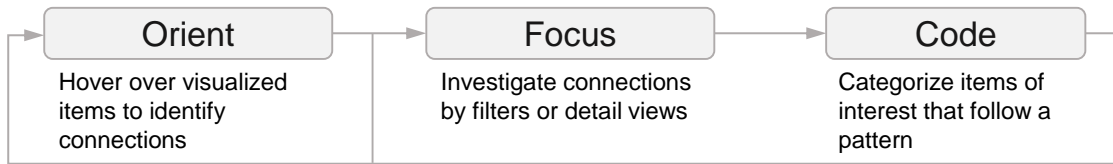
Figure 6.2. The flow of analysis designed into VizScribe. The user initially *orients* themselves to the various visualizations, using hover operations to explore connections. They then *focus* on items of interest, and finally begin *coding* these items into categories. The resulting coded visualizations are then explored for further insight into the data.

The user's process of analyzing the visualized datasets involve moving from one of the above acts to the other, as depicted in Figure 6.2.

### 6.2.2.1 Orient

Figure 6.3 shows some of the ways in which users can orient themselves to the data. Hovering on a line of the transcript highlights the corresponding mark in the timeline representation of the transcript, and vice versa. Similarly, hovering on a mark in the coded timeline view also highlights the corresponding line on the transcript. Hovering on a word in the word cloud highlights all the lines on which the word occurs, and shows a filtered view of these words in the timeline view (Figure 6.3(B)). Hovering on the sketch timeline view described in Section 6.2.1.1 display thumbnails of the corresponding sketches (Figure 6.3(C)).
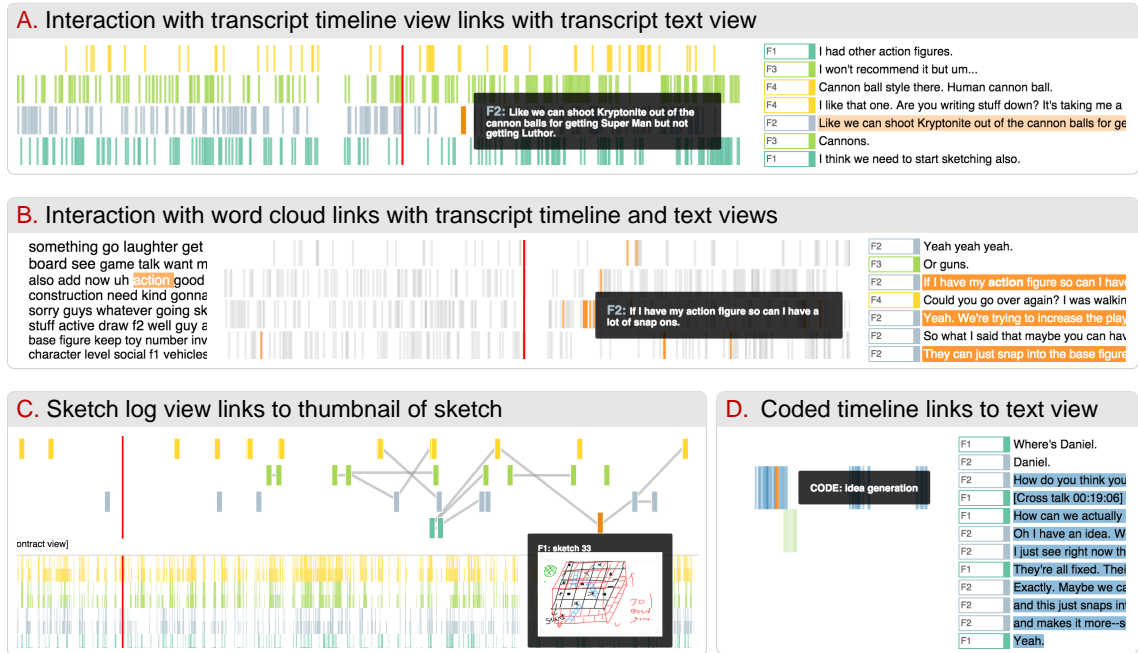
Figure 6.3. Various forms of brushing and linking used in the framework, to facilitate the envisioned *orient* and *focus* tasks. (A) shows the transcript text on the right, linked to a time-sequence representation of utterance "events" color-coded by speaker ID. Hovering on an element in the time-sequence view highlights the corresponding text in the transcript view, and vice versa. (B) shows similar interactive linking, but this time, hovering on a word in the word cloud shows all its occurrences in the time-sequence and text views. (C) shows the sketch log view, where each "event" represents a sketch save operation. A hover event on this view shows a thumbnail of the saved sketch. The interactions in (D) are similar to *A*, except the time-sequence view shows all utterance events assigned to a particular user-defined code.

These interactions provide a means to ensure that the user is not overwhelmed with the multiple data representations. They also allow the user a means of checking if they are at a point of interest in the dataset, before performing an overt interaction to *focus* on that point for a closer examination.

### 6.2.2.2   Focus

In protocol studies, context is one of the main relations that needs to be examined. Yi et al. [127] describe the act of "selection" in Infovis as marking a data item of interest to keep track of it. Selecting an item in the temporal view has the effect of connecting this element to the video timeline, by skipping the video to the timestamp corresponding to the visualized element, to provide context. The timeline indicator on all other temporal views also skip to this timestamp, providing a visual cue of other temporal co-occurrences. Correspondingly, the transcript view scrolls to a participant utterance closest to this time, highlighting what was said close to the instant of interest.

In the transcript view, selecting any word from the word cloud persistently highlights all corresponding lines in the text and timeline view of the transcript. This allows the user to scroll around on the transcript, or skip to the timestamps of interest in the video to examine the data for patterns. Selecting a line on the transcript, when done through a keyboard combination, skips the video to the timestamp corresponding to that line, allowing the user to view any activity of interest occurring around the time. It also allows the user to check for any temporally co-occurring item of interest in the timeline visualizations.

Such interactions also allow for the *filtering* and *details-on-demand* tasks specified by Shneiderman [122]. Filtering is the removal or de-emphasis of uninteresting items, allowing users to focus on their interests. Details-on-demand refers to obtaining details of a particular selection, usually displayed on separate pop-up windows. In VizScribe, filtering is achieved in several ways: selecting a block of text in the transcript updates the word cloud to contain words from only the selected block. This filtering is extended to speakers and coded sections of the transcript: selecting a speaker (with a keyboard shortcut) on the transcript timeline, updates the word cloud to only show words spoken by that speaker, while selecting a code from the coded timeline view updates the word cloud in the same way. VizScribe has two forms
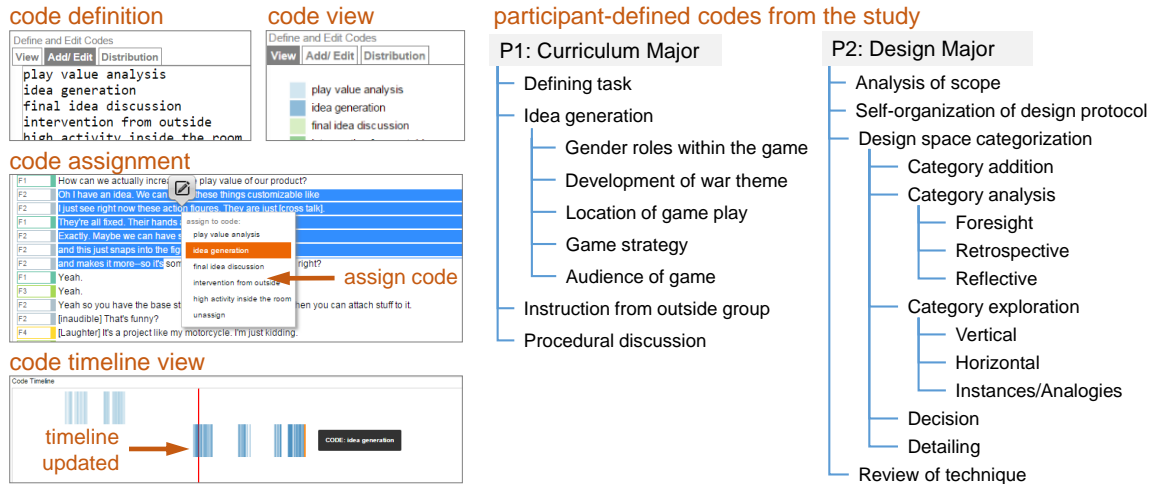
Figure 6.4. Coding in VizScribe and sample codes generated by participants from our formative studies. The screen captures from VizScribe on the left show the code definition and code assignment. The hierarchical codes on the right were defined by the participants (P1 & P2) in the formative study. Though using the same dataset for the coding task, the participants produced very diverse codes, based on their research backgrounds (curriculum development vs. design).

of details-on-demand tasks: selecting a word in the word cloud (through a keyboard shortcut) generates a *keyword in context* view showing text immediately preceding and succeeding every occurrence of that word, allowing the user to identify patterns in the context of utterance of that word. A similar selection of an item in the sketch activity timeline shows a larger view of the sketch in a separate window, allowing the user to examine its details.

### 6.2.2.3 Code

Typically, this data exploration is followed by the categorization of utterances and behavior, and coding or annotating these with the identified categories. VizScribe is provided with a text-entry field for determining such code. Hierarchies of code are encoded by the user through the use of tabbed indentations in the text entry. Once a

code is defined, they are assigned to a selected block of text through a context menu. A unique color for each code allows the coded transcript text to be highlighted on demand, as explained in Section 6.2.2.2. At any stage, the coded transcript can be exported as a comma-separated value (CSV) file for use with other tools. Figure 6.4 shows the coding process and sample codes created by two participants in our formative studies. The diversity of the code generated and the hierarchies formed show how VizScribe supports open, axial coding practices.

## 6.3  Implementation

The introduction of visualization libraries such as D3.js [129] has resulted in the web browser emerging as an appropriate platform for Infovis, for both collaboration and dissemination. We chose to implement VizScribe as a web-based framework, with the data uploaded and stored at the server, and then processed and visualized at the client end. This implementation has the added advantages of platform independence, minimal installation requirements, and for extension to a collaborative VA framework.

The application is hosted from a Node.js[1] server. Design activity data can be uploaded to this server by an analyst using VizScribe to create a coding session. The data is cached and hosted for all client web browsers running the VizScribe interface. Almost all data processing into structural and graphical elements for the visualizations occurs at the client end. All visualization elements in VizScribe have been developed using HTML5 and JavaScript, with the visualizations generated primarily using the D3 library. The video is rendered using video.js[2], an open-source web-based player.

The data processing pipeline is shown in Figure 6.5. Input data is first uploaded to the server in the form of video and a transcript file, with additional time-stamped data including, but not restricted to sketch logs and corresponding image files of sketches, instrumentation logs and activity logs. While it is essential that the transcript file includes timestamps, speaker identifiers for each line of spoken text are optional. The

---

[1]http://nodejs.org/
[2]http://videojs.com
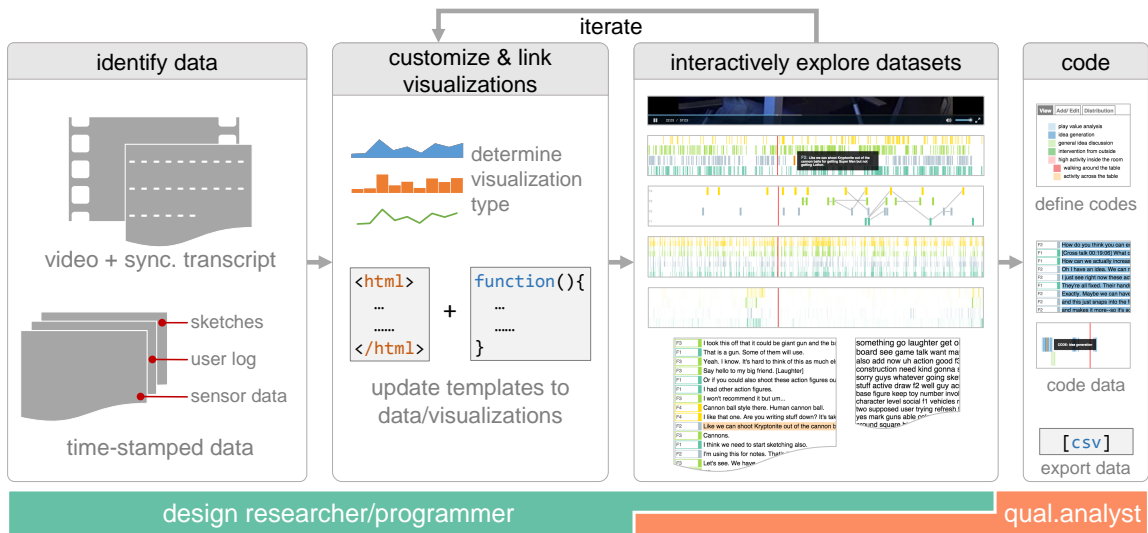
Figure 6.5. Visualization pipeline used in the VizScribe framework. The standard inputs (required) are in the form of a video and a timestamped transcript. The VizScribe web application generates a default timeline and word cloud visualization for the transcript. Predefined code "templates" cater to other timestamped code, where the researcher with some programming know-how, can iteratively customize these templates to explore different visualization forms and ways to link to the transcript and video. Once the visualizations are finalized, the qualitative coder can take over, coding the transcript and exporting the coded data to a comma-separated value (csv) file.

current data import requires transcripts to be in a comma-separated value (.CSV) format, but is extensible to include other formats such as the SubRip (.SRT) format or closed captioning formats.

All log files, including sketching activity logs, need to have timestamped entries, and corresponding references to additional media where applicable, such as sketches or images. These additional media files, which can include images, vector graphics, and notes, are uploaded as a single archive. This upload mechanism, showcased with sketches in Figure 6.5, is extensible to the other media types.

The transcript is reformatted into objects as specified under the *Document Object Model* (DOM), ensuring cross-browser compatibility. In addition, this allows each
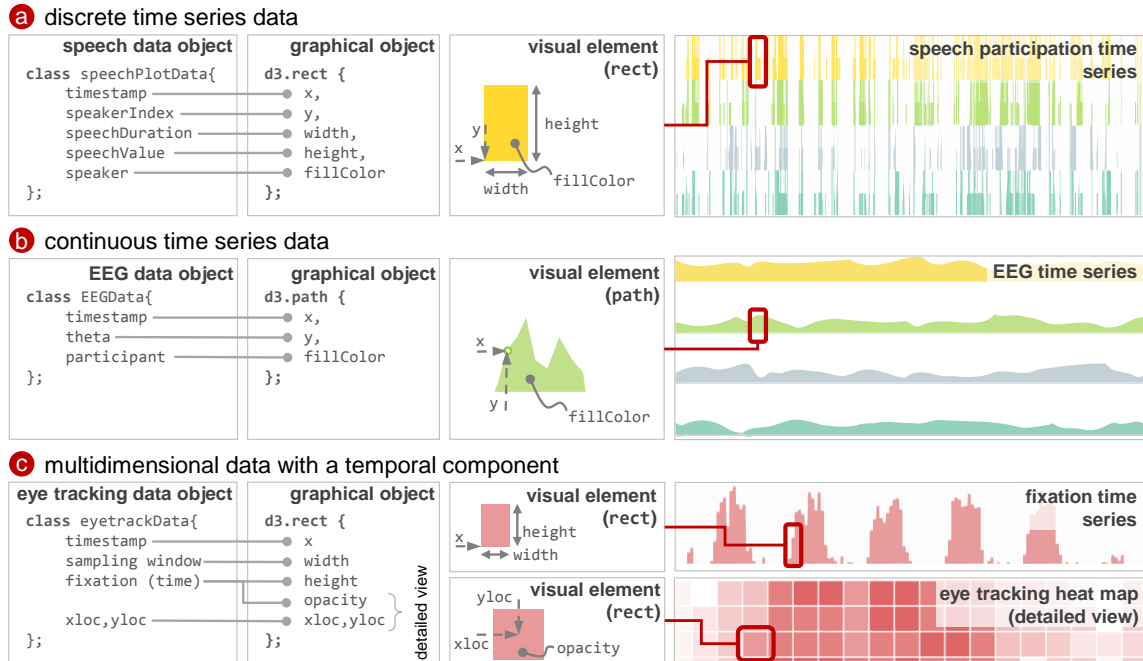
Figure 6.6. Extending or customizing a timeline visualization to fit custom time-series data involves mapping attributes of that data to a corresponding visual object. VizScribe uses D3 for this purpose, whose data structure makes this mapping possible. The above figure illustrates how this extension is possible for three main categories of data: (a) discrete time series data where data is sampled at intervals, (b) continuous time series data where data is read in a stream, and (c) multidimensional data with a temporal component, where a time series visualization needs to be augmented with additional visualizations. For all three categories, the above figure illustrates ways in which the VizScribe timeline views can be extended to incorporate such data by mapping data attributes to geometric attributes of appropriately chosen D3 elements. Hover/click behaviors are then specified, allowing for interaction with the data.

object of the DOM to be programmatically accessed. VizScribe uses the JQuery library[3] to easily access and manipulate DOM elements, and implement the interactive linking between objects as discussed in the previous sections.

[3]http://jquery.com

Timeline entities are parsed and mapped to attributes of D3 graphical objects, aligned with the video timeline. We take advantage of D3's *data* operator, a format-agnostic array of values that are linked to visual elements displayed on the screen. Figure 6.6 shows how a timeline dataset is mapped on to a D3 object. This is also the means behind VizScribe's extensibility: new time-series datasets can be represented in an appropriate visualization by adapting any of the existing time-series visualization classes. We identify three kinds of data that would typically be used in protocol studies: discrete time series data measured over intervals of time, such as speech participation measures, continuous or streaming data such as EEG sensor readouts, and multi-dimensional data that have temporal components, such as eye-tracking data or user behavior logs that require additional visualizations in addition to time-series visualizations. Figure 6.6 illustrates how, in each of the above three categories, data elements can be mapped to appropriately chosen geometric elements. Standard event controllers area available that can be changed to determine actions to perform when hover or select event occurs. Selecting an appropriate visualization is an iterative process. By experimenting with the various geometric entities available in D3, appropriate visualizations can be explored. We envision VizScribe to be used by design researchers with some programming background, so that they may adapt the visualizations to forms of data pertinent to their work. Using the existing templates, the data upload page, as well as corresponding client and server-level script templates can be updated to receive new datasets. In order to aid this extensibility, we will provide VizScribe as an open-source software.

### 6.3.1 Collaborative Coding

The idea behind developing VizScribe as a web-based framework was to open up the possibility of collaborative coding. To implement collaborative coding, we used the WebSocket protocol[4], which provides full-duplex communication channels over a

---

[4]https://www.websocket.org/
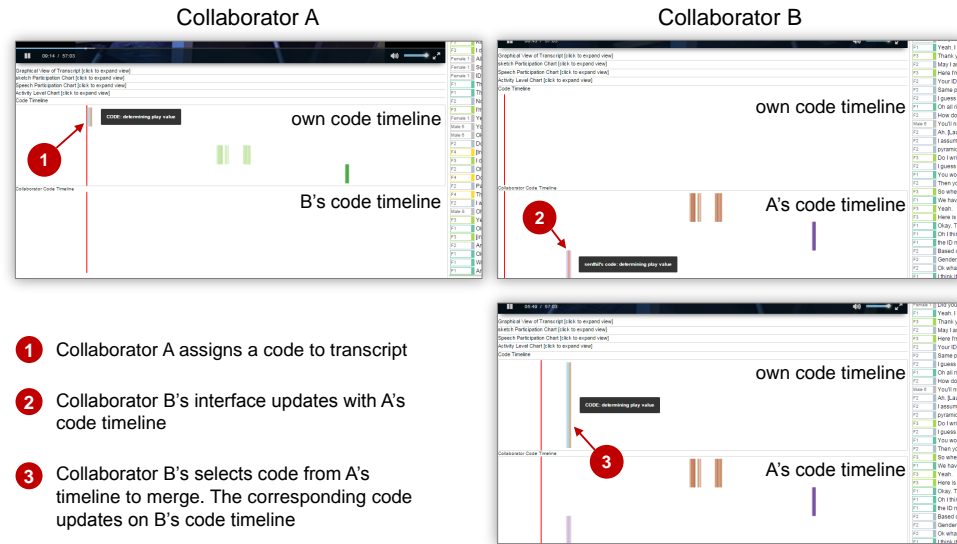
Figure 6.7. Collaborative coding prototype for VizScribe, implemented using the WebSocket framework to enable real-time updating between client(s) and server. In the example shown, collaborator B sees collaborator A's code timeline, and selects a range of code assignments, which are automatically merged into A's timeline.

a single Transmission Control Protocol (TCP) connection. The WebSocket protocol allows a live interaction between client(s) and their server. In VizScribe, this allows the creation of an additional timeline view of a "collaborator code", that updates in real-time whenever any one of the clients assigns or unassigns a new code. Selecting a range of `rect` elements from the collaborator's code timeline "merges" the selected codes into one's own timeline, as long as a code with the same name is defined in advance. Figure 6.7 illustrates this real-time update and merging of the code timelines between two collaborators. While the prototype is implemented for two collaborators, this protocol can be scaled to include any number of clients, subject to server traffic and client processing power.

## 6.4    Evaluation

The goal of VizScribe is to provide a platform for design researchers to view and explore coordinated representations of multimodal data that they have collected through protocol studies. VizScribe also aims to fill a gap that existing CAQDAS tools do not address: the development of custom visualizations to incorporate emerging forms of data that are collected to support such studies. However, a comparative study with an existing CAQDAS tool would not be productive, as these tools are general-purpose, used for analyzing documents, images and multimedia files. They would have little support for data such as user logs or sensor data. This was confirmed with a pilot study with two practitioners of NVivo, a commercially available qualitative analysis tool. We thus focused on evaluating VizScribe's usefulness in qualitative analyses, and on gauging the effectiveness of the customized interactive data visualizations that other tools do not provide. We performed two sets of studies.

- *Formative evaluation*: This study sought to identify weaknesses in the design and implementation of VizScribe, which helped us understand what forms of interactive visual representations were useful, ones that were redundant, and others that needed enhancement. We performed two sets of formative studies:

  - *Prescribed Coding:* We studied how, given the same coding task, participants navigated the interface.

  - *Open Coding:* We studied how flexible VizScribe was in the hands of two analysts with diverse analysis goals.

- *Summative evaluation:* This helped us understand the versatility of the interface in answering certain closed- and open-ended questions of the kind that interest design researchers.

For all the above studies, a brief demonstration of VizScribe and its features was provided to the participants, after which they were allowed as much time as they

needed to familiarize themselves with the interface. Before both studies are explained, Section 6.4.1 on context will set up the datasets used for the studies.

## 6.4.1  Context: The Design Sessions

Given our goal to better understand design processes through various forms of data collected during the process, we recorded two student teams from a mechanical engineering graduate course on product design. The teams were recorded when working on a design modification assignment that required the team to first categorize a given toy according to its "play value" [90], and then modify it to extend and/or change its play value. The teams used skWiki [98]—made available as open source[5] by its creators—to create, exchange, and modify sketches of their ideas, enabling us to log their sketching activity. The sketch log data from skWiki was used to create a timeline visualization of their sketching activity described earlier in this document. Both sessions lasted approximately one hour, and were video recorded with the consent of the participants. One team was also provided with wearable sociometric sensors [119] to record their speech patterns and movements for the duration of the assignment. We used a 15-minute video segment and sketch data from the team *without* sociometric sensors for the formative evaluation, and the complete 60-minute segment and associated data from the team *with* sociometric sensors for the summative evaluation. This decision was in accordance with the goals of the evaluations—with the formative evaluation focusing on studying the efficacy of providing temporal and transcript views for coding, and the summative evaluation for understanding high-level user behavior trends.

The formative study was conducted on a Dell Optiplex 760 desktop, running a Windows 7 operating system. The summative study was conducted on a Lenovo Y460 laptop, also running Windows 7. In both studies, VizScribe was run on a Google

---

[5]https://github.com/karthikbadam/skWiki

Chrome browser, and displayed on a 20-inch LCD screen. The data was uploaded and rendered on VizScribe before the start of the experiment.

### 6.4.2 Formative Study

The formative evaluation of VizScribe was split into two studies as mentioned earlier: one to understand, given a prescribed coding task, how the participants navigated the interface, and the other to determine how flexible the interface was to do a diverse set of open coding tasks. Both studies used the same dataset, mentioned in Section 6.4.1.

#### 6.4.2.1 Prescribed Coding

We recruited 6 graduate students (5 male, 1 female) as paid participants for this study. Participants were aged between 27 and 32 years. All participants were experienced in one of two processes: 3 were communication students, and thus were experienced in ethnographic studies involving similar coding processes, and 3 were engineering design students, experienced in design processes. Two of the participants had prior experience in protocol analysis.

Participants were given 60 minutes to use VizScribe, during which they were required to familiarize themselves with the data shown on the interface, and identify and code instances of *idea generation*, when a subject in the video comes up with an idea for the toy design, *idea extension*, when a suggested idea is modified or developed, and *idea critiquing*, when feedback is provided to a suggested idea. We asked them to code segments of the transcripts as such.

At the end of the session, participants were asked to answer a set of questions eliciting feedback on their experience with the interface. This included both open-ended questions on their overall experience and feedback, and their evaluation of the usefulness and ease of use of specific features of VizScribe on a Likert scale.

### 6.4.2.2   Open Coding

We recruited two graduate students (both paid) for this task, one male and one female, aged 29 and 34 years. One of the participants, a curriculum and instruction student in art education, had prior experience with coding in NVivo, and the other, a student of mechanical engineering majoring in design, had no prior experience in coding or protocol analysis. They were asked to use an inductive open coding technique [130]—which was first explained to the participants—to code and then categorize their observations. Participants had 60 minutes to perform this coding task, followed by an open-ended survey where they detailed the coding process they used, the features they used in VizScribe, and their overall feedback. Our goal was to evaluate the utility and ease of use of VizScribe in a real-life coding scenario, and identify challenges that arise in data representation and coding in such scenarios.

### 6.4.3   Summative Evaluation

The formative studies helped us make further design and implementation refinements described in detail in Section 6.5. We then performed a more detailed study with tasks that attempted to ask questions with more ecological validity. Summative qualitative evaluations are useful when the emphasis is on understanding high-level behavioral trends, and mental models of participant behavior [131]. In addition to video, transcript, and sketching data, this evaluation used sociometric sensor data that captured the *activity levels* (data from the accelerometer sensors), as well as the *speech participation* (speaking to listening proportion) of each participant.

We recruited ten paid participants for this study: five male and five female, all between 23 and 36 years of age. Six were PhD design majors in mechanical engineering, and two (one master's and one PhD) were engineering education majors. Of the remaining two, one was a master's student in computer information technology who had a background in teaching undergraduate design courses, and one was a post-doctoral researcher in educational psychology, focusing on design education.

Nine of these participants had performed qualitative analysis, ranging from document analysis, to interview coding, to analyzing videos of participant gestures. Of these, one had prior experience with CAQDAS tools.

All participants used VizScribe to perform 7 tasks centered around a 60-minute video of the session explained in Section 6.4.1, The 7 tasks were as follows:

**T1** Identify the sketch/sketches that represent the *final idea* chosen by the team.

**T2** Who was the team member that originated the idea that finally evolved into this final idea? Identify the member, and code the part of the transcript that refers to the first mention of this idea.

**T3** Did the team discuss other ideas before they narrowed down on one final idea? If so, describe these ideas.

**T4** Code the parts of the transcript that you identify as the start and end of the team's divergent process (ideation) and the start and end of the team's convergent process (evaluation and selection).

**T5** Using the word cloud, identify themes in each team member's utterances.

**T6** Identify and code three instances with no speech overlap between team members, and three instances where there is speech overlap. What behavior differences can you observe between the two categories?

**T7** Using the activity timeline, identify with timestamps three unique instances of activity or movement, and describe them.

Tasks T1 and T2 have components that can have "correct answers" or answers that are comparatively less open to interpretation. We used these tasks as "ground truth" tests, to determine if the VizScribe interface helps the participants glean such information. Tasks T3 through T7 are more open-ended and were intended to study

how VizScribe's representations of the customized datasets are useful in understanding aspects of the presented data that are not immediately apparent from just the video and the transcript.

## 6.5 Results and Discussion

Results from formative studies helped identify features that were useful in exploring multimodal data, and features that needed refinement. The summative study helped understand the different means by which analysts approach design protocol studies, and the versatility of the various visualizations used in VizScribe in catering to these different means. The results from both evaluations and their implications are detailed in this section.

### 6.5.1 Formative Evaluation and Subsequent Redesign

Participants chiefly liked the way multimodal data was presented, and found it easy to navigate in the interface. One participant summarized it well: *"(VizScribe is a) useful tool for observing multiple modes of data. I was able to relate the transcript to actual body language from the video and sketching activity. I liked how I needed minimal instructions to use the tool. A browser based environment really helped, as I was familiar with most browser based interactions."*

Participant feedback, obtained both quantitatively on a 5-point Likert scale, and qualitatively as comments, revealed that the interactive transcript was the most useful feature of VizScribe (reported by all 6 students). The word cloud was deemed the least useful (by 4 out of 6 students), partly due to the small size of the data (participants found it easier to read the transcript of the 15-minute video), and partly due to a lack of a context for using this visualization. Among the temporal views, the sketch timeline was deemed intuitive (by 5 out of 6 students), but not relevant, partly because interactions with the view were not linked to other views at the time of the study, and partly because of the small size of the data. The coded timeline was the only

other timeline view for this study, and participants emphasized the need to assign multiple codes, as well as better granularity (at the word level) of code assignment.

In the open coding study, both participants found it easy and intuitive to create multiple hierarchies of codes. However, the representation of codes on the transcript started to become problematic when numerous codes were created. The color-coded display seemed to become cognitively difficult to process when there were 10–12 colors, one for each code, painted over the transcript. Finally, by comparing the codes generated by both participants, we found that both viewed the same dataset for approximately the same duration, and yet created markedly different codes (Figure 6.4).

#### 6.5.1.1   Redesign

Based on the observations made from the formative evaluation and from the participant feedback, enhancements were made to some of the visualizations.

*Scale-Robust Timeline View:* The transcript timeline visualization became less readable when longer videos of around 60 minutes were used. The visualization was thus extended to indicate, through color codes and position, the different speakers in the visualization. A "magic lens" local zoom was added to the view for easier selection of a visualized element.

*Text Filters:* The word cloud view was made more useful by introducing more techniques for "focus" tasks explained in section 6.2.2. This included the filtering tasks explained in the section, such as word clouds specific to one speaker's utterances, word clouds specific to certain coded parts of the transcript, or a dynamic word cloud that updates itself to a selected block of text.

*Interactive Code Linking:* The color limitation that was faced by using more than 12 codes was mitigated to an extent through enhanced interactions. The code colors on the transcript were made non-persistent, i.e. they "faded" two seconds after coding, so that transcript legibility was not compromised. Users could bring back the color on the transcript by hovering or clicking on a corresponding code in

Figure 6.8.  Task-wise usage of the different elements of VizScribe, aggregated over all participants. Views such as the word cloud, the transcript timeline, and the sketch timeline can be deemed the most versatile, since they are used across most of the tasks. The distribution shows that most of the exploration of the provided dataset occurs through the "sketch timeline": the view showing the creation and development of every sketch by the subjects in the video.

the coded timeline view. This ensured that the transcript was overlaid with only one code (color) at any point of time, making it visually easier on the user.

*Extensibility:* Most of the extensibility discussed in section 6.3 was developed as a result of this study, based on participants suggesting various datasets that could be connected to the VizScribe timeline.

The summative study was conducted after implementing the above changes.

### 6.5.2   Results and Implications from the Summative Study

Our choice of tasks for this study aimed to help us understand the kinds of visualizations that are applicable to certain tasks, and to determine the versatility of the visualizations, i.e., whether they are used for a varied set of tasks. Of the "closed-ended" tasks, all ten participants had correct answers to T1, while eight participants got T2 right. The two who had a different answer were not necessarily wrong: they chose the participant who had *sketched* the idea, and not the member who *originated* the idea. The consistent answers to these tasks confirm the veracity of the data representations.

Using participant activity logs collected over the duration of the study, we can see that the transcript timeline and the sketch timeline were clearly the more versatile views: they were used more in tasks T1 through T4 (Figure 6.8). To a lesser extent, so was the word cloud, used sparingly during these tasks. Tasks T5, T6, and T7 were *designed to* require the use of the word cloud, the speech participation timeline, and the activity timeline respectively. The most interesting interactions of the participants involved filtering and navigating text views, analysts using different visualizations for the same task, and observations of the video-recorded behavior by interacting with activity sensors.

*Filtered Navigation:* We observed participants using the word cloud as a filter for the transcript view, especially when performing a focused search for an idea. In completing T2, for example, the participants, having previously identified that the final product was a board game, used the word cloud and its link with the transcript (text and timeline) to filter it for all occurrences of the word "board". One participant explains: *"there were many cases where I saw an interesting pattern on the timeline, watched the video, read the transcript, found a word, and then saw where else it occurred in order to see if there was a pattern"*. By navigating around these filtered sections of the transcript, they were able to identify the right times. For task 5, where the word cloud use is explicitly required, the participants were required to identify themes from the word cloud of each video subject's utterances. However, in addition to this, three participants tagged the same subject as the leader of the team captured in the video—an unexpected and interesting use of the word cloud. This form of subject characterizations through a summary view, while convenient, can be antithetical to the rigor of good coding practice. This is the space that the VizScribe framework is designed to occupy: it allows the analyst to skim through a larger dataset, identify questions of interest, and perhaps focus on these questions while taking a closer look at the details.

*Multiple Approaches to Explore Datasets:* Perhaps the most important behavior we observed was that different participants had different ways to orient themselves
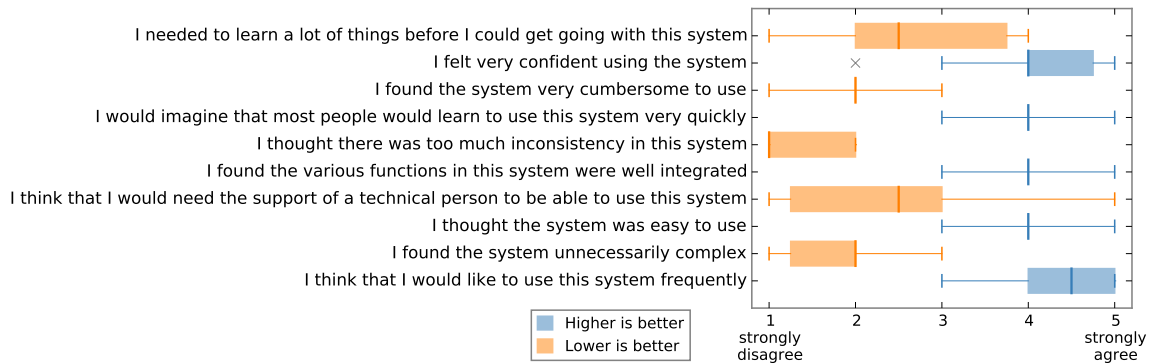
Figure 6.9. The System Usability Study (SUS) scores shown category-wise, aggregated over all participants. Plots in orange are better when lower, while plots in blue are better when higher. An overall score of 75.5 was obtained.

around the dataset, and picked different "entry points" into the dataset. Most notable were the use of the transcript timeline view, the transcript (text view), and the sketch timeline view. This makes sense: conceptual design predominantly involves sketches and explanations of these sketches, when working in a team. Providing representations of both these activities to the analysts allows them the flexibility of choosing the data representation based on the research question being asked. It would thus seem that it is not just the dataset that determines the most useful representation, but also the *user* who determines it. While this may seem obvious, this insight is often missing from most CAQDAS tools: there are few alternate representations of the same dataset.

*Temporal Correlation of Multiple Data Streams:* Task T7 required participants to pay attention to the activity timeline view, and note unique instances and corresponding subject activity in the video. This was done to raise the participants' awareness of the usefulness of such representations. Sure enough, participants found that they could not only make note of larger events such as subjects leaving the room, but also smaller instances of them leaning across a table, or even picking up a toy

from the table. One participant remarked *"traditionally, it has been difficult to track down the correlation between verbal behavior with gestures, sketching behaviors. With the help of these features, researchers would have a much easier time to pinpoint the relations between these behaviors."*

Overall, the framework was very positively received by the participants. In addition to their feedback, they were also required to respond to the System Usability Scale or SUS [132]. VizScribe scored an overall 75.5 out of a possible 100 points. The individual break-up of the scale, along with the participant score distribution is shown in Figure 6.9. The participant with prior experience with CAQDAS tools summarized it best: *"The tool is a great synthesis of what other tools have been missing, as I have used them."*

### 6.5.3   A Note on the Data Analysis Spiral

The process of qualitative data analysis is aptly described by Creswell [133, p. 182] as a "data analysis spiral". This spiral represents organization and management of data, reading, reflecting, annotating the data, identifying and comparing contexts and categories, interpreting and describing, as well as representing and visualizing the data. This is an iterative process, involving alternating deep dives into the data, reading it and creating memos (reflective notes) to get a sense of the data as a whole, examining it in parts to identify patterns of interest, as well as "winnowing" it to select information of relevance. Drisko [134] emphasizes how qualitative data analysis software has made possible research endeavors into multimedia data, by allowing the display of images, audio, and video, and the relationships between data that can be inferred or specified directly on images or indirectly on video and audio timelines.

Just as the use of multimedia added a richer layer to qualitative analysis, sensors such as eye-tracking sensors, EEG sensors, and inertial measurement units provide data that allow for richer research questions to be answered. VizScribe provides a general framework for the display of such datasets, allowing the researcher to create

meaningful visualizations of such data (as shown in Figure 6.6), and link it to traditional multimedia data. We saw in Section 6.4.3 how this enabled functions such as *filtered navigation*, *multiple approaches to explore datasets*, and *temporal correlation of multiple data streams* that allowed participants to access and make sense of such sensor data. VizScribe thus provides computer support for qualitative analysis in the light of emerging techniques of data acquisition.

It bears stating that qualitative data analysis *requires* that the analyst familiarize themselves with every bit of the data. In this sense, VizScribe is to be seen as a framework to allow a deeper study of this data and its connection to other related data, and not as a means of "skimming the surface". While it is possible to use VizScribe in the latter manner, it would defeat the purpose of qualitative analysis.

## 6.6 Limitations and Future Work

VizScribe has been designed for navigation and coding of data that otherwise cannot be easily visualized into a single, dashboard view. To an extent, this requires pre-processed data, for example, all data needs to be time-stamped, transcripts benefit from speaker identification, and so on. While the coded data can be exported from VizScribe for additional analyses, VizScribe would be a more effective tool if it offered end-to-end processing and analytics required for qualitative analysis. We envision the following future enhancements to VizScribe:

*Preprocessing:* This involves both the implementation of algorithms such as the Penn Phonetics Lab Forced Aligner [135] for synchronizing the transcript to the audio track, as well as allowing interactive selection and customization of the information visualizations used. The current implementation allows customization when performed programmatically. However, based on the data imported (keylogs, browsing data, biometric data), the user should be allowed to interactively select visualizations.

*Data-Aware Annotation and Coding:* VizScribe currently has the framework for allowing multiple timeline visualizations, but not timeline-level coding. This is par-

ticularly important for coding sketches, movements across the design space, gestures, and so on. Our future plans include a data representation of codes applied directly on the timeline, and consolidated together with all representations. Additionally, the absence of data—be it radio silence, video inactivity, or audio silence—is often as significant as the presence of data. Thus, gaps in the visualizations, where no geometric or text entity is displayed, need also be amenable to navigation and coding.

*Analytics:* The next step for VizScribe would be to incorporate visual analytics to process the data for a higher-level exploration. This includes, but is not limited to, features such as named entity recognition and tagging, parts-of-speech tagging, semantic distance-based filtering of the transcript based on the defined codes. Sociometric data would also benefit from such analytics to better support embodied and socio-material interactions. Additional use of analytics would be in processing the coded data to provide meaningful results. This can include inter-coder reliability calculations and axial coding, which includes the ability to generate matrices to compare instances tagged with intersecting codes.

*Dissemination:* With visualization toolkits such as D3, the web browser has already become a medium for both generation and dissemination of visual representations. Future work in VizScribe will also look at means to export the final data and analyses into reports or presentations.

## 6.7   Conclusions

We have outlined the requirements of design protocol analysis tools, emphasizing the need for building custom timeline views so that the analyst can set up their own visual representations of design activity. We then presented VizScribe, a visual analytics-based framework for representing and exploring design protocols. Our framework bridges a gap that is currently not addressed by existing qualitative data analysis tools, namely, the processing and presentation of new and emerging forms of data such as sensor data and user log data. VizScribe imports video, transcript,

and other log data, and uses linked and interactive representations for the user to navigate and explore, code, and export the data. We refined and enhanced visualizations and interaction modes of the framework by conducting formative studies. We then performed a summative evaluation of VizScribe, through which we showed the advantages of (a) filtered navigation, in helping identify context-specific patterns in the multimodal data visualizations, (b) multiple ways for analysts to approach the same dataset, and (c) supporting the identification of verbal and non-verbal relations within datasets. Finally, we found that the transcript, timeline, and sketch views were versatile visualizations, with custom views bringing newer ways of navigating the dataset and obtaining newer insights.

# 7. CLOSING NOTES

## 7.1 Conclusions

Computer support for the informal and ostensibly unstructured nature of early design is fraught with complexity. This is partly due to the nature of early, conceptual design where designers use sketches and verbalizations in the form of notes, annotations, and discussions to externalize their ideas. Traditional computer support tools for detailed design, for instance CAD applications, require the designer to follow prescribed, highly codified steps to externalize their ideas. This imposition of a prescriptive process does not hamper the designer significantly when the details of the problem and of the design are clear. However, extending such tools and operations to early design with a view to replacing the simple pencil-and-paper medium would prove intrusive to the designer. At the same time, this informal workflow—where designers tend to use their tacit knowledge and insights to form connections and explore ideas—are problematic to the design researcher, who has to make sense of the design process and of the designers' thinking by using video recordings and by examining the artifacts created and overwritten over time.

The goal of this thesis was to present tools and methods to help both designers and design researchers make sense of the unstructured verbal and text data generated in early design. To recapitulate, this thesis has three main contributions:

### 7.1.1 Text Visualization to Understand Collaborative Ideation

In order to obtain an overview and detailed view of the unstructured verbal data generated in collaborative ideation, we studied two groups of 9 design teams each where each group followed a different process of ideation. Using concept maps to identify dominant and connected keywords from the text, we obtained an overview

that showed us how the group of teams that followed an unstructured brainstorming processes stayed at a lower level of detail while achieving a wider breadth of design space exploration. The same overview for the group of teams that followed an alternating divergent-convergent process of ideation showed a higher level of detail, but explored a narrower breadth of ideas. The lexical dispersion plots revealed (when applied to transcripts of team discussions) *when* these keywords occurred during the brainstorming process, and revealed (when applied to team reports) *where* these keywords occurred. The text concordance plots aided disambiguation by revealing context. We posited that a combination of these visualizations would provide a powerful means for the design researcher to sift through such unstructured verbal data.

### 7.1.2   Cognitive Models to Understand Collaborative Sketching

We turned to the other form of idea externalization ubiquitous among designers: sketches. We identified requirements for a digital tool to aid collaborative sketching, and used skWiki, a web-based collaborative sketching framework, to perform transverse and longitudinal studies of collaborative conceptual design. We identified three main uses of skWiki's "paths" model:

- **Group awareness:** The representation of each team member's activity as a "path" of sketches provided the other members with a representation of each other's activity, allowing them to use skWiki effectively in both collocated and distributed settings. This representation was also useful to the design researcher in observing the team's activities without having to intrusively observe them.

- **Cognitive aspects of branching:** skWiki allowed designers to duplicate each other's ideas and modify them without the fear of overwriting the original ideas. This allowed them to create design alternatives that were related to, but distinct from the original ideas (*lateral transformations* [17]). The team was also able to

add detail to each other's ideas (*vertical transformations* [17]). The persistence and replicability of sketches eased both of these cognitive processes.

- **Cognitive aspects of merging:** While skWiki allowed for merging sketches, the operations required to perform such merges were prescriptive and therefore intrusive, resulting in no instances of merge being used in the earlier study. We thus focused on the specific problem of merging by using an application that allowed a designer to merge two given sketches, thus simulating merging in a collaborative design scenario. In addition to the traditional freeform selection of sketch elements, we explored an alternate selection technique that used stroke aggregation based on temporal occurrences of strokes. We identified four cognitive operations involved in merging, namely *reuse, refactor, repurpose, and reinterpret*, that designers used to explore the design space through merging and recomposing sketches. We found the difference between the emergent and source sketches to be positively correlated to the number of reinterpret operations, and negatively correlated to the number of reuse operations.

### 7.1.3  Understanding Design Team Behavior using Visual Analytics

We adapted the text visualization techniques used earlier, along with the "path" representation as used in skWiki, together with traditional protocol analysis data such as video and transcript, and presented a Visual Analytics-based framework called VizScribe. VizScribe allows the design researcher to create linked views of videos, transcripts, sketches, and other data that have a temporal aspect. Through the use of an *orient-focus-code* interaction paradigm, VizScribe allows the researcher to explore the data, identify points of interest, and interactively filter and query the dataset, using the linked views to provide context. It also allows the researcher to code segments of the transcript using an open, inductive coding technique. Studies conducted with design researchers and protocol analysts indicated that the linked views provided a variety of ways in which the users could access the data, with the

text views providing a combination of overview and detail, further augmented by the sketch "path" views and other sensor data visualizations. We also illustrated the flexibility of VizScribe in adapting to multimodal datasets. VizScribe's web-based framework was also used to prototype a "collaborative coding" scenario.

In summary, this thesis illustrates how the use of appropriately selected data representation techniques can help both the designer and design researcher in their tasks. In addition, we illustrated how computer support for collaborative sketching can go beyond merely emulating paper, by identifying existing cognitive operations for design space exploration that are made more convenient through such computer support tools, and by proposing newer cognitive operations for the same that are highly difficult, if not impossible without such computer support tools.

## 7.2   Future Work

The overarching goal of this work is to provide tools and methods to better understand collaborative conceptual design. Verbal and sketch data, as mentioned earlier, are steeped in ambiguity. The representations proposed and used in this work dispel some of this ambiguity by representing explicit connections between data, i.e. connections such as temporally co-occurring events in VizScribe, the locations and counts of keywords of interest in the text visualizations used, and paths connecting sketches based on editing histories. The next step to aid sense-making of such data would be to explore more implicit connections. These include, but are not restricted to, semantic connections between words, representing hierarchical relationships between concepts, and establishing relationships between sketches and verbal data. These and other directions for future research are outlined below:

- **Semantic relationships between words:** Verbal communication is inherently imprecise due to synonymy (the use of different words to refer to the same thing) and polysemy (the same word having different meanings). While text concordances provide a means to manually identify instances of polysemy,

synonymy is an issue currently not addressed by the text representations used in VizScribe. Similarly, the use of anaphors—using an expression whose interpretation depends on the context in which another expression is used—means that certain information is lost when using the text visualization techniques proposed in this thesis. An example would be the use of pronouns to refer to a previously defined noun, or designers pointing to a component in a sketch and refer to it as "that" or "this thing". The latter reference is lost unless all temporal sketch data are captured along with transcribed text. While some issues like anaphora resolution and text disambiguation remain open problems, lexical databases such as WordNet [136] provide hierarchical semantic relationships between nouns (hypernymy, or is-kind-of relationship) and between verbs (troponymy, or is-a-way-to relationship), in addition to synonym sets at each hierarchical level. Using existing NLP algorithms, a network between all verbal data used can be computed based on semantic distance measures [137]. Such measures can also be used to provide a hierarchy of concepts based on hypernymy to view the concepts verbalized at different levels of abstraction.

- **Domain-aware representations of verbal data:** Most lexical databases use general text corpora for calculating information content measures, and the relationships mentioned earlier. When it comes to design, domain-specificity is lost when using such general text corpora. Semantic relationships based on, say, troponymy can be made more design-specific by using function taxonomies [138]. Similarly, text-entailment recognition techniques [139] can be used to make causal relationships such as "if we choose this material, we will exceed the weight limit" which in turn can be used to infer design constraints. Using parts-of-speech (POS) tagging to distinguish functional attributes from structural attributes, named-entity recognition (NER) that will help identify domain-specific terms, and coding and identifying and mapping design decisions to form Linkographs [140] are other directions for future work.

- **Implicit relations between sketches:** In collaborative design sessions, designers can form mental connections between their sketches and those around them, which inspire them to sketch new ideas. These connections that are tacitly formed by designers are a challenge to record, both in traditional protocol analyses, and through systems such as skWiki. Identifying non-intrusive ways to capture and record these connections, perhaps through eye tracking sensors or gesture recognition techniques, would be a significant challenge.

- **Shared sketching spaces:** Early design often takes place on large whiteboards where designers often sketch together, exchanging ideas fluidly by sketching in a shared space. While devising the necessary hardware and software for such activities is out of the scope of this work, an interesting exercise would be to conceptualize how (and if) the skWiki paths model scales to such an interaction paradigm. skWiki currently emulates the notion of a shared design notebook: each designer operates in their own digital space, insulated from the others. The paths model for a "whiteboard" version of skWiki, and a visual representation of such a model pose interesting challenges, with promising implications.

- **Collaborative coding paradigms:** While a collaborative coding prototype was created to illustrate the strength of the web-based framework, creating a real-time, meaningful representation of collaboration is an altogether different problem. For instance, is it enough to show the coding activity between two researchers? Showing all activities of exploration, querying, and filtering would create an information overload and subsequent distraction for both participants. On the other hand, there are scenarios where a novice can study how an expert researcher/coder orients themselves to the data to explore and code them. At a methodological level, when is it useful to collaboratively code the data? Inter-coder reliability requires a degree of independence between the coders. On the other hand, some discussion about coding techniques are necessary for the sake of methodological consistency. Meaningful representations of coding

activity between researchers or coders with similar domain expertise may need to be different from the representations used between researchers with different domain expertise. Developing VizScribe in this direction will need comparative analyses with different representations of collaboration.

Jon Kolko [7] describes how the conceptual design process may appear to an observer: "The artifacts developed by the designer are messy, usually drawn in the midst of deep and reflective thinking; they are sketches drawn in Sharpie, incomplete sentences, and crude diagrams lacking adequate captions or descriptions. . . it is not immediately clear how one derived (the design idea) from (the research data)". While the design researcher will certainly continue to find the design process complex, it is safe to say that with appropriate computational support, there will gradually be fewer complications in analyzing such processes.

LIST OF REFERENCES

LIST OF REFERENCES

[1] M. W. Glier, S. R. Schmidt, J. S. Linsey, and D. A. McAdams. Distributed ideation: Idea generation in distributed capstone engineering design teams. *International Journal of Engineering Education*, 27(6):1281, 2011.

[2] C. J. Walthall, S. Devanathan, L. G. Kisselburgh, K. Ramani, E. D. Hirleman, and M. C. Yang. Evaluating Wikis as a communicative medium for collaboration within colocated and distributed engineering design teams. *Journal of Mechanical Design*, 133(7):071001, 2011.

[3] G. Pahl and W. Beitz. *Engineering Design: A Systematic Approach*. Springer, 1996.

[4] S. Pugh. Concept selection: a method that works. In *Proceedings of the International Conference on Engineering Design*, pages 497–506, 1981.

[5] K. T. Ulrich. *The Mechanical Design Process*. University of Pennsylvania, first edition, 2011.

[6] H. Rittel and M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.

[7] J. Kolko. Abductive thinking and sensemaking: The drivers of design synthesis. *Design Issues*, 26(1):15–28, 2010.

[8] R. Owen and I. Horváth. Towards product-related knowledge asset warehousing in enterprises. In *Proceedings of the international symposium on Tools and Methods of Competitive Engineering*, pages 155–170, 2002.

[9] W. Buxton. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2007.

[10] D. G. Ullman, S. Wood, and D. Craig. The importance of drawing in the mechanical design process. *Computers & Graphics*, 14(2):263–274, 1990.

[11] K. Henderson. *On line and on paper: Visual representations, visual culture, and computer graphics in design engineering*. MIT press, 1999.

[12] B. Jonson. Sketching now. *International Journal of Art & Design Education*, 21(3):246–253, 2002.

[13] T. Dorta. Implementing and assessing the hybrid ideation space: a cognitive artefact for conceptual design. *International Journal of Design Sciences and Technology*, 14(2):119–133, 2007.

[14] K. A. Ericsson. Protocol analysis and expert thought: Concurrent verbalizations of thinking during experts performance on representative tasks. In K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, editors, *The Cambridge handbook of expertise and expert performance*.

[15] M. Dinar, J. J. Shah, J. Cagan, L. Leifer, J. Linsey, S. M. Smith, and N. V. Hernandez. Empirical studies of designer thinking: Past, present, and future. *Journal of Mechanical Design*, 137(2):021101, 2015.

[16] D. O. Olguín, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland. Sensible organizations: Technology and methodology for automatically measuring organizational behavior. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(1):43–55, 2009.

[17] V. Goel. *Sketches of thought.* The MIT Press, 1995.

[18] L. L. Bucciarelli. *Designing Engineers.* MIT Press, Cambridge, MA, 1996.

[19] A. H. Vera, T. Kvan, R. L. West, and S. Lai. Expertise, collaboration and bandwidth. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 503–510, 1998.

[20] M. Michalko. *Thinkertoys: A Handbook of Creative-Thinking Techniques.* Ten Speed Press, 2010.

[21] A. F. Osborn. *Applied imagination.* Charles Scribner and Sons, New York, 1953.

[22] B. Rohrbach. Creative nach regeln: Methode 635, eine neue technik zum lösen von problemen. *Absatzwirtschaft*, 12(19):73–75, 1969.

[23] A. B. VanGundy. Brain writing for new product ideas: an alternative to brainstorming. *Journal of Consumer Marketing*, 1(2):67–74, 1984.

[24] A. Purcell and J. S. Gero. Drawings and the design process: A review of protocol studies in design and other disciplines and related research in cognitive psychology. *Design studies*, 19(4):389–430, 1998.

[25] G. Pahl and W. Beitz. *Engineering design: a systematic approach.* Springer, 2 edition, 1999.

[26] R. Van Der Lugt. Brainsketching and how it differs from brainstorming. *Creativity and Innovation Management*, 11(1):43–54, 2002.

[27] J. J. Shah, N. Vargas-Hernandez, J. D. Summers, and S. Kulkarni. Collaborative sketching (C-Sketch)–an idea generation technique for engineering design. *Creative Behavior*, 35(3):168–198, 2001.

[28] T. Salvador, J. Scholtz, and J. Larson. The Denver model for groupware design. *SIGCHI Bulletin*, 28(1):52–58, 1996.

[29] S. Slimani, C. Ferreira, L. Médini, and P. Ghodous. Conflicts mitigation in collaborative design. *International Journal of Production Research*, 44(9):1681–1702, 2006.

[30] B. M. Blair and K. Hölttä-Otto. Comparing the contribution of the group to the initial idea in progressive idea generation. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers in Engineering Conference*, 2012.

[31] R. H. McKim. *Experiences in Visual Thinking.* Brooks/Cole Pub. Co, Monterey, CA, 1972.

[32] F. L. McKoy, N. Vargas-Hernández, J. D. Summers, and J. J. Shah. Influence of design representation on effectiveness of idea generation. In *Proceedings of the ASME Design Engineering Technical Conference*, pages 39–48, 2001.

[33] M. Prats, S. Lim, I. Jowers, S. W. Garner, and S. Chase. Transforming shape in design: observations from studies of sketching. *Design Studies*, 30(5):503–520, 2009.

[34] L. B. Kara and K. Shimada. Supporting early styling design of automobiles using sketch-based 3D shape construction. *Computer-Aided Design & Applications*, 5(6):867—876, 2008.

[35] J. Dorsey, S. Xu, G. Smedresman, H. Rushmeier, and L. McMillan. The mental canvas: A tool for conceptual architectural design and analysis. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, pages 201–210, 2007.

[36] Q. Chen, J. Grundy, and J. Hosking. An e-whiteboard application to support early design-stage sketching of UML diagrams. In *Proceedings of the IEEE Symposium on Human Centric Computing Languages and Environments*, pages 219–226, 2003.

[37] M. C. Yang. Observations on concept generation and sketching in engineering design. *Research in Engineering Design*, 20(1):1–11, 2009.

[38] S. K. Badam, S. Chandrasegaran, N. Elmqvist, and K. Ramani. Tracing and sketching performance using blunt-tipped styli on direct-touch tablets. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 193–200, 2014.

[39] S. Zabramski and S. Neelakannan. Paper equals screen: a comparison of a pen-based figural creativity test in computerized and paper form. In *Procedings of the Conference on Creativity and Innovation in Design*, pages 47–50, 2011.

[40] C. G. van Dijk. New insights in computer-aided conceptual design. *Design Studies*, 16(1):62–80, 1995.

[41] H. Ishii and M. Kobayashi. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 525–532, 1992.

[42] S. L. Minneman and S. A. Bly. Managing a trois: A study of a multi-user drawing tool in distributed design work. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 217–224, 1991.

[43] J. C. Tang and S. L. Minneman. VideoDraw: a video interface for collaborative drawing. *ACM Transactions on Information Systems*, 9(2):170–184, 1991.

[44] L. Wang, W. Shen, H. Xie, J. Neelamkavil, and A. Pardasani. Collaborative conceptual design—state of the art and future trends. *Computer-Aided Design*, 34(13):981–996, 2002.

[45] A. Clayphan, A. Collins, C. Ackad, B. Kummerfeld, and J. Kay. Firestorm: A brainstorming application for collaborative group work at tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 162–171, 2011.

[46] N. Streitz, J. Geissler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz. i-LAND: An interactive landscape for creativity and innovation. In *Proc. ACM Conf. on Human Factors in Computing Systems*, pages 120–127, 1999.

[47] J. Hailpern, E. Hinterbichler, C. Leppert, D. Cook, and B. P. Bailey. TEAM STORM: demonstrating an interaction model for working with multiple ideas during creative group work. In *Proceedings of the ACM Conference on Creativity & Cognition*, pages 193–202, 2007.

[48] U. B. Sangiorgi, F. Beuvens, and J. Vanderdonckt. User interface design by collaborative sketching. In *Proceedings of the ACM Conference on Designing Interactive Systems*, pages 378–387, 2012.

[49] F. Geyer, J. Budzinski, and H. Reiterer. IdeaVis: a hybrid workspace and interactive visualization for paper-based collaborative sketching sessions. In *Proceedings of the Nordic Conference on Human-Computer Interaction*, pages 331–340, 2012.

[50] S. Finger, D. Gelman, A. Fay, M. Szczerban, A. Smailagic, and D. P. Siewiorek. Supporting collaborative learning in engineering design. *Expert Systems with Applications*, 31(4):734–741, 2006.

[51] C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1):39–58, 1991.

[52] J. Munson and P. Dewan. A concurrency control framework for collaborative systems. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 278–287, 1996.

[53] T. Grossman, J. Matejka, and G. Fitzmaurice. Chronicle: capture, exploration, and playback of document workflow histories. In *Proc. ACM User Interface Software & Technology*, pages 143–152, 2010.

[54] J. D. Denning, W. B. Kerr, and F. Pellacini. MeshFlow: interactive visualization of mesh construction sequences. *ACM Transactions on Graphics*, 30(4):66:1–66:8, 2011.

[55] M. J. Rochkind. The source code control system. *IEEE Trans. in Software Engineering*, 1(4):364–370, 1975.

[56] F. Chevalier, P. Dragicevic, A. Bezerianos, and J.-D. Fekete. Using text animated transitions to support navigation in document histories. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 683–692, 2010.

[57] J. Stempfle and P. Badke-Schaub. Thinking in design teams - an analysis of team communication. *Design Studies*, 23(5):473 – 496, 2002.

[58] K. H. Michael D. Mumford and I. Robledo. Methods in creativity research: Multiple approaches, multiple levels. In M. Mumford, editor, *Handbook of organizational creativity*. 2011.

[59] J. Gero, H. Jiang, and C. Williams. Does using different concept generation techniques change the design cognition of design students? In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers in Engineering Conference*, 2012.

[60] Y. Jin and P. Chusilp. Study of mental iteration in different design situations. *Design Studies*, 27(1):25 – 55, 2006.

[61] D. A. Waterman and A. Newell. PAS-II: an interactive task-free version of an automatic protocol analysis system. In *Proceedings of the ACM International Joint Conference on Artificial intelligence*, pages 431–445, 1973.

[62] J. Diederich, I. Ruhmann, and M. May. KRITON: a knowledge-acquisition tool for expert systems.

[63] G. Huber and C. Garca. Computer assistance for testing hypotheses about qualitative data: The software package AQUAD 3.0. *Qualitative Sociology*, 14(4):325–347, 1991.

[64] P. M. Sanderson, J. M. James, and K. S. Seidler. SHAPA: an interactive software environment for protocol analysis. *Ergonomics*, 32(11):1271–1302, 1989.

[65] C.-J. Lu and S. W. Shulman. Rigor and flexibility in computer-based qualitative research: Introducing the coding analysis toolkit. *International Journal of Multiple Research Approaches*, 2(1):105–117, 2008.

[66] M. Pourmohamadi and J. S. Gero. LINKOgrapher: An analysis tool to study design protocols based on FBS coding scheme. In *Proceedings of the International Conference on Engineering Design*, 2011.

[67] P. A. Duff and J. Séror. Computers and qualitative data analysis: Paper, pens, and highlighters vs. screen, mouse, and keyboard. *TESOL Quarterly*, 39(2):321–328, 2005.

[68] C. Ros, Y.-C. Wang, Y. Cui, J. Arguello, K. Stegmann, A. Weinberger, and F. Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 3(3):237–271, 2008.

[69] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 221–227, 1996.

[70] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proceedings of the ACM conference on Human factors in computing systems*, pages 457–466, 2008.

[71] M. Monroe, R. Lan, J. Morales del Olmo, B. Shneiderman, C. Plaisant, and J. Millstein. The challenges of specifying intervals and absences in temporal queries: a graphical language approach. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2349–2358, 2013.

[72] J. Olsson and M. Boldt. Computer forensic timeline visualization tool. *Digital Investigation*, 6, Supplement(0):S78–S87, 2009.

[73] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. PortVis: a tool for port-based detection of security events. In *Proceedings of the ACM workshop on Visualization and data mining for computer security*, pages 73–81, 2004.

[74] C. Stab, K. Nazemi, and D. W. Fellner. SemaTime – timeline visualization of time-dependent relations and semantics. In G. Bebis et al., editor, *Advances in Visual Computing*, volume 6455 of *Lecture Notes in Computer Science*, pages 514–523. Springer Berlin Heidelberg, 2010.

[75] P. André, M. L. Wilson, A. Russell, D. A. Smith, A. Owens, et al. Continuum: designing timelines for hierarchies, relationships and scale. In *Proceedings of the ACM symposium on User interface software and technology*, pages 101–110, 2007.

[76] S. Rubin, F. Berthouzoz, G. J. Mysore, W. Li, and M. Agrawala. Content-based tools for editing audio stories. In *Proceedings of the ACM symposium on User interface software and technology*, pages 113–122, 2013.

[77] P. Brundell, P. Tennent, C. Greenhalgh, D. Knight, A. Crabtree, C. O'Malley, S. Ainsworth, D. Clarke, R. Carter, and S. Adolphs. Digital replay system (DRS)– a tool for interaction analysis. In *Proceedings of the International Conference on Learning Sciences (Workshop on Interaction Analysis)*, 2008.

[78] A. Fouse, N. Weibel, E. Hutchins, and J. D. Hollan. Chronoviz: a system for supporting navigation of time-coded data. In *ACM Extended Abstracts on Human Factors in Computing Systems*, pages 299–304, 2011.

[79] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[80] M. Hoey, M. Scott, and G. Thompson. *Patterns of Text: in Honour of Michael Hoey.* John Benjamins Publishing Company, 2001.

[81] M. Wattenberg and F. B. Viegas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008.

[82] C. Collins, F. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 91–98, 2009.

[83] M. Wattenberg. Arc diagrams: Visualizing structure in strings. In *IEEE Symposium on Information Visualization*, pages 110–116, 2002.

[84] C. Collins, S. Carpendale, and G. Penn. DocuBurst: Visualizing document content using language structure. *Computer Graphics Forum*, 28(3):1039–1046, 2009.

[85] S. Pugh. *Total design: integrated methods for successful product engineering.* Addison-Wesley Publishers Ltd., 1991.

[86] T. Brown. Design thinking. *Harvard Business Review*, 86(6):84, 2008.

[87] P. Laseau. *Graphic thinking for architects and designers.* Van Nostrand Reinhold (New York), 1980.

[88] M. Michalko. *Thinkertoys: a handbook of creative-thinking techniques.* Ten Speed Press, Berkeley, CA, 2006.

[89] E. Taborda, S. Chandrasegaran, L. Kisselburgh, T. Reid, and K. Ramani. Enhancing visual thinking in a toy design course using freehand sketching. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers in Engineering Conference*, 2012.

[90] B. M. Kudrowitz and D. R. Wallace. The play pyramid : A play classification and ideation tool for toy design. *International Journal of Arts and Technology*, 3(1):36–56, 2010.

[91] A. Smith and M. Humphreys. Evaluation of unsupervised semantic mapping of natural language with leximancer concept mapping. *Behavior Research Methods*, 38(2):262–279, 2006.

[92] M. Chalmers and P. Chitson. Bead: Explorations in information visualization. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval*, pages 330–337, 1992.

[93] Leximancer. *Leximancer Manual, Version 4.* Leximancer Pty Ltd., 2011. See also URL `http://www.leximancer.com`.

[94] NLTK Project. Natural Language ToolKit, 2012. URL `http://www.nltk.org`.

[95] P. André, M. Schraefel, J. Teevan, and S. T. Dumais. Discovery is never by chance: designing for (un)serendipity. In *Proceedings of the ACM conference on Creativity and cognition*, pages 305–314, 2009.

[96] J. Hong, G. Toye, and L. J. Leifer. Personal electronic notebook with sharing. In *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 88–94, 1995.

[97] F. Lakin, J. Wambaugh, L. Leifer, D. Cannon, and C. Sivard. The electronic design notebook: Performing medium and processing medium. *The visual computer*, 5(4):214–226, 1989.

[98] Z. Zhao, S. K. Badam, S. Chandrasegaran, D. G. Park, N. Elmqvist, L. Kisselburgh, and K. Ramani. skWiki: A multimedia sketching system for collaborative creativity. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1235–1244, 2014.

[99] D. G. Jansson and S. M. Smith. Design fixation. *Design Studies*, 12(1):3–11, 1991.

[100] E. S. Ferguson. *Engineering and the Mind's Eye.* The MIT Press, 1992.

[101] H. Zhu. From WYSIWIS to WISINWIS: role-based collaboration. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 6, pages 5441–5446, 2004.

[102] K. Schmidt. The problem with 'awareness': Introductory remarks on 'awareness in CSCW'. *Computer Supported Cooperative Work*, 11(3-4):285–298, 2002.

[103] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proc. ACM Conference on the World Wide Web*, pages 211–220, 2007.

[104] C. Gutwin and S. Greenberg. Design for individuals, design for groups: Trade-offs between power and workspace awareness. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 207–216, 1998.

[105] D. Zahner, J. V. Nickerson, B. Tversky, J. E. Corter, and J. Ma. A fix for fixation? rerepresenting and abstracting as creative processes in the design of information systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 24(02):231–244, 2010.

[106] G. Goldschmidt. The dialectics of sketching. *Creativity Research Journal*, 4(2):123–143, 1991.

[107] D. A. Schon and G. Wiggins. Kinds of seeing and their functions in designing. *Design Studies*, 13(2):135–156, 1992.

[108] M. Suwa, B. Tversky, J. Gero, and T. Purcell. Seeing into sketches: Regrouping parts encourages new interpretations. In *Visual and Spatial Reasoning in Design*, pages 207–219. 2001.

[109] Z. Zhao, W. Benjamin, N. Elmqvist, and K. Ramani. Sketcholution: Interaction histories for sketching. *International Journal of Human-Computer Studies*, 82:11–20, 2015.

[110] D. G. Ullman. *The Mechanical Design Process*, volume 4. McGraw-Hill, New York, 2009.

[111] M. Waldron and K. Waldron. Conceptual CAD tools for mechanical designers. In *Proceedings of Computers in Engineering Conference*, volume 1, pages 203–209, 1988.

[112] A. Oak. What can talk tell us about design?: Analyzing conversation to understand practice. *Design Studies*, 32(3):211–234, 2011.

[113] K. Henderson. The role of material objects in the design process: A comparison of two design cultures and how they contend with automation. *Science, Technology & Human Values*, 23(2):139–174, 1998.

[114] A. Dong. The latent semantic approach to studying design team communication. *Design Studies*, 26(5):445–461, 2005.

[115] J. W. Kan, J. S. Gero, and H.-H. Tang. Measuring cognitive design activity changes during an industry team brainstorming session. In *Design Computing and Cognition*, pages 621–640. Springer, 2011.

[116] J. J. Thomas and K. A. Cook. Illuminating the path: The research and development agenda for visual analytics. Technical report, 2005.

[117] C. Ware. *Information visualization: perception for design.* Elsevier, 2012.

[118] A. Newell. On the analysis of human problem solving protocols. In *International Symposium in Mathematical Methods in the Social Science*, 1966.

[119] T. Kim, E. McFee, D. O. Olguin, B. Waber, and A. Pentland. Sociometric badges: Using sensor technology to capture new forms of collaboration. *Journal of Organizational Behavior*, 33(3):412–427, 2012.

[120] J. Heer and B. Shneiderman. A taxonomy of tools that support the fluent and flexible use of visualizations. *ACM Queue*, 10(2):1–26, 2012.

[121] M. Carpendale. Considering visual variables as a basis for information visualization. Technical Report 2001-693-16,, Department of Computer science, University of Calgary, Calgary, AB, Canada, 2003.

[122] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.

[123] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10:327–340, 2011.

[124] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1217–1226, 2008.

[125] M. Harrower and C. A. Brewer. Colorbrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[126] A. C. Robinson. Collaborative synthesis of visual analytic results. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 67–74, 2008.

[127] J. S. Yi, Y. a. Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[128] C. Müller-Tomfelde. Dwell-based pointing in applications of human computer interaction. In C. Baranauskas, P. Palanque, J. Abascal, and S. Barbosa, editors, *Human-Computer Interaction*, volume 4662 of *Lecture Notes in Computer Science*, pages 560–573. 2007.

[129] M. Bostock, V. Ogievetsky, and J. Heer. D$^3$ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[130] B. G. Glaser and A. L. Strauss. *The discovery of grounded theory: Strategies for qualitative research.* Transaction Publishers, 2009.

[131] N. Elmqvist and J. S. Yi. Patterns for visualization evaluation. *Information Visualization*, 14(3):250–269, 2015.

[132] J. Brooke. SUS-a quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194, 1996.

[133] J. W. Creswell. *Qualitative inquiry and research design: Choosing among five approaches*. Sage, 2012.

[134] J. W. Drisko. Qualitative data analysis software: An overview and new possibilities. In A. E. Fortune, W. J. Reid, and R. L. M. Jr., editors, *Qualitative Research in Social Work*, pages 284–303. 2 edition, 2013.

[135] J. Yuan and M. Liberman. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878, 2008.

[136] G. A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[137] A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*, volume 2, pages 2–2, 2001.

[138] J. M. Hirtz, R. B. Stone, S. Szykman, D. McAdams, and K. L. Wood. Evolving a functional basis for engineering design. In *Proceedings of the ASME Design Engineering Technical Conference*, 2001.

[139] J. Herrera, A. Penas, and F. Verdejo. Textual entailment recognition based on dependency analysis and WordNet. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 231–239. Springer, 2006.

[140] G. Goldschmidt. *Linkography: unfolding the design process*. MIT Press, 2014.

VITA

VITA

Senthil Chandrasegaran obtained his bachelor's degree in mechanical engineering in from the Regional Engineering College (now National Institute of Technology), Trichy, India, in 2000. He then worked as a design engineer in Tata AutoComp Systems in Pune, India, developing computer-aided design (CAD) models for automotive interior trims and automotive lighting systems. In 2004, he joined Alcyon Engineering, an engineering consultancy specializing in structural analysis and knowledge-based engineering. He worked in both areas until 2009 before he started graduate school in the direct PhD program in the School of Mechanical Engineering at Purdue. His research interests include design theory and methodology, information visualization, and specifically the integration of computer support tools to understand and support conceptual design.