ABSTRACT

Title of Dissertation: ZOOMABLE USER INTERFACES FOR THE AUTHORING AND DELIVERY OF SLIDE PRESENTATIONS

Lance Everett Good, Doctor of Philosophy, 2003

Dissertation directed by: Professor Benjamin B. Bederson
Department of Computer Science

Millions of slide presentations are being authored and delivered with computer software every day. Yet much of the computer's power for these tasks remains untapped. Existing interaction techniques leave presenters wrestling with limited size computer displays to get meaningful overviews of their work. Without these overviews, they have trouble finding patterns in their data and experimenting with alternate organizations. They also have difficulty communicating the structure of large or complex talks to the audience and keeping the audience oriented during unexpected transitions between ideas. A natural solution is Zoomable User Interfaces (ZUIs) since they offer the capability to view information at multiple levels of detail and smoothly transition between ideas. This work presents two ZUIs, Niagara and CounterPoint, for authoring and delivering slide presentations.

Niagara is a ZUI workspace for authoring presentation content with techniques to improve authoring in the zoomable environment. Empirical evaluations of ZUI-based authoring tools revealed performance improvements and subjective preferences over folder-based interfaces for organization tasks. Users were 30% faster with ZUIs than with folders in completing a simplified shape organization task. Some classes of users were also faster with ZUIs than with folders in completing a text-based organization task. Users performing both tasks exhibited a strong preference for ZUIs over folders.

CounterPoint provides a number of features to simplify the creation and delivery of ZUI presentations. The effects of these presentations on the audience were evaluated in a controlled comparison of presentations with slides only, slides with spatial layouts, and slides with spatial layouts and animation. The study revealed a strong subjective preference and higher ratings of organization for presentations with spatial layout.

Feedback was also gathered from presenters who used CounterPoint to deliver over 100 real-world presentations. They indicated that CounterPoint helped them communicate overviews and multi-level presentation structures. More experienced CounterPoint presenters also found that CounterPoint helped them keep the audience oriented when navigating the presentation in response to audience feedback.

ZOOMABLE USER INTERFACES FOR THE AUTHORING AND DELIVERY OF SLIDE PRESENTATIONS

by

Lance Everett Good

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2003

Advisory Committee:
- Professor Benjamin B. Bederson, Chair
- Professor Doug Oard
- Professor Adam Porter
- Professor Ben Shneiderman
- Dr. Mark J. Stefik

DEDICATION

To Cristie

ACKNOWLEDGEMENTS

I had the opportunity to work with two great advisors. Ben Bederson was my academic advisor and always gave me extremely helpful advice and direction. Ben's creativity and pragmatism inspired and encouraged me. I also appreciated Ben's intellect and hope I can take away some of his great reasoning skills. My unofficial second advisor was Mark Stefik. I benefited immensely from the experiences and wisdom that Mark shared with me. Mark always challenged me with his highly scientific approach to problems and motivated me with his excitement for new ideas. I also have to thank Mark for going out of his way to allow me to work at PARC.

Thanks to Doug Oard, Adam Porter, and Ben Shneiderman for agreeing to be on my committee. They also gave helpful corrections and advice that improved this dissertation. Thanks also to Evan Golub and Bill Killam for letting me use their HCI classes in my audience user study.

I would also like to thank the members of the labs I worked in. I really enjoyed the cooperative spirit in the Human-Computer Interaction Lab and appreciated all the knowledge and kindness shared with me by the HCIL's students, staff, and faculty. Thanks especially to Juan Pablo Hourcade for his help throughout my time at Maryland. Thanks also to Allision Druin, Jesse Grosjean, Harry Hochheiser, Hilary Browne

Hutchison, Hyunmo Kang, Jaime Montemayor, Catherine Plaisant, Anne Rose, and Bongwon Suh. Thanks to Allison Farber for designing artwork for CounterPoint.

I have also appreciated my time in the Information Sciences and Technology Lab at PARC. I have enjoyed getting to know the members of ISTL and have benefited from the diverse academic backgrounds in the lab. I especially appreciated the willingness of lab members to participate in my (often-tedious) user studies. A special thanks to the past and present members of my area, Patrick Baudisch, Eric Bier, Sascha Brawer, Nathan Good, Bill Janssen, Paula Newman, Ken Pier, Kris Popat, and Polle Zellweger, for their advice and feedback throughout my time at PARC. The members of UIR, Stu Card, Ed Chi, Julie Heiser, Lichan Hong, Jock Mackinlay, Peter Pirolli, and Christiaan Royer also gave me good advice on statistics, user studies, and how I should present my work. Of course, I also have to thank my office neighbors Jeanette Figueroa and Mali Sarpangal for their friendliness and administrative support.

Thanks to my friends Mike and Melissa Fuller, Joshua and Alicia Ackerson, and Lori Walker for their support and being willing to be guinea pigs in my studies. I am grateful for the support of all my other friends, especially Chris Brown, Trey Jackson, Jim Lin, Ted Hseih, Rich Hsu, Dave and Caroline Seinknicht, Jonathan Berent, Timothy Landhuis, Sandra Park, Hiro Tsuda, Margaret Venables, and Greg and Karin Jackson, who helped me keep my sanity during my time as a student.

I am also especially grateful to my family. Thanks to my parents, Ron and Lori Good, for instilling in me the value of education. My sister, Rhonda Good, was also a source of encouragement. I also send a shout out to my brother-in-law, Mike Kelly - maybe there is still hope for us Aspergers. Also my baby, Ainsley (the cutest...baby...ever), always had a big smile to cheer me up when I got home from work. And finally, I could never acknowledge everything my wife Cristie has done to make this dissertation possible. She supported me in countless ways and sacrificed herself in taking care of Ainsley in the final months of my writing. Cristie also suffered through a barrage of pilot studies and gave me many good ideas for improving the usability of my software.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1:
Introduction

Formal face-to-face communications tend to be accompanied by visual aids when ideas can be more efficiently conveyed through visuals than by spoken word. The most common type of visual aid employed in these situations is the slide show. Slide shows are used in almost every possible setting including the military, businesses, homes, religious institutions, and educational settings [40, 58]. One specific slide show presentation tool, PowerPoint, is reportedly being used to author or deliver over 30 million presentations per day [82]. These numbers are only likely to increase as the volume of human knowledge increases.

Before personal computers became ubiquitous, slide shows were typically delivered using 35mm slides or overhead transparencies. These media dictated that a slide show consist of a sequence of fixed images. Although the slides could be reordered, the decision about which slides to use and the order in which they would appear was established prior to giving a presentation. Conventions evolved to help keep the audience oriented during a slide show, such as placing "outline" slides in a long presentation. A division of labor also emerged for the preparation of slides. The presenter determined the content of the slides, but a graphic artist with specialized skills prepared the actual slides.

Computer tools, such as PowerPoint [89], have revolutionized slide show processes. These tools have made creating highly polished presentation slides possible for nearly all computer users. Computer-based presentation tools were originally designed as a substitute for the services provided by a graphic artist. As a result, much of the functionality provided by these tools is focused on graphic design. This includes controls for visual properties such as layout, font sizes, number of bullets per slide, colors, etc. As PCs, laptops, and video projectors have become more common these tools have subsequently been adapted from the graphic design task to the additional tasks of preparing presentation content and actually delivering the presentation.

Given that the entire presentation process, from conception to delivery, has moved to a virtual medium, the question arises as to whether computer-based presentations could be improved now that they are not limited by the constraints of physical slides. Physical media require that the presentation be divided into a linear sequence of static fixed-size images. In contrast, computer-based presentations are potentially free from all of these limitations offering non-linear, dynamic, variable-sized visualizations. How could these computer tools help a presenter create and organize their ideas? Could more powerful organizing structures be used than a linear sequence? What capabilities could help the speaker or audience understand the ideas better and stay oriented during a presentation? Could slide shows be made more reusable, adaptable in real-time to the requirements of an audience?

1.1 What's so hard about authoring a presentation?

Besides the work required for the graphic design of presentation slides, there is work to be done in preparing the content itself. In fact, authoring presentation content is cognitively demanding work. The presenter ultimately has to decide what information will be included in the presentation, how that information will be organized, and how to best turn that organization into a story for the target audience.

The processes involved in authoring texts, such as presentation content, have been the subject of much research in recent years. There is some evidence to suggest that these processes differ from person to person [14]. In fact, given the difficulty and cognitive complexity of the task, it would be surprising if this were not the case. Yet, several components of these authoring processes have been identified as common across many authors. Hunter and Begoray [55] analyzed several models of authoring text and identified four common components in these models including: generating, organizing, composing, and revising. The generating component is where ideas are collected and recorded. These ideas can come from the author's introspection or from external sources. Next, the organizing component involves making decisions about abstractions and ordering leading to hierarchical and linear structures. In the composing stage, the author takes the structure developed in the generating and organization stages and turns it into an actual usable product. For presentations, this typically means creating the actual presentation media, such as slides. Lastly, the revision stage involves reviewing the work, adding new ideas and fixing inconsistencies with the original organization. The model implies a loose linear ordering to the four activities. However, this ordering is not strict and is quite likely to be violated. Indeed, one of the models cited by Hunter and Begoray is quite adamant that these processes are not sequentially completed [49].

The research cited above is not specific to authoring presentation content. In fact, it is largely targeted at "writing" tasks that result in some form of printed publication. Yet even this type of writing is extremely diverse, including many different media, genres, and formats. As a result, a fundamental assumption in this previous research is that all text authoring tasks involve essentially the same processes. Perhaps the most compelling reason why the processes are the same for these diverse authoring tasks comes from Flower and Hayes who suggest three constraints faced by writers across writing tasks [32]. First, an author is constrained by knowledge because it must be converted from mental models and distributed sources to a single coherent whole. Second, written speech compels the author to obey various written language conventions that often are not followed in the author's internal dialog or external spoken language. Lastly, the rhetorical problem constrains the author to thinking about the intended audience and the conventions of the presentation format. Naturally, these three constraints have varying importance at different stages of different authoring tasks. In the specific case of authoring presentations, all three constraints must be satisfied, though the written speech constraint tends to have lower emphasis since presentation slides often contain sentence fragments and limited amounts of text. Nevertheless, because these same constraints arise for a wide range of authoring tasks, the processes needed to solve these tasks are also similar.

One of the common themes in the early stages of authoring text is the ability to try many alternatives. This is particularly evident in brainstorming or idea generation, where more ideas are often created than are actually used in the final product [33, 104]. Exploring alternatives is also important in the organization phase, since many organizations are typically considered before one is ultimately chosen. In fact, Neuwirth, et al. present evidence that the number of different perspectives considered in exploring a set of concepts influences the quality and creativity of the resulting document [76]. As a result, it is important for the author to be able to switch between many alternative organizations as the presentation becomes more complete. The more difficult it is to explore alternatives, the fewer alternatives the presenter is likely to consider.

The presenter’s ability to explore alternatives is largely influenced by the formality of an authoring tool. Unnecessary formality occurs when the representations and structures of the tool do not match the users' tasks [101]. Mismatches between the task and the tool increase the costs to the user in converting from their mental representations to the system’s representations. In addition, formal structures can introduce modification costs that reduce the chances that an author will explore alternative organizations. Some of the most well known examples of this type of formality mismatch were noted in the context of a textual authoring tool called Notecards where users sometimes had to interrupt their work to think of a name for a note or where to put a note [46]. A similar presentation-specific example occurs in many slide authoring tools when a presenter is entering ideas and a slide becomes full. At this point, the presenter is forced to make a decision to either reorganize the slide or move to a new slide to continue at the point where the fault occurred. In this case, the presenter's goal is simply to enter the information but the tool imposes a formal division between slides to which the presenter must adapt.

Beyond simply exploring alternatives, Flower and Hayes describe text authoring in terms of finding solutions to an "excessive number" of constraints [32]. Because this authoring process requires the author to make choices from such a large constraint space, it is cognitively demanding work. An author’s grasp on the large quantity of information and large number of constraints can be extremely delicate.

As an author works, many ideas about the presentation's content and structure pass through the author's short-term memory. If interruptions arise, these fleeting ideas can be temporarily lost or even permanently forgotten. These interruptions can take several forms. One of the most common interruptions in slide authoring software is similar to what Dertouzos calls the perfection fault [25]. The perfection fault occurs whenever an author becomes involved in manipulating aspects of the visual appearance of a document beyond what is necessary or beneficial to the authoring task. This type of interruption has been reported to plague slide authoring software to such an extent that slide authors often neglect their content in favor of visual appearance [39, 40, 113]. Focusing on graphic design in this way provides users with a false sense of preparedness. Because the presentation looks good, authors are often lulled into thinking the content is better than it is. Similarly, presenters tend to ignore the subtleties and complex relationships of difficult concepts, choosing instead to simplify ideas to one line phrases that conform to the bulleted lists provided by many slide authoring tools [82, 113]. Perhaps more cognitively harmful is that the consideration for the visual design of presentation slides can introduce additional constraints to an already cognitively demanding task. It also compounds the amount of effort needed to maintain a set of slides as the visual appearance must often be adjusted as the content grows and changes.

Another type of interruption can arise from the low-level interactions with a computer based authoring tool. Human factors researchers have used GOMS (Goals Operators, Methods, and Selection) models or other keystroke level models to characterize these types of interactions (for example, see the work of Card, et al. [16]). Some typical operations considered in low-level models of computer-based tools include pointing, clicking, keystrokes, hand movements, mental preparation, etc. These operations become distractions when they occur too frequently, take too much time to complete, or require too much attention or concentration. Here they can displace the author's thoughts about content with thoughts about how to accomplish certain sub-tasks with the tools provided. One example specific to authoring presentation content arises in the low-level interactions involved in comparing objects on a computer display. Comparisons are extremely common in brainstorming and organizing presentation content as authors look for patterns and consider alternative groupings and orderings. Presentation authors are frequently scanning their information for patterns and looking for the best way to structure and order that information. In a large workspace, like a whiteboard or a table, an author can spread the information out and do comparisons between objects with eye and head movements. On a computer screen this same task often requires the author to scroll or move between slides or folders in order to see the objects to be compared. The result is that the author often must trade fast eye movements for slower hand movements, making the task much more time consuming and thus cognitively disruptive.

This previous example hints at one of the overarching difficulties in authoring text on a computer display, namely insufficient screen space. Computer displays are typically 30 to 40 times smaller than ordinary tables or whiteboards [51]. Research suggests that in many situations the limited view of the computer display can make both writing and reading tasks more difficult than with traditional physical media because of a lack of good global awareness in the text [35, 72, 97]. Sharples presents a table comparing the properties of several media used for authoring textual content [98]. The table reflects that computers have many characteristics missing in other media but typically lack the facility for providing good overviews of textual content. This limitation is noted more generally by Henderson et al. [51]. They describe the similarities between the space management strategies employed on a computer display and in computer memory. Both situations can lead to thrashing, where more time is spent managing space than doing work if additional mechanisms are not introduced. Although this situation will be mitigated by the availability of larger computer displays, it is likely to remain an important issue for portable devices such as laptops and PDAs.

1.2 What's so hard about delivering presentations?

The art of presentation is to arrange ideas in a way that is understandable and holds the audience's interest. If there are only a few things to say, we can often just make a short speech rather than bothering with a slide show. Slide shows are most useful when there is too much to just "say", and when we want to reinforce the audience's understanding by combining visual images with spoken words. Yet even slide shows can provide inadequate support for many presentations.

One place where slide shows are often inadequate is in helping the presenter communicate the structure of a presentation to the audience. A slide show does not inherently provide any visual support to the presenter in explaining how the individual slides relate to one another nor how they fit into the larger structure of the presentation. Presenters have adapted to this limitation by using outline or overview slides. Outline slides are typically used to layout the structure of the presentation to help the presenter communicate relationship between topics. However, presenters tend to use these overview slides sparingly. This is due in part to the extra effort required to create these slides and keep them up to date as the presentation is reorganized. Presenters also tend to feel that including too many overviews can seem pedantic, as suggested by the extreme case in which an overview slide is shown after each individual content slide. Perhaps the most limiting aspect of outline slides is that a single outline slide often can't convey the structures of large presentations or presentations with multiple levels of detail. Multiple outline slides can be used but the audience is forced to mentally integrate the disconnected structures. More often than not in these situations, presenters rely on their verbal descriptions of the presentation structure or develop ad hoc visual designs.

Having an insufficient portrayal of the structure can be particularly disruptive when a presentation is very interactive, such as a business or military briefing. In these cases, the audience will often interrupt the presenter at inappropriate times to ask questions that would have been more appropriately asked at a previous point or that will be covered later in the presentation. Similarly, the presenter will often have more information than can be covered in the available time. Yet, the audience is often unaware of the additional “backup” information on a topic so they may hesitate to ask questions when more details are readily available.

Slide shows are also inadequate in supporting the presenter when unexpected things happen during a presentation. There are often last minute changes in the amount of time available for a presentation or in who will be in the audience. As mentioned, it is also common for the audience to ask questions about topics that were already covered, that will be covered later, or that are discussed in backup slides that are not included in the main presentation. When these situations occur, the presenter is likely to need to navigate between nonconsecutive slides in the presentation. During navigation in a traditional slide show, the transitions between slides typically convey no information about the underlying content. Even worse, these transitions are often filled with spurious graphical animations that can distract the audience from the presentation content. The result is that these traditional slide transitions do not help communicate the conceptual transition and the presenter is left to provide the context for the topic shift. For a well-scripted presentation, the presenter can anticipate the limitation and account for it by inserting explanatory transition dialog. However, when unexpected navigations occur during the presentation, the presenter must improvise a transition to describe how two slides relate or how the next slide fits into the larger presentation structure.

Improvising to meet the demands of changing time constraints or audience feedback puts a lot of stress on the presenter. The presenter is typically thinking of what to say while simultaneously trying to navigate and manage a slide show. This kind of multitasking conflicts with a fundamental limitation in human short-term memory that makes it difficult to speak and simultaneously attend to other cognitive tasks [106]. As a result, the cognitive overhead of slide show navigation should ideally be minimized to allow presenters to focus on what they are going to say. Yet in reality, navigating a traditional slide show is often unnecessarily demanding because the slides are agnostic of the structure and content of the presentation. To jump out of the scripted presentation order, a presenter must perform an unaided linear search to find the particular idea or slide of interest.

In addition, the interaction techniques used to search are often problematic. In the days of physical slides or transparencies, the linear search meant flipping through slides in a carousel or through transparencies in a pile. In software presentation tools, this search often means scrolling through linear lists of slide titles, grids of thumbnails, or the actual full screen slides. These scrolling interfaces typically make use of traditional GUI widgets that were designed primarily for use at an individual workstation (for example, see discussion in [75]). As a result, they can be inappropriate in the presentation setting as they require a high level of attention and pointing precision. They can also be visually distracting and unprofessional since they do not follow the same design constraints as the rest of the presentation.

One technique used to facilitate impromptu navigations in software slide shows is hyperlinking. Hyperlinks reduce the amount of attention and precision needed for navigation and eliminate the need for a linear search. Hyperlinks can also be designed to visually conform to the rest of the presentation. However, hyperlinks are frequently not created because presenters are typically unable and unwilling to anticipate all the points at which they may want to deviate from the primary storyline.

1.3 Finding Solutions Based on Zoomable User Interfaces

Most of the problems described above for both authoring and delivering presentations become more evident as the amount of information increases. The more information a presentation contains the more work there can be in organizing it. When there are a lot of ideas, there are more supporting visuals that the presenter needs to access during the talk. In both authoring and delivery, the presenter is concerned with distinguishing the main points from the details and finding the best story to communicate the important ideas to the audience.

One of the constants in presentations is that information is delivered in chunks. For example, there is only so much information a video projector can display at once. The combination of large amounts of information, a need for structure, and limited space is at the root of many of the difficulties in preparing and delivering information. It is also a good match for Zoomable User Interfaces since these interfaces directly address the problem of handling variable amounts of information.

This work proposes techniques based on Zoomable User Interfaces (ZUIs) as solutions to the problems described above. ZUIs are an alternative to traditional techniques for visualizing information. ZUIs display information on a conceptually infinite two-dimensional plane. They allow users to change their view of this plane through panning and zooming to access more information than can typically be displayed on a single screen. A fundamental characteristic of zooming and panning operations in ZUIs is that they are animated. These types of animations give a sense of physical movement by mimicking such physical acts as sliding a paper on a table (panning), looking at a paper more closely for detail (zooming in), or holding a paper at a distance for more context (zooming out). ZUIs have been used in such settings as visualizing histories [52], authoring and presenting children's stories [13, 30], traversing file system hierarchies [9], and image browsing [6].

1.3.1 Authoring Presentation Content

ZUIs are fundamentally based on 2D workspaces. 2D workspaces such as whiteboards, blackboards, and tabletops have long been recognized for their beneficial qualities for brainstorming and organizing ideas. Many of these same qualities have been demonstrated for organization tasks with 2D workspaces on computer displays. One important quality for an organization task is that these workspaces exploit human memory for spatial location. In a computer-based organization task, spatial organizations have been shown to improve both recall speed and recall error rates over non-spatial organizations [93].

2D workspaces also make no limitations on where you can move objects or how objects have to be aligned. This allows authors to indicate subtleties of information relationships such as positioning objects midway between two related groups of objects or positioning objects to indicate uncertainty in their order. The ability to portray subtle or ambiguous relationships frees the author from having to make premature decisions about how objects are structured. As a result, the interface has an informal quality that encourages authors to explore alternative relationships in their data to find the best organizations.

Because ZUIs extend the capabilities of 2D workspaces, they provide both the benefits of 2D plus additional benefits from viewing the workspace at multiple levels of detail. ZUIs offer natural parallels to the behavior of an author using a physical workspace. When working on physical surfaces, authors frequently alternate between focused work in one portion of the workspace and more global comparisons over the entire workspace. An author can accomplish these same tasks in ZUIs through zooming. To focus on the details, the author zooms in; to get more context, the author zooms out. These multilevel views can help alleviate some of the global awareness problems that have been reported in previous computer-based authoring tasks.

Since ZUIs allow for better overviews of the workspace, many of the interruptions caused by expensive low level operations, such as comparing and moving objects, are reduced. Specifically, because all objects can be brought into a single view, the author can quickly scan and compare objects in the workspace with fast eye movements. Spatial awareness then helps the author stay oriented in the workspace and remember specific object positions. Information can also be quickly compared at multiple levels. Individual objects can be compared, but groups of objects and groups of groups can also be easily compared. In general, more of the constraints needed for organizing a lot of data can be brought quickly to mind and analyzed because the constraints are all visible at once. Likewise, many object manipulations are also made easier with a global perspective. Perhaps the most common operation that is simplified is moving an object from one group to a specific location within another group.

Although ZUI workspaces have several inherent benefits for authoring presentation content, they have suffered from usability limitations that may have kept them from being employed extensively for these types of authoring tasks. Most importantly, ZUIs have not been well studied for use with abstract data. As a result, one of the biggest questions in using ZUIs for authoring presentation content is how ZUIs can provide meaningful zoomed-out views for primarily text-based content while preserving the value of spatial arrangement.

Likewise, there is a need for improved interactions in 2D workspaces to support an author in gradually creating presentation structure. Existing techniques for grouping and structuring text content were also not designed for zoomable environments. As a result, zoom-compatible techniques are needed to aid in moving from informally organized spatial arrangements to the more formal representations used in the final presentation media. This thesis explores a design in which information can be incrementally and tentatively organized into multiple levels and viewed robustly at these different levels.

Arranging information into multiple levels is a key principle in organizing it. This enables a viewer to see the forest for the trees, that is, to see the big points without being overwhelmed by the little ones. However, during authoring, the aggregation of ideas into groups and levels is a delicate matter. A field of ideas without structure can be overwhelming in the sheer number of objects. However, premature assignment of items into groups and levels can also be problematic and make it difficult to move from a mediocre organization into a better one. A design issue explored by this thesis is how a ZUI workspace can enable tentative and exploratory creation of an organization fostering an efficient search for a good organization of ideas for a presentation.

1.3.2 Delivering Presentations

ZUIs are also a natural medium for delivering presentations. ZUIs facilitate putting structure on a presentation that extends beyond the slide boundaries of the traditional slide show metaphor. Because this structure can be reflected in the presentation's layout in the space, the structure actually becomes part of the presentation. In moving between points in the presentation, ZUIs can reveal this structure to the audience through viewpoint animations. This provides a natural analogue to traditional slide show outlines or overviews. But in contrast to traditional outline slides, these overviews remain consistent with the presentation organization because they are views of the actual presentation structure. In addition, these overviews are inherently available at multiple levels since the visual arrangement reflects the multi-level semantic organization. Research also suggests that animations, such as those used in ZUIs, may assist the viewer in integrating multiple views of structural components [7, 12]. Moreover, presenters may be more inclined to include additional overviews if they are presented naturally in the course of transitioning between slides rather than being explicitly inserted as additional content in the presentation.

By improving overviews and the visibility of the presentation structure, ZUIs may also help the audience ask timelier and more informed questions. ZUI overviews can naturally indicate to the audience when a topic has been completed and remind them of topics that are yet to come. They can also indicate where more detailed information is available on a topic but that was not covered by the presenter. This allows the audience to confidently probe for more details, further clarification, or additional examples when they were not included in the original presentation script.

An additional benefit to having the presentation data arranged on a single ZUI surface is that the overviews and structure are made visible by the ZUI animations during unexpected transitions in the presentations. The characteristics of these ZUI animations, including length and speed, can also reflect the semantic distance between two topics. This combination of overview and animation reinforces for the audience both the context of where they were in the presentation and the context of where they are going. The result is that ZUIs provide visual support to the presenter in transitioning between ideas in the presentation. This is particularly important for improvised transitions or movements between widely separated pieces of the presentation since presenters have not anticipated or practiced oral transitions for these improvisations.

Presenters also typically organize their presentations conceptually using a hierarchical structure (for example, see [33, 55]). As a result, the ZUI navigation operations (zoom-out and zoom-in) can use this structure to facilitate a more efficient search when jumping between ideas in the presentation. Rather than the fundamentally linear O(n) navigation search of the slide show metaphor, ZUIs can support a less complex O(log(n)) hierarchical search. This ultimately means that the ZUI navigation controls require less attention, freeing presenters to focus on what they are going to say.

Again in addition to these ZUI benefits, usability challenges also arise in applying ZUIs to presentation delivery. One of the primary difficulties is in navigating a zoomable environment. Existing navigation techniques in these types of environments are extremely powerful and flexible, however, they are often difficult to use due to the controls necessary for navigating in an additional dimension. Improving navigation is particularly important for presentation delivery as the presenter typically navigates the presentation in view of the audience. In this context, mistakes can both embarrass the presenter and waste valuable time. What is needed is a ZUI navigation framework to support highly interactive presentations that can be traversed according to the interests of the audience.

In addition, existing ZUI authoring tools are not specialized for presentation tasks. As a result, many common presentation operations involving talk structure, paths and orderings, and polished spatial layouts are not directly supported in existing ZUI authoring tools. Yet, before a presentation can be delivered using a ZUI, each of these individual components must be specified. Because these components multiply the amount of work needed to author a ZUI presentation, one of the biggest obstacles to using ZUIs for presentation delivery is providing tools to simplify or, better still, automate these tasks. Yet, these simplified tools must also provide the presenter with ultimate control in mapping semantic attributes, such as order and hierarchy, to visual attributes, such as positions and sizes.

Lastly, the ZUI environment also raises the conceptual question of whether the slide metaphor is still appropriate. The ability to view the presentation structure in a single continuous space at multiple zoom levels introduces issues around how to best represent information relationships, both for utilizing human spatial abilities, readability, and general visual appearance.

1.4 Niagara

The usability challenges for authoring presentation content are addressed in this work in a tool called Niagara, shown in Figure 1. Niagara provides a freeform zoomable workspace with objects and interactions to support authoring presentation content while avoiding the most common authoring distractions. Several new interaction techniques are introduced to deal with the complexities of authoring in ZUIs.

One of the most important techniques introduced here for text authoring in zoomable environments is automatic text reduction. Automatic text reduction combines standard geometric zooming and summarization to create more meaningful views of text objects when zoomed out. These enhanced zoomed-out views allow the author to maintain an awareness of object content while also mimicking the global context provided by zoomed-out overviews.

Another important technique for authoring presentation content is automatic grouping. Automatic grouping acts as a bridge between the formal structures used in more polished organizations, such as slides and folders, and the informal organization that is used in early brainstorming and clustering of ideas. Automatic grouping allows the user to explicitly create, modify, or destroy groups of objects simply by changing object arrangements in the space. These groups can then offer many of the advantages of more formal representations including: having a label, supporting group-specific system operations, and having the ability to be modified by the user as a group. Moreover, these groups can be assigned unique colors to further improve the recognizability of groups of text objects when zoomed out.

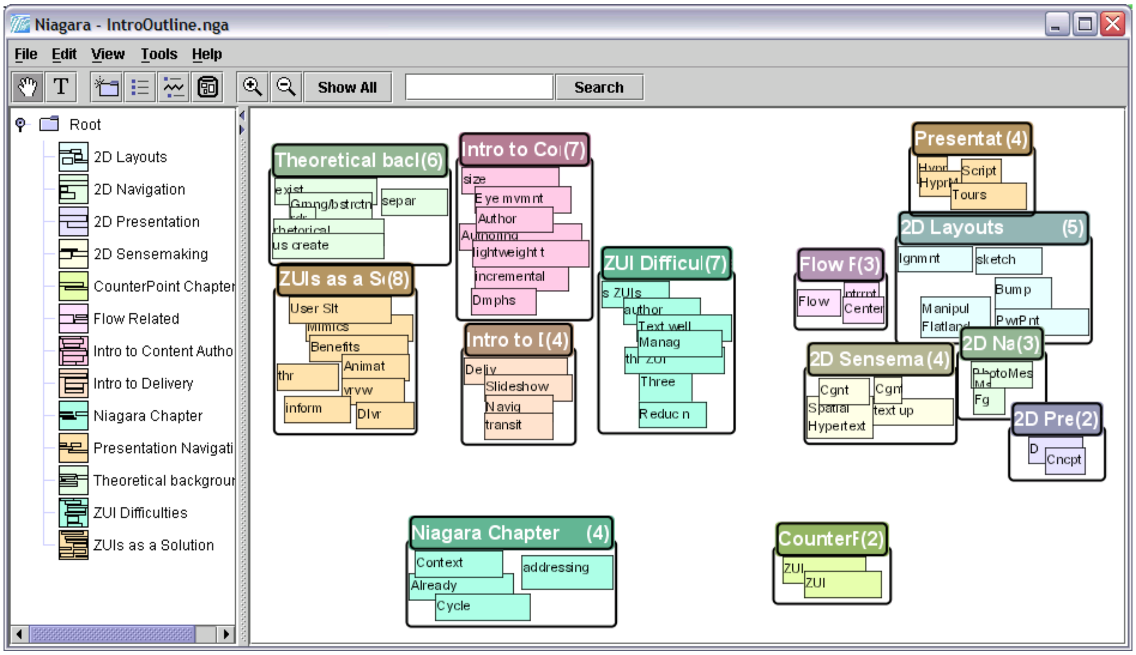
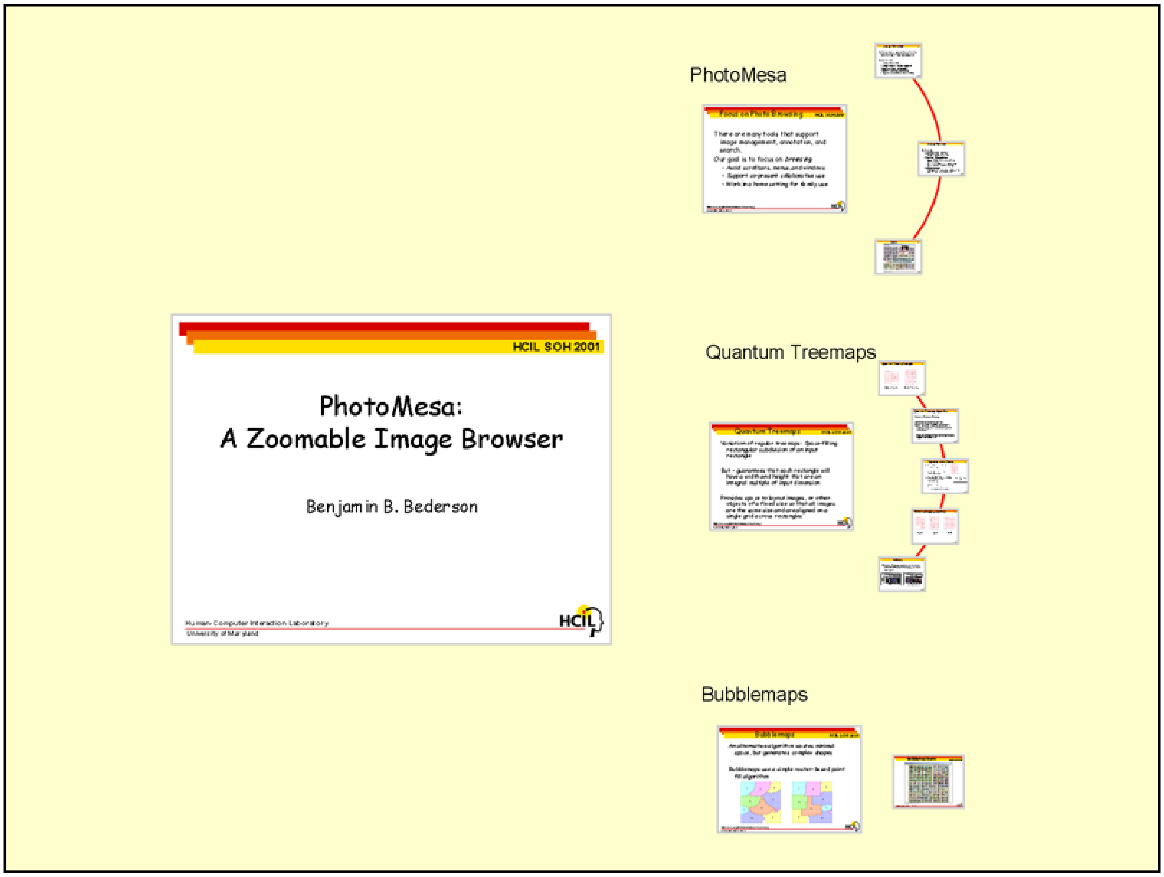


Figure 1: A screenshot of Niagara being used to author parts of this document.

1.5 *CounterPoint*

This work also introduces CounterPoint, shown in Figure 2, as a presentation tool based on the ZUI metaphor. As such it offers solutions to the problems presenters face with the slide show metaphor. CounterPoint also presents solutions to many of the common usability challenges associated with ZUI environments. CounterPoint simplifies the authoring of zooming presentations by extending the slide metaphor to zoomable space. As a result, much of the complexity of authoring in a ZUI environment is simplified to the more familiar task of slide layout using commonly available tools.

CounterPoint then provides tools to organize the individual slides into a single unified hierarchy. CounterPoint uses this hierarchy to simplify several of the ZUI specific authoring tasks. First the spatial layout of the slides can be created based on this structural organization. Similarly, CounterPoint can also use the structural hierarchy to determine a default ordering of the slides for the presentation's scripted path.



HCIL SOH 2001

PhotoMesa:
A Zoomable Image Browser

Benjamin B. Bederson

Human-Computer Interaction Laboratory
University of Maryland

Figure 2: A screenshot of a CounterPoint presentation on PhotoMesa [6].

1.6 Contents

Chapter 2 describes previous work relating to ZUI authoring presentation content and the research involved in designing Niagara. Chapter 3 describes previous work related to computer-based presentation delivery and the research involved in designing CounterPoint. Chapter 4 discusses Niagara’s unique features and some of the more interesting implementation details. Chapter 5 describes CounterPoint’s architecture and some unique features. Chapter 6 offers some areas for future work in both presentation authoring and delivery. Chapter 7 concludes with a summary of the major results and contributes of this dissertation. Appendix A provides the questionnaires used in various user studies throughout the dissertation.

Chapter 2:
Authoring Presentations and Niagara

2.1 Related Work

Computer supported authoring tools have been one of the long term goals of computer science research, dating back to the work of Bush [15], Sutherland [108], and Engelbart [31]. As a result, there have been many computer systems focusing on many different aspects of computer supported authoring. One particular subset of computer authoring tools has focused on using 2D workspaces for authoring and organizing abstract information. Some of the more related systems to the current work include Aquanet [68], Beach [90], Cognoter [33], Data Mountain [93], Dolphin [45], Flatland [74], gIBIS [20], the Interactive Mural [44], MUSE [37], Notecards [46], OneNote [79], PowerPoint [89], Storyspace [107], TinderBox [110], Tivoli [84], VIKI [70], VKB [100], and Web Squirrel [115].

Niagara builds on individual aspects of many of these tools as described below. In addition, although previous systems address similar tasks to those addressed in this chapter, they have not been systematically studied to determine which user interface techniques are best suited for different authoring tasks. Nevertheless, some of the most related evaluations below are reviewed below.

2.1.1 Authoring Systems and Techniques

The work in this chapter concentrates on reducing the formality of 2D workspaces for authoring abstract data to better match the tools to the task. A similar emphasis can also be found in the work of Marshall and Shipman, who have explored this idea of reducing formality in a progression of authoring tools starting with Aquanet [68], then

VIKI [70], and most recently VKB [100]. Each of these systems was built around a free-form 2D workspace where the author could create, organize, and structure information in different ways. In the Aquanet system, users were able to specify various characteristics of their information through schemas or templates. Marshall et al. found that users tended to avoid these formal specifications whenever possible [68]. As a result, the subsequent tools, VIKI and VKB, allow the user to signify relationships through spatial arrangement and visual characteristics. These types of informal structures allow users to refine and evolve relationships over time rather than having to specify them up front. Marshall, Shipman, and colleagues have concentrated on the different kinds of structure and representations that allow for emergent structure. However, unlike the work in this chapter, they have not concentrated on studying the various user interface elements and interaction techniques employed for the task.

There are also commercial implementations of authoring systems that use spatial layout and visual attributes for structuring text. Several systems are sold by Eastgate Systems including Web Squirrel [115] and the more recent TinderBox [110]. Web Squirrel is a 2D workspace for organizing internet shortcuts. Users can organize their shortcuts spatially or by dividing them into explicit collections. TinderBox has many of the same features and interactions as Web Squirrel. However, rather than focusing on web shortcuts, TinderBox supports creating and organizing arbitrary text notes. Microsoft is also preparing to release a related system called OneNote [79]. OneNote virtually emulates a physical notebook. It allows authors to create and spatially organize notes, lists, images, URLs and other objects on virtual 2D pages.

The software prototypes described in this chapter differ from these previous systems in that they introduce several new techniques for interacting with text objects in a zoomable 2D workspace. Two of the most prominent interface techniques introduced in our work are automatic text reduction, a semantic zooming technique for text, and automatic grouping, an interaction technique to facilitate a natural transition from spatial clusters to formal structure. Previous work relating to these two features is described below.

2.1.1.1 Semantic Zooming of Text

Niagara uses the ZUI paradigm for authoring and organizing text. As a result, strategies for presenting text in zoomable environments are an important component of this work. One well known technique used to represent objects in zoomable environments is semantic zooming. Semantic zooming involves changing an object's appearance based on the current zoom level and has been an integral part of Zoomable User Interfaces since they were first created [9, 10, 85]. Perlin and Fox' original ZUI paper [85] even describes a semantically zoomable text object: "When text is visibly small it appears only as a title. As the user zooms in, this expands to include an abstract. Further zooming reveals first an outline with short text descriptions, then finally the full text." A host of previous systems have employed this kind of semantic text zooming. One specific example is the DOI tree [17] which switches between several predefined text representations based on available space. Our approach differs from this one in that it does not require pre-defined levels of abstraction. Instead, it dynamically generates text representations based on the space requirements at the current zoom level using various linguistic techniques.

Text representations were also addressed in the work of Shipman et al. They proposed reducing text object sizes to address limited display space in a 2D authoring environment [102]. Their work concentrated on allocating space for a collection of text objects based on multiple foci visualization techniques. In contrast, our text reduction technique focuses on automatically generating textual representations for changing space requirements. Restated, the technique of Shipman et al. allocates space for objects while our technique creates more meaningful representations for text objects.

Another technique used to improve reduced size representations of textual documents is enhanced thumbnails [117]. Enhanced thumbnails overlay highlighted text labels on top of ordinary web page thumbnails. These thumbnails were shown to offer both the benefits of ordinary thumbnails plus the benefits of text summaries for finding information in web searches. The enhanced thumbnails technique makes use of structure in the HTML document and keywords used in the internet search. In contrast, the automatic text reduction technique described in this chapter does not rely on these assumptions. Instead, automatic text reduction is intended to provide more meaningful reduced size representations for arbitrary text objects.

Reduced text representations have also been the focus of a large body of work relating to summarization [67]. In summarization, the goal of the text reduction is to identify the most salient information. A summarized text is intended as a stand-in for the longer text and should be comprehensible, correct, salient, and representative. In Niagara, the goal is different. Text need not carry the full meaning and need not be comprehensible or strictly correct. Instead, Niagara’s reduced text representations are intended to act as recognizable visual cues and reminders of the full text. Niagara’s reduction algorithms are also combined with zooming to dynamically access a range of reduced representations for a given object.

2.1.1.2 Automatic Structuring Behaviors

Moving from informal spatial organizations to more formal structures, such as outlines, is also an important component of authoring in Niagara. One previous technique used to create structure from spatial organizations is spatial parsing. VIKI implemented a form of spatial parsing based on object positions, sizes, and other visual attributes that was used to infer object relationships [103]. These inferred relationships are then shown to the author in selecting groups of objects, suggesting formal collections of objects, and suggesting composite types or templates. Niagara's automatic grouping differs in that it is invoked explicitly by the user when objects are moved close together. This immediately and persistently reflects the inferred groups to the author. Niagara also automatically modifies object colors when adding or removing objects from a group. Although the techniques in both systems are intended to reduce the overhead of creating structure, they otherwise differ in purpose. VIKI allows users to create customized composite types and relationships using position and visual properties; Niagara uses automatic grouping and color to create representations that emphasize group membership and enhance recognizability under zooming.

Another system, WebSquirrel [115], uses a variant of spatial positioning to determine group membership. In WebSquirrel, a user can establish a labeled signpost for a group. Items near a signpost or near other items near a signpost are then treated as belonging to the group and a bounding rectangle is drawn surrounding the group items. One of the problems with this technique is that group membership can potentially be confusing or unpredictable. For example, when groups are moved near each other, WebSquirrel can re-assign items from one group to another according to relative distances to signposts. A similar difficulty is that if an intermediate item in an “island chain of items” is moved out of a group, then the distant part of the chain becomes isolated and is no longer considered part of the group. Both of these problems are solved in Niagara by making group assignment be an explicit user-directed operation. An object is only added or removed from a group when the user explicitly moves the object into or out of the group.

Another recent system from Microsoft, called OneNote [79], also uses explicit user actions to automatically create structure. OneNote automatically adds, removes, and indents objects in bulleted lists when they are moved in the OneNote workspace. Objects can be positioned anywhere in the OneNote workspace, but they can only be structured into well-aligned, hierarchical lists. In contrast, Niagara does not enforce any layout constraints on automatically created groups. This allows authors to position objects within groups to informally indicate relationships.

Components of the Beach [90] groupware system also provide techniques for moving from an unstructured 2D workspace to a hierarchical organization. The system provides interaction rules for different kinds of objects that help in gradually creating structure. In particular, title objects exert a kind of “magnetic” force that can be used to attract and formalize a cluster of related objects in the workspace. Niagara’s approach differs from this approach in that group objects never have to be explicitly created. Instead, groups are created and deleted simply by changing the position of objects in the workspace.

Many other systems such as PhotoMesa [6] use automatically assigned colors as part of a scheme to represent group membership. Again, Niagara differs from these in that it supports a user-created, free-form layout rather than enforcing a structured layout such as a treemap. Moreover, Niagara also creates its groups and assigns colors based on the user's explicit actions in positioning objects in the 2D workspace.

2.1.2 Evaluations

This chapter describes a small qualitative user study consisting of 4 subjects that looks at how people's behavior differs in organizing information for a presentation using several different kinds of 2D workspaces including PowerPoint, sticky notes on a whiteboard, and our own software prototype. A number of previous user studies have looked at similar tasks using similar 2D authoring environment. These previous evaluations are described in the first section below.

This chapter also presents two larger quantitative studies comparing ZUIs to more traditional techniques for organization tasks. Our first quantitative study compares the performance of 13 subjects organizing shapes into groups based on common visual properties. The two interface conditions compared in the study are a ZUI and a folder-based interface. Our second quantitative study replicates the previous study but replaces shapes with text objects. This second study consists of 14 subjects.

No controlled studies that we are aware of have compared ZUIs to other interface techniques for this type of task. Nonetheless, there have been a number of studies that have compared ZUIs or zooming to different interface techniques for other types of tasks. These studies are described in the second section below.

2.1.2.1 Evaluations of 2D Authoring Environments

Jones and Dumais performed several experiments exploring the benefits of 2D spatial location versus naming for a news article retrieval task [60]. This study did not directly involve computers but rather compared spatial location with naming as filing strategies for paper notes. The tasks in the study required subjects to read articles and then file paper notes with numbered references to the articles using one of 6 conditions: 1) a subject-selected position of a numbered label on a piece of paper, 2) a randomly ordered paper list with subject-selected names, 3) a subject-selected position of a named label on a piece of paper, 4) a combination of the named list and the numbered label position, 5) the numbered label position on a paper schematic of an office, and 6) the numbered label position in a physical office. The study consisted of a total of 120 subjects and used a between subjects design. The results from the study revealed that retrieval with location-only cues was significantly less accurate than with name-only cues, even when names were limited to 2 letters. Retrieval of numbered labels in the physical office was also less accurate than with the name-only cues on paper, though the difference was only marginally significant. One potential problem in this study is that they assumed that the value of spatial location comes from choosing an object's position. However, in their own literature review they cite several studies that demonstrate an incidental and effortless memory for the location of information, such as the position of text on a page in a book. This suggests that merely seeing an object's position, rather than choosing it, may trigger spatial memory. The study is also not representative of actual work practice since the spatial arrangements used in most real-world applications, such as those described in this chapter, are reinforced by rearranging existing objects as new objects arrive. The studies described in our work essentially explore these same issues of naming versus location but focus on an organization task in a more realistic setting.

Notecards was one of the earliest computer-based 2D authoring environments to be studied [46]. Notecards was a hypertext authoring environment that allowed the author to create virtual note cards that the user could then organize and link together. The system also allowed many notes to be open at once in separate windows. The author could then organize these windows spatially in the Notecards workspace. Monty performed a longitudinal study of a single author using this system over a 7 month period for a paper writing task [72]. Monty identified a number of common operations in the author's writing process as well as problem areas in the tool's support of these processes. The qualitative study in this chapter differs from this study in that it focuses on the similarities and differences in the organization component of an authoring task across 3 different tools. The study also includes multiple subjects to identify common operations across different users.

A similar longitudinal usage study was performed for the Aquanet system [68] (system described above). This study consisted of a small set of Aquanet users making sense of a large collection of information over the course of 2 years. The study focused on the kinds of structure and relationships people created in authoring with Aquanet. The study described in this chapter differs in that it identifies behaviors across several different authoring tools. The current study also specifically focuses on organization behaviors and does not explore the issues of linking or argument structure.

Marshall et al. also performed a qualitative study of the VIKI system [69] (system described above). This study looked at 15 subjects performing a time-constrained task that required the participant to make a product purchase decision based on information they gleaned from 75 articles. Subjects performed the task using either paper copies of the articles and standard office supplies or electronic copies of the articles and a version of VIKI. Although it compares similar tools to those compared in the study in this chapter, the VIKI study addresses a different task than this work. As mentioned, the study in this chapter focuses on the organization component of a presentation authoring task.

A more recent study by Prante et al. [90] looked at 3 commercially available groupware tools for brainstorming and organizing ideas among co-located groups. The study looked at 45 subjects performing a set of 3 creative tasks using a different one of the 3 groupware tools for each task. Subjects solved the tasks in groups of 3 connected via networked computers. Despite being in the same room, subjects were not allowed to talk to one another. The first tool used in the task provided a structured mind-mapping interface and required subjects to take turns in modifying the workspace. The second tool provided an unstructured whiteboard-like surface that also required subjects to take turns in modifying the workspace. The third tool provided a similar whiteboard-like surface but allowed multiple participants in the group to modify the workspace at the same time. The study used a within subjects design and found that subjects using the simultaneously editable workspace created significantly more objects than in the other turn-taking tools. More relevant for this work, subjects using the structured workspace created more ideas than those using the unstructured workspace. In general, this study looks at similar kinds of tasks to those in this chapter. However, the work in this chapter is primarily concerned with an individual's process in performing the tasks rather than a group. Moreover, this chapter focuses on the organizing component of an authoring task rather than idea generation.

2.1.2.2 Evaluations of ZUIs and Zooming

One of the earliest studies involving zooming was performed by Beard et al. [5]. This study compared scrollbars to an overview window for navigating a balanced binary tree. The study also included two different kinds of controls for the overview window. The first control type, described as “roaming,” involved dragging a detail region in the overview to change the view in the main window. The second control type, described as “roam and zoom,” required the user to drag a new rectangle in the overview each time to specify the view in the main window. This allowed the user to zoom out and zoom in by dragging bigger and smaller rectangles respectively. The study consisted of 6 subjects and had a within subjects design. The primary result from the study suggests that using an overview window is faster than scrolling for navigation in certain kinds of environments. No significant difference was observed between the different overview controls. The studies in the current chapter differ from this work in that they involve organizing information rather than simply navigating it. In addition, the kind of zooming studied in this chapter is also different in that it is driven by directly interacting with the workspace rather than with an overview window. The zooming described here is also animated to assist the user in understanding how the view has changed.

Ghosh and Shneiderman performed a related study comparing an overview plus detail interface with a zooming-only interface for visualizing timeline data [41]. The tasks in this study required subjects to find answers to information contained on the timeline. The study included 14 subjects and had a between-subjects design. The study presents a borderline significant result suggesting that overview plus detail is slightly faster than the zooming only condition. The studies in this chapter again differ from this study in that they look at different tasks. In addition, the data spaces in this study are essentially one dimensional rather than 2D as typically occur in ZUIs.

Another comparison of zooming and overview windows was performed by Hornbæk et al. [53]. This study compared a ZUI plus an overview window to a ZUI without an overview window for a set of map tasks. The ZUI in both conditions allowed the user to pan, zoom in, and zoom out using mouse gestures. The overview window allowed the user to move and resize a detail rectangle to change the view in the main window. The study included two kinds of tasks including navigating to a particular target and browsing for specific map features. The study also examined a single level map, where all objects were labeled with the same size font, and a multi level map, where map labels were sized according to the geographic object that they labeled. The study consisted of 32 subjects and used a within subjects design. The primary results from the study indicate that subjects are faster using the ZUI without an overview when the task involved the multilevel map or navigation. Here again, this study differs from the controlled studies in this chapter in the type of task examined. Rather than map-based tasks, this chapter examines tasks involving abstract information. Likewise, rather than browsing or navigation tasks, the current chapter looks at organization tasks.

Plumlee et al. performed a similar comparison of zooming versus a multiple window strategy [88]. The tasks they investigated involved comparing groups of shapes laid out in a large 2D workspace to determine whether a probe group matches a sample group. Tasks varied in the number of shapes to be compared between groups. Prior to running a user study, the authors present a cognitive model of the task based on the capacity of working memory. They then used this model to predict the performance of a zooming interface versus a multi-window interface for the task. The model predicted that the multi-window interface would perform better as the number of objects in the groups increased. They then performed a user study with 17 users comparing these two interface techniques for the group comparison tasks. The study analysis used a within subject design and found a significant effect of interface type on completion times. This study is quite similar to the studies performed in this chapter. However, this study has intentionally negated some of the benefits of zooming by using textured backgrounds on the zooming workspace to camouflage objects. In contrast, our work tries to preserve as much of an object's value as possible when zoomed out. In addition, rather than comparing ZUIs to multi window strategies as was done in this study, the studies in this chapter compare ZUIs to a folder or paging strategy.

Combs et al. compared ZUIs to several other commercially available image management systems for a collection of image retrieval tasks [19]. The ZUI used in this study provided a zoomable grid of images in which the user could zoom and pan. The other systems compared in the study used a 2D grid of thumbnails, a layout of images on a 3D plain, and an arrangement of images in a spin-able 3D "lazy susan." The study used two image tasks including one that involved searching for a specific image and the second that involved browsing for an image to send to friend. Three different sized image sets were also examined including 25, 75, and 225 images. Thirty subjects participated in the study, which combined a within subjects and a between subjects design. The results from the study indicated that the ZUI and the 2D image browsers have lower task completion times and higher subjective satisfaction than the 3D browsers. As with the other previous studies of ZUIs, these differ from the controlled studies in this chapter in that they only look at searching and browsing and do not look at organization. Moreover, it is expected that graphics-heavy tasks are more naturally suited for ZUIs than the textual tasks in this chapter since graphics retain more of their value when reduced in size.

Another study, by Paez et al, looked at using ZUIs for a document reading task [80]. They performed a between-subjects experiment with 36 participants comparing a document laid out in a ZUI workspace to a hyperlinked document in a traditional web browser. The task in the study required the user to find the answer to 5 questions within the document using one of the two interface types. They found no significant differences between interface types in completion times, comprehension measures, disorientation measures, or subjective satisfaction. However, subjects in the study did report that the ZUI provided good skimming and overview capabilities and was easy to learn. This study is interesting since it shows that users were able to navigate a ZUI workspace to complete a reading-heavy task. The studies in this chapter extend this work by looking at tasks that involve authoring in addition to reading text.

Looking again at hypertext, Hightower et al. compared a web browser with a ZUI history tool to an unaided web browser in two usability studies [52]. The ZUI history tool laid out thumbnails of visited web pages in a 2D tree representing their hierarchical web history. The user was then able to interact with the tool to navigate the tree and reload previously visited web pages into the browser. The tasks used in the first usability studied required subjects to find particular facts on individual web pages and make comparisons between groups of web pages. This first study consisted of 36 subjects and used a within subjects design. The results from the first study indicated a subjective preference for PadPrints and a decrease in the number of pages accessed with PadPrints. The second study required subjects to navigate a group of web pages to find the answers to an initial set of questions. They were then asked a set of questions that required them to return to pages visited in the process of answering these questions. The second study consisted of 36 subjects and also used a within subjects design. The results from this second study indicated faster completion times with PadPrints for the tasks that involved revisiting pages. Page accesses were also again lower with the PadPrints condition. Once again the studies in this chapter differ from the PadPrints study in that they address authoring rather than navigation or search tasks.

Robertson et al. carried out a related study comparing a spatial interface called Data Mountain to the more traditional hierarchical system in Internet Explorer 4 for organizing web favorites [93]. The Data Mountain presented the user with an inclined plane on which they could spatially arrange thumbnails of bookmarked web pages. The Data Mountain interface was not a traditional 2D workspace and did not support a traditional notion of zooming. However, as web page thumbnails were moved closer to the bottom of the screen they grew in size, effectively giving the user a limited zoom control on individual pages. In addition, the interface provided a mechanism to view a single web page in detail which essentially “zoomed in” even further on the individual page. Robertson et al. then performed a user study that required subjects to create and organize bookmarks for 100 web pages. Once the organization was completed, subjects were then given tasks asking them to find a particular bookmark based on 4 different cues. The cues included title, summary, thumbnail, and combination of all three. The study consisted of 32 users and used a between subjects design. The study results indicated that the spatial interface reduced the time to store the bookmarks as well as reducing the number of retrieval errors made. The bookmark organization task in this study is similar to the organization task explored in this chapter. However, rather than focusing on retrieval, our work focuses on the benefits of spatial arrangements for the organization task. In addition, our work examines organization of text only content rather than more visual web page thumbnails.

2.2 Niagara

The authoring model described in the introduction included four phases: generating ideas, organizing, composing, and revising. Although there is certainly research needed in each of these areas, this chapter focuses on the organization component of the process. One of the reasons for this focus comes from previous research in computer based authoring environments, such as Notecards [46] and VIKI [101], that suggests that many of the difficulties in authoring tasks arise in defining the work's structure and the relationships between objects. Further, the organization component seems to be one of the places where the problem of getting an overview of the text is particularly difficult on a computer. In the Notecards experiment described above, the author using Notecards in the study reported a desire to print his notes and "spread them out on a table" in order to organize them [72]. Similarly, the research cited by Severinson Eklundh [97] suggests that organization is one of the places where computer-based authoring tools provide insufficient overviews.

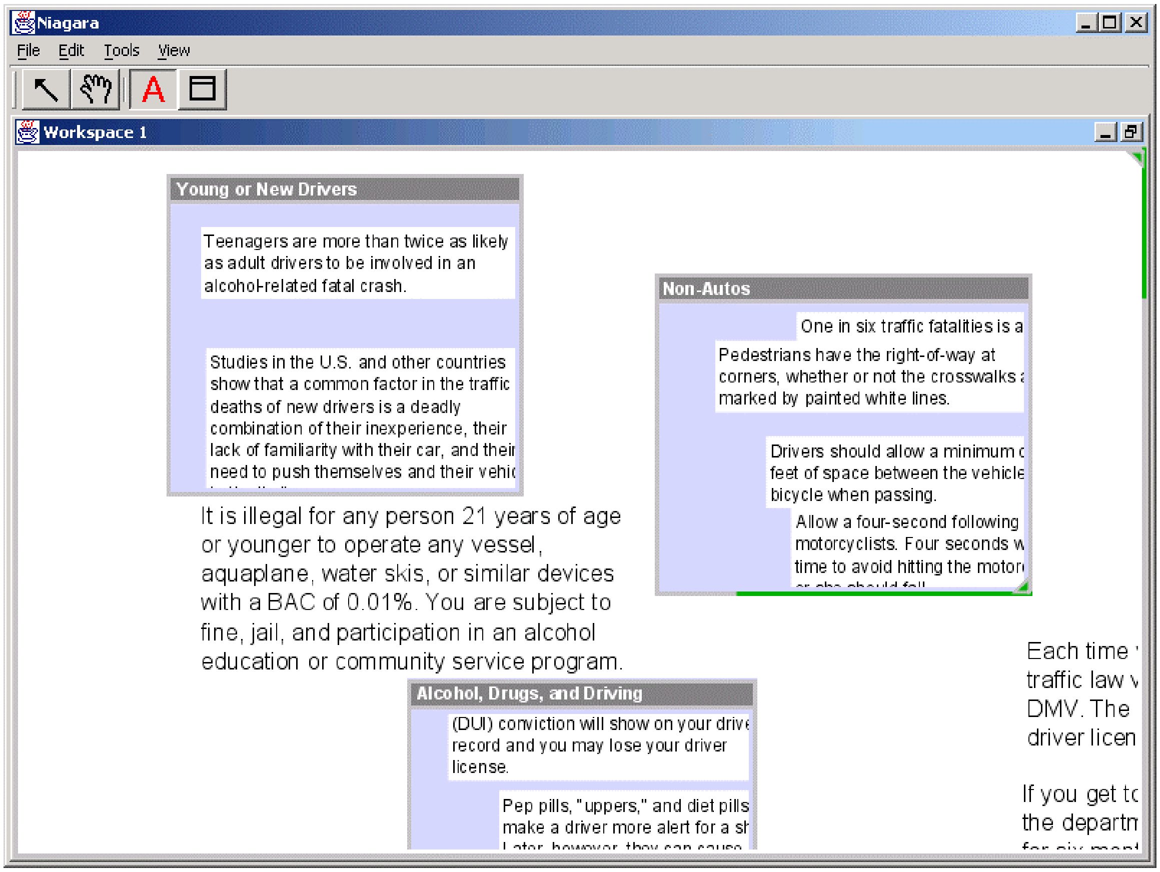
This chapter addresses these text organization challenges in the specific context of authoring content for presentations. However, as discussed in the introduction, there are similar constraints to those in presentation authoring across many kinds of text-authoring tasks. In addition, the interaction techniques described in this chapter make no explicit assumption as to the final use of the organized information. Consequently, the techniques described in this chapter are likely to apply to many different authoring tasks.

The techniques described in this chapter for organizing text are implemented as part of an authoring tool called Niagara. A primary goal in developing Niagara was to understand the extent to which improved interactions and automation on a small but active display could reclaim the overview capabilities inherent in large passive physical surfaces such as whiteboards and tabletops.

Niagara was refined through a cycle of building prototypes and evaluating the prototypes with user studies. Below, three iterations in this cycle are described. In each iteration, insights were discovered into presentation authoring and the optimal computer interfaces needed to support this task. These three iterations led to a controlled study comparing folders to zooming for a shape-based organization task. This shape study was then duplicated in a nearly identical study involving text. Finally, the chapter describes some initial results involving real users organizing information for real presentations.

2.3 First Design Iteration - Niagara version 1

Because it addresses problems similar to those addressed by spatial hypertext systems, our work began with a prototype similar to several well known spatial hypertext systems including VIKI, VKB, and Web Squirrel [70, 100, 115]. This prototype provided a starting point that allowed us to observe actual users interacting with the system to see what worked and what didn't. Figure 3 illustrates a populated workspace in Niagara version 1.



Teenagers are more than twice as likely as adult drivers to be involved in an alcohol-related fatal crash.

Studies in the U.S. and other countries show that a common factor in the traffic deaths of new drivers is a deadly combination of their inexperience, their lack of familiarity with their car, and their need to push themselves and their vehic

It is illegal for any person 21 years of age or younger to operate any vessel, aquaplane, water skis, or similar devices with a BAC of 0.01%. You are subject to fine, jail, and participation in an alcohol education or community service program.

One in six traffic fatalities is a

Pedestrians have the right-of-way at corners, whether or not the crosswalks a marked by painted white lines.

Drivers should allow a minimum c feet of space between the vehicle bicycle when passing.

Allow a four-second following motorcyclists. Four seconds w time to avoid hitting the motor

(DUI) conviction will show on your drive record and you may lose your driver license.

Pep pills, "uppers," and diet pills make a driver more alert for a sh

Each time traffic law v DMV. The driver licen

If you get tc the departn

Figure 3: A screenshot of the first version of Niagara with text objects and collections.

The goal of Niagara's freeform workspace is to allow the author to easily create and maintain information. The meaning of the information in the workspace is held both in the content (that is, words) as well as in the spatial arrangement. An author tends to locate closely related items near each other so that they can be conveniently used together. Parallel arrangements of items suggest parallel meanings. Persistence of arrangement speeds access in that the author roughly remembers the location of items and does not need to search for them. It also speeds understanding when an author uses parallel visual structures to imply parallel meanings. The user invests time to arrange items, and presumably benefits both from convenient use and speedier access because he remembers where things are.

There is a delicate balance in supporting an author using a 2D workspace like Niagara's. If the system automatically changes an author's arrangements, the author may have to invest time and energy to learn the new arrangement and possibly to rearrange it. Conversely, if the system provides no assistance in positioning objects, then the user can waste time aligning objects or making space for new objects. Reflection on our own computer authoring experiences suggested that a significant amount of time was devoted to this latter case of rearranging and aligning objects. Our reflections were reinforced by research suggesting that spatial arrangements that involve overlap are undesirable and could even lead to users losing objects [93]. Consequently, Niagara implemented a no-occlusion contract in its initial version. This contract prevented objects from obscuring one another so that objects could not become hidden. Whenever objects were moved, were resized, or grew to accept new content, they automatically bumped other objects out of the way. This contract brought a physical quality to the interface so that every object could be used as a tool to align or bump other objects out of the way. More specific object interaction contracts were also implemented to automate the creation of common alignments. Further details of the specific features and interactions in this version of Niagara are described in Chapter 4 on Feature and Implementation Details.

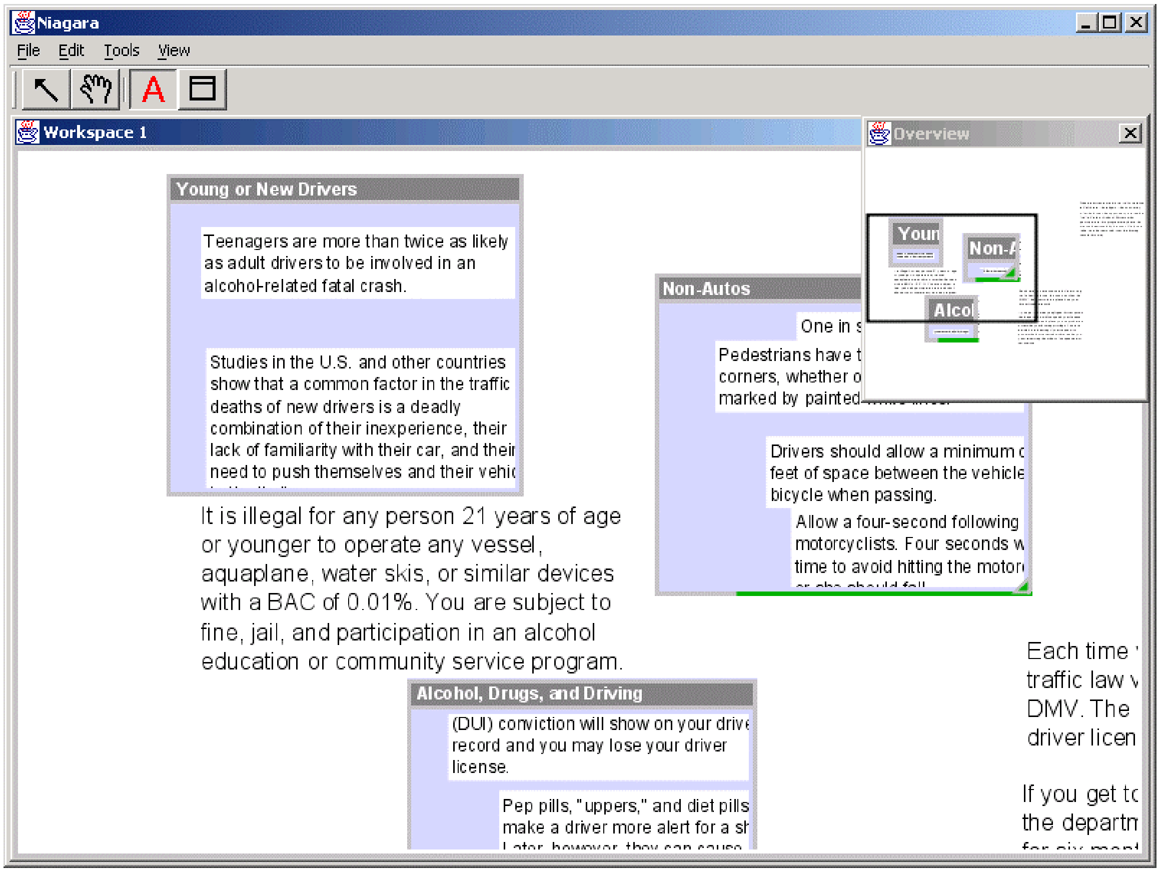
2.3.1 Study

Following the development of this first Niagara prototype, a small informal observational study using two subjects was performed. The subjects were told to organize a set of short text passages into groups with the ultimate purpose of presenting the information. The task consisted of several supporting operations such as reading the passages, clustering related items, creating and labeling collections, and repairing the evolving structure. The text passages for the study were taken from the California Drivers Handbook. Supplying the content in this way provided some control over the scope of the organizations created and also simplified the task by eliminating some activities such as generating or gathering information.

This study quickly illuminated a recurring problem with the Niagara version 1 prototype. As expected, the number of text items quickly exceeded the available screen space. Once this occurred, the users had problems managing the populated workspace as it extended beyond their immediate view. As a result, a significant amount of panning was necessary to manage the widely distributed objects. Operations such as finding and moving items became correspondingly more expensive and dominated the task completion times.

2.4 Second Design Iteration - Niagara version 2

The most significant change to the Niagara interface that resulted from the initial informal study was the addition of a spatial overview, shown in the upper right hand corner of Figure 4. The overview was meant to address the problems that subjects encountered in managing a workspace that were larger than the containing window. As with overviews in other types of programs [105], it was expected that the spatial overview in Niagara would provide a mechanism for rapid navigations as well as a general aid to spatial awareness. In addition, the overview was also expected to accelerate one of the most common operations, namely moving an item to a known collection. Rather than needing to navigate to an off-screen collection, users could drop objects into off-screen collections directly in the overview. This behavior was further encouraged through semantic zooming on the titles of collections (see Figure 4) to maintain their readability in the overview.



Teenagers are more than twice as likely as adult drivers to be involved in an alcohol-related fatal crash.

Studies in the U.S. and other countries show that a common factor in the traffic deaths of new drivers is a deadly combination of their inexperience, their lack of familiarity with their car, and their need to push themselves and their vehic

One in s

Pedestrians have t corners, whether o marked by painted

Drivers should allow a minimum c feet of space between the vehicle bicycle when passing.

Allow a four-second following motorcyclists. Four seconds w time to avoid hitting the motor

It is illegal for any person 21 years of age or younger to operate any vessel, aquaplane, water skis, or similar devices with a BAC of 0.01%. You are subject to fine, jail, and participation in an alcohol education or community service program.

(DUI) conviction will show on your drive record and you may lose your driver license.

Pep pills, "uppers," and diet pills make a driver more alert for a sh

Each time traffic law v DMV. The driver licen

If you get tc the departn

Figure 4: A screenshot of populated workspace in the second version of Niagara. A spatial overview is available in the top right corner of the workspace.

2.4.1 Study

A second observational study was conducted with the second Niagara version containing the spatial overview. This study was intended to provide data on the frequencies and costs of operations involved in this grouping task. This would allow for the identification of any bottlenecks caused by the interface. In order to highlight the tradeoffs of different interface techniques, the study also asked subjects use PowerPoint and sticky notes on a whiteboard. PowerPoint was chosen based on informal feedback from colleagues indicating its use in similar organizational tasks. Similarly, sticky notes on a whiteboard were included because they represent the optimal organization environment, offering tangible interactions and abundant display space.

While it was expected that display size played a dominant role in a user's performance on the task, the study was intended to identify the interface interactions that were most limiting to a user performing the task. Rather than trying to carefully isolate the influence of a specific set of variables, this study was meant to explore the different overall affordances provided by Niagara, PowerPoint, and sticky notes on a whiteboard.

2.4.1.1 Method

The subjects were 4 employees from PARC. Three subjects were members of research staff and one was an administrator. The content used in the study was 3 sets of 60 facts, containing 6 to 60 words, again taken from the California driver's handbook. In the sticky notes condition, the facts were printed on sticky notes such that each sticky note contained a single fact. In the software conditions, the facts appeared one at a time on an "electronic tablet" where the user could drag them into the application. Figure 5 shows a pilot subject using sticky notes to complete the organization task.

The experimenter provided a short tutorial and training period in Niagara and, when needed, PowerPoint. Each trial was limited to 30 minutes or was completed when the subject was satisfied with the grouping of all 60 items. Subjects performed the grouping task using all three tools in a single session. Tool and fact set orderings were counterbalanced to minimize any sequencing effects.

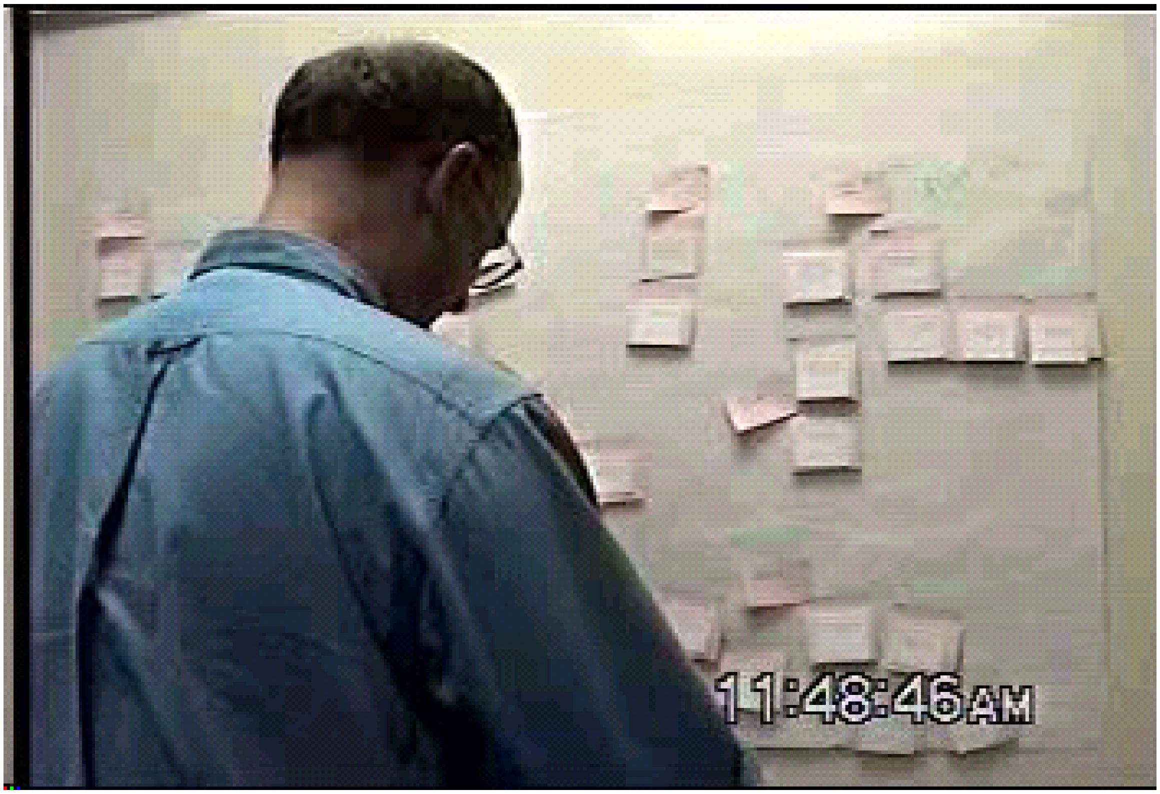


Figure 5: A pilot subject organizing the driving facts using sticky notes on a whiteboard.

All sessions were videotaped and subjects were asked to use a think-aloud protocol. A set of descriptive activities was identified to be used as a set of tool independent operations to code the video. These operations are described in Table 1. Two important components of the task that are noticeably absent from this chart include reading and thinking. These were not counted since they frequently occurred without any visible or audible manifestation. Lastly, since this was primarily an exploratory study, inter-rater reliability was not computed for the coded video.

Following the study, subjects completed a short questionnaire regarding their experiences using each of the tools. This questionnaire is available in Appendix A.

Table 1: A description of the operations coded in the video.

Operation	Description
Create Collection	Creating an explicitly defined collection – with borders or an initial title
Labeling	Changing a label or adding a label to a cluster or a category that was left unlabeled when created
Navigation	Panning, Scrolling, or Zooming (only applicable for software conditions)
Non-Task Adjustment	Non-navigational operations that did not have semantic meaning for the task
Placement	Moving an item from the “tablet” to the workspace
Reclassify	Moving an item from one explicitly defined category to another
Spatial Clustering	Changing the position of an item to indicate a semantic relationship

2.4.1.2 Results

Single factor ANOVAs with repeated measures were used to analyze the results in preference to t-tests since the study compared 3 study conditions. Post hoc tests were then used to reduce the chances of Type I error in performing multiple comparisons. In particular, Scheffé tests were used as they are conservative yet more sensitive for complex (in addition to pairwise) comparisons between conditions than Tukey tests.

If should be noted that parametric tests, including ANOVAs and t-tests, are used throughout this dissertation since the research involves human subjects measures, which tend to be normally distributed [53]. Even when the normality assumption is violated, the t-test and ANOVA are still likely to be valid [53].

Single factor repeated measure ANOVAs were used to evaluate the effect of tool type on completion time, on the frequency of operations coded in the video, and on subjective satisfaction. The means for the collected data are shown in Table 2. These analyses found a significant effect of tool type on completion time (F(2,6)=11.12, p =

0.01), non-task adjustments (F(2,6)=13.68, p = 0.01), collection creation (F(2,6)=40.32, p < 0.001), collection labeling (F(2,6)=6.37, p = 0.03), and for the questionnaire item "What is your overall reaction to using Post-It notes for this task?" (1=Frustrating, 5=Satisfying) (F(2,6)=10.34, p = 0.05).

Post hoc Scheffé tests indicated that sticky notes were significantly different from both the software tools in: shorter completion times (p = .01), fewer non-task related adjustments (p = 0.01), fewer collection creations (p < 0.001), more collection labelings (p = 0.04), and higher subjective satisfaction (p = 0.05). Further Scheffé tests indicated that significantly fewer non-task related adjustments were made (p = 0.03) and more collections were created (p = 0.01) in PowerPoint than in Niagara.

Table 2: The means and standard deviations of dependent variables measured in the study.

	Stickies	**PowerPoint**	**Niagara**
Completion Time (min.)	21.14 (σ 1.65)	28.07 (σ 3.87)	30 (σ 0)
Create Collection	1.0 (σ 2.0)	9.0 (σ 1.41)	6.25 (σ 3.40)
Labeling	9.0 (σ 6.16)	0.25 (σ 0.5)	3.25 (σ 3.95)
Navigation	0 (σ 0)	17.75 (σ 7.8)	14.0 (σ 17.26)
Non-task Adjustments	0.75 (σ 1.5)	18.75 (σ 14.77)	36.5 (σ 9.18)
Placement	60.0 (σ 0)	49.75 (σ 16.17)	57.25 (σ 4.27)
Reclassify	3.25 (σ 4.03)	21.0 (σ 12.14)	19.75 (σ 11.93)
Spatial Clustering	16.25 (σ 6.02)	1.5 (σ 1.73)	5.5 (σ 4.51)
Subjective Satisfaction (1=Frustrating, 5=Satisfying)	5.00 (σ 0.0)	2.75 (σ 1.26)	2.75 (σ 1.26)

2.4.1.3 Discussion

One activity that differentiated all 3 tools was non-task related adjustment. These types of adjustments were likely minimized in the whiteboard condition due to the highly intuitive and well practiced physical interactions. In contrast, these adjustments seemed to be particularly common in the Niagara condition because of quirks, bugs, and performance limitations that were a result of the software's experimental nature.

The difference between the number of collections created in Niagara and PowerPoint is likely a result of the different spatial paradigms suggested by the tools. PowerPoint presents space as slides, where each slide is a disjoint white rectangle surrounded by an infinite gray plane. This seemed to encourage people to use the visual slide boundaries as semantic divisions as well. In contrast, Niagara presents space as a set of nested infinite surfaces. These infinite surfaces seemed to make less urgent the need to divide objects into explicit collections since users were able to indicate groups through whitespace instead.

The remaining differences in completion times, collection creations, and labelings between the whiteboard and the software tools are likely to be the result of the whiteboard's larger display size. Because it offered virtually unlimited display space for this task, the whiteboard allowed users to spread out their information yet still be able to get an overview of that information. Similarly, having the larger surface allowed subjects to quickly scan all the facts with head or eye movements whereas collections on the software tools often required more time-consuming mouse or keyboard interactions for an equivalent scan.

Because the software tools have less display space, users had to develop strategies to compensate for this limitation. One of the first strategies observed was that subjects frequently placed new items into an unlabeled region in the workspace to help identify patterns. This technique was named "staging" because items passed through an initial scanning and comparison stage prior to a more formal organized stage. Once a pattern was detected in this staging area, a new collection was created and the matching facts were moved to that collection. Collections allowed subjects to reclaim display space by reducing the amount of space used by similar facts. In PowerPoint, each collection was typically represented by a single slide that could be reduced to an icon and title in a column on the left edge of the screen. In Niagara, each collection was a rectangular window inside a larger workspace that reduced the scale of the contained items and clipped these items to the boundary of the window. Once similar items were put into collections in either tool, subjects were left with more room to process newly arriving facts.

An early assumption was that labels would be necessary for quickly deciding which group an object belongs to. It was predicted that without labels, subjects would frequently need to reread the items in a cluster to recall its membership criteria. Clearly the small number of labeling operations in the sticky notes condition does not support this prediction. Instead, this behavior is likely to depend on the number of items organized. Since the maximum number of collections subjects created in the study was 10, subjects were likely able to remember a large fraction of these membership criteria at any one time. In the cases where the subject could not remember, a quick scan of one or two items usually served as a sufficient reminder.

Despite their limited use in the whiteboard study condition, labels are likely to be more important in actual presentation authoring tasks that are completed over a number of days rather than in a single half hour session. Over extended periods, the author is more likely to forget group contents and positions between the authoring sessions. In these cases, formal labeled collections are likely to compete with informal spatial clustering as a grouping strategy (see Table 3). In choosing between spatial clustering and labeled collections the user makes tradeoffs between work up front and work later.

Table 3: Tradeoffs the organizer makes in using spatial clusters vs. formal collections.

	Time To:		
	Create	**Modify**	**Determine Membership Criteria**
Informal Spatial Clusters	Low	Low	Medium/High
Formal Collections	Medium/High	High	Low

One last practical observation from this study was that the spatial overview in our first prototype was not effectively utilized. One problem with the overview was that it occluded a portion of the workspace so that objects disappeared beneath it. A second, perhaps more debilitating, problem was that the labels on collections often displayed too few characters to be recognizable. Several examples of these clipped labels can be seen in the overview in Figure 4. This type of clipping on labels was implemented so that objects would use the same proportion of space at all zoom levels. If this proportional space constraint was not enforced, objects would begin to overlap as the space became more crowded. For example, if the "Young or New Drivers" collection label in Figure 4 were not clipped to the collection boundaries then it would overlap the "Non-Autos" collection, potentially reducing readability and accessibility of one or both collections.

Moreover, extending collection labels beyond the collection boundaries tends to reduce the value of spatial location in the overview since label bounds would not correspond directly to collection bounds. Henderson et al. arrived at similar conclusions concerning a spatial overview in developing Rooms [51].

2.5 Third Design Iteration - Niagara version 3

From the preceding study, a tradeoff in the design of the overview was identified. The two primary choices for the overview are a geometrically reduced representation of the workspace or an abstracted representation of the structure of the space. Geometric or spatial overviews are popular in image editing software and map based systems while abstract or structural overviews are popular in presentation editors or hierarchical browsers such as file systems. The spatial overview has the advantage that it preserves many of the geometric properties of objects in the workspace including size, distance, and position. However, spatial overviews sacrifice much of the value of non-geometric object properties such as textual labels on collections in the space. More generally, the utility of the spatial overviews is fundamentally constrained by physical properties of the workspace including size, space population, and aspect ratio since these properties limit the level of detail that can be shown in the overview. In contrast, the structural overview has the opposite properties in that it preserves the non-geometric properties like text labels at the cost of size, distance, and position. Because structural overviews are not tied to geometry, their utility does not depend on the physical properties of the workspace.

Because of these tradeoffs, the third round prototype replaced the spatial overview with a structural or outline overview (see Figure 1). Since the overview was largely intended to provide access and readability of collection labels, a structural overview or outline view seemed to offer a more useful layout for text labels than a spatial overview. This is particularly true for nested hierarchical collections since a spatial overview could only show collections at a single level in the hierarchy while an outline view could show several levels of nesting at once.

Although these latest changes to the overview are likely to improve Niagara's usability for grouping tasks, they still do little to address the fundamental problem of limited screen space. As a result, a new technique was added to Niagara called automatic text reduction to create more meaningful representations of text objects under zooming to increase Niagara’s usable screen size. The automatic text reduction technique, shown in Figure 6, combines font size reduction and content reduction to decrease the space requirements of a text object while preserving the legibility of key words. In other words, when the user zooms out to see more objects at once, automatic text reduction tries to keep some of the text readable for the zoomed out objects. Because users have often explicitly incorporated the text objects into the workspace, they are usually familiar with the content of each object. As a result, the purpose of text reduction is primarily to allow users to recognize the text objects they incorporated into the workspace earlier. In addition to increasing usable screen space, automatic text reduction has the potential to assist users in abstraction. Using content reduction techniques such as eliminating common words may help users to more easily identify patterns such as rare, recurring key words or related concept terms.

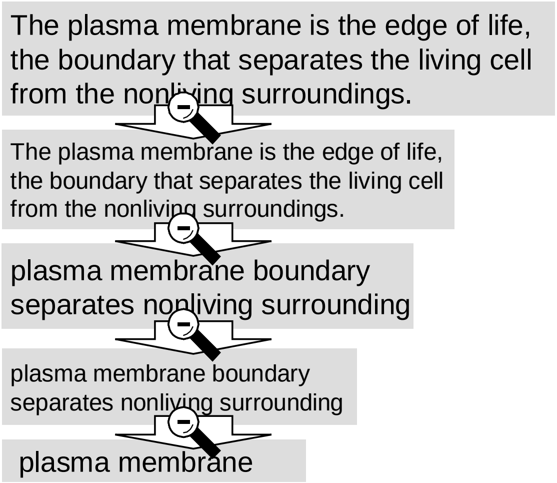
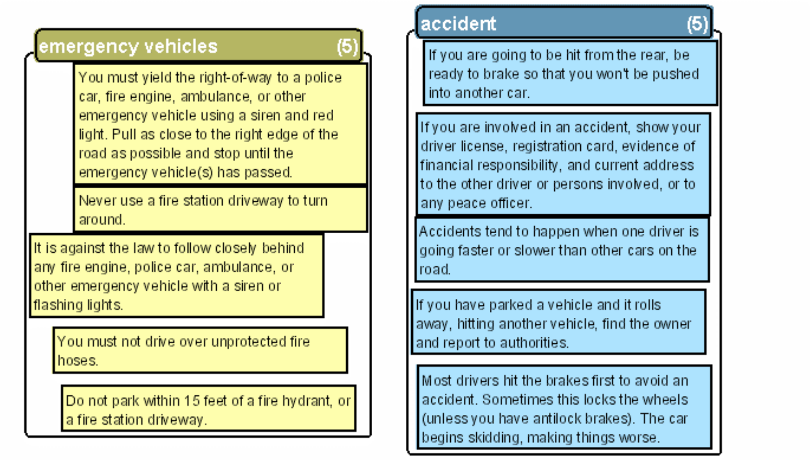

The plasma membrane is the edge of life, the boundary that separates the living cell from the nonliving surroundings.
The plasma membrane is the edge of life, the boundary that separates the living cell from the nonliving surroundings.
plasma membrane boundary separates nonliving surrounding
plasma membrane boundary separates nonliving surrounding
plasma membrane

Figure 6: An example of automatic text reduction applied to an object at several different zoom levels.

A second feature introduced in Niagara to make text objects more usable in the zoomable space is automatic grouping. Automatic grouping assigns objects to groups when the user moves multiple objects close to one another. The system then indicates the object's assignment to a group by automatically changing its background color to match the other objects in the group. The groups are just as quickly dismantled when the user the drags the objects out of the group. One of the primary advantages of these groups is that they have a consistent coloring that is easily recognizable when the view is zoomed out. In addition, the system also provides facilities for operating on the automatically created groups as a whole. The user can move the groups as a single object via a group handle. Likewise, an iconic representation of the groups appears in the structural overview of the space to allow for rapid editing of or navigation to the group. These groups can also be given a label to further identify them as the contents become more stable. Labeled automatic groups are shown in Figure 7.



You must yield the right-of-way to a police car, fire engine, ambulance, or other emergency vehicle using a siren and red light. Pull as close to the right edge of the road as possible and stop until the emergency vehicle(s) has passed.

Never use a fire station driveway to turn around.

It is against the law to follow closely behind any fire engine, police car, ambulance, or other emergency vehicle with a siren or flashing lights.

You must not drive over unprotected fire hoses.

Do not park within 15 feet of a fire hydrant, or a fire station driveway.

If you are going to be hit from the rear, be ready to brake so that you won't be pushed into another car.

If you are involved in an accident, show your driver license, registration card, evidence of financial responsibility, and current address to the other driver or persons involved, or to any peace officer.

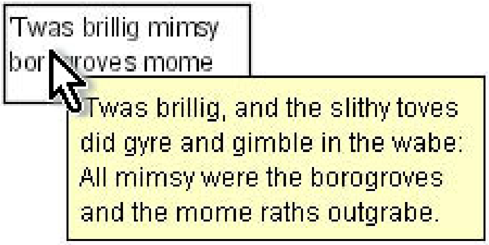
Accidents tend to happen when one driver is going faster or slower than other cars on the road.

If you have parked a vehicle and it rolls away, hitting another vehicle, find the owner and report to authorities.

Most drivers hit the brakes first to avoid an accident. Sometimes this locks the wheels (unless you have antilock brakes). The car begins skidding, making things worse.

Figure 7: A screenshot of automatically created groups in Niagara.

One final, more standard, feature added to improve the usability of zooming in this version of Niagara was full text tooltip popups, shown in Figure 8. These work just like standard WIMP tooltips by showing the full text of an object in a transient window when the mouse is positioned over a zoomed out text object. The primary difference these popup texts have from traditional tooltips is that they appear without requiring a pause when the user moves the mouse over an object in order to accelerate reading multiple items. The normal tooltip pause was eliminated in order to allow users to quickly access the full text of an object for fast comparisons between object. A similar feature is implemented in PhotoMesa [6], a zoomable photo browser, to show the photo currently under the cursor.



Twas brillig mimsy bor roves mome

Twas brillig, and the slithy toves
did gyre and gimble in the wabe:
All mimsy were the borogroves
and the mome raths outgrabe.

Figure 8: A screenshot of the full text tooltips provided in Niagara.

2.6 Studying a Simplified Task

There are a number of challenges in designing studies to evaluate the technology in an authoring system like Niagara. Perhaps the biggest challenge is that text-based authoring tasks take a long time to complete, primarily because of the large amount of information that needs to be read and understood. This means that experimental subjects can get tired, bored, or distracted in completing the tasks and therefore exhibit inconsistent behavior. Long task completion times also make experiments less viable both for participants and administrators. Likewise, these tasks involve many cognitive operations including reading, abstracting, comparing, searching, group-labeling, etc. As a result, there can be a lot of variability in the way subjects combine these different operations to complete the tasks. An additional problem is that many text-based organization tasks do not have a single correct answer. This makes it difficult to consistently evaluate many users' output from a particular task.

Because of these difficulties, a simpler task was needed that would retain the defining characteristics of the text-based organization task. In a text-based organization task, the organizer puts objects into groups based on common topics. The challenge in creating these groups is that each text object has many different topics at multiple levels of abstraction. As a result, each object can potentially fit into many different groups. The ultimate goal is to find the most strongly related groups of objects while respecting constraints such as limiting group size and balancing the number of objects between groups.

To simulate this text organization task, a simplified task was created that substitutes shapes for text objects. Similar to the task with text objects, the simplified task requires subjects to sort shapes into groups where objects in the groups are strongly related by visual properties. Because subjects would not be told the common properties for each group, the task would primarily require subjects to look for the strongly related groups. The task would start with a random arrangement of shapes and would be completed when the shapes had been sorted into a specified number of equal-size groups.

The particular instance of this task used in our experiments involves finding groups with two common properties. For example, the completed task might have one group containing circles with green backgrounds, another group containing shapes with blue backgrounds and pink borders, etc. An example of this task can be seen in Figure 9. Just as with the text objects, the shapes have a number of properties on which they can be compared including shape, background color, background texture, border color, border texture, and size. Each of these six properties has six values leading to 6^6 or 46,656 different possible shapes. The total number of different group membership criteria is (6 choose 2) x (6 choose 1) or 90 different criteria.

Using shapes instead of text reduces the amount of information involved in the task which in turn reduces task completion times. This simplification also reduces the number of cognitive operations needed to complete the tasks, therefore reducing variability. Further, because shapes can have well defined visual properties, they facilitate a quantitative measure of the quality of results.

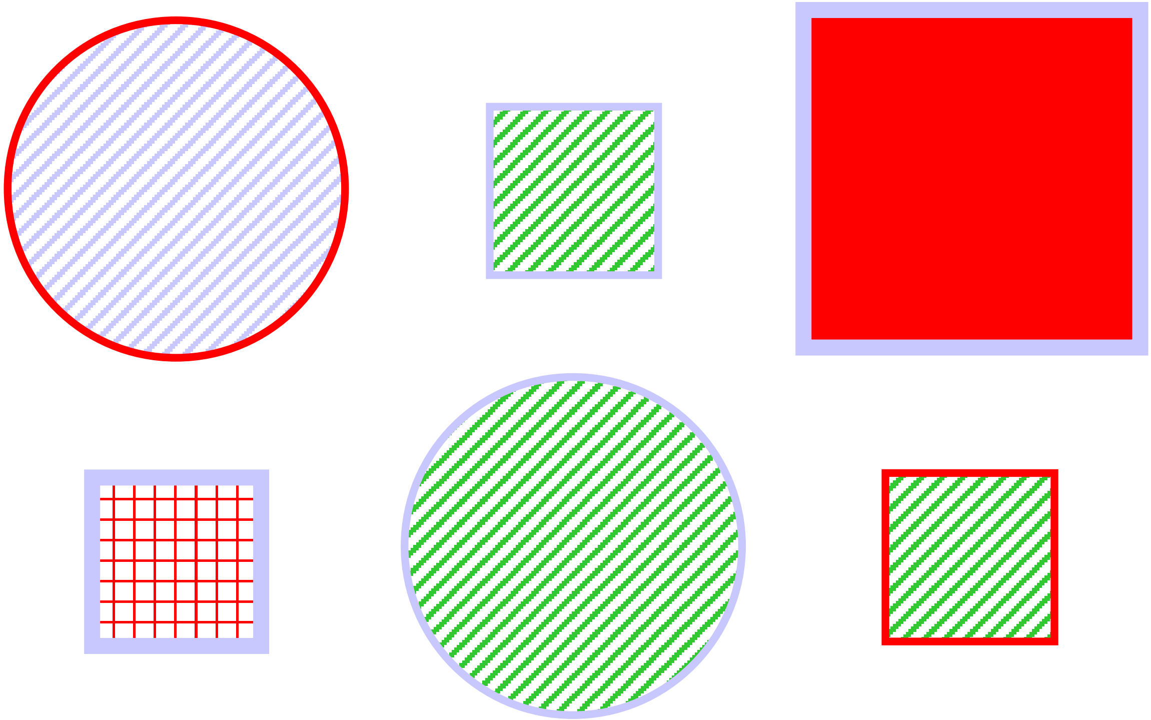


Figure 9: An example shape grouping task. One group has grouping criteria of squares with blue borders. The second group has grouping criteria of diagonal striped textures and thin borders.

Admittedly, there are differences between the shape task and text task. In particular, because it uses perceptual properties such as color, size, shape, and texture, this shape task introduces the issue of pre-attentive versus conscious processing. However, since task completion times were expected to be on the order of fifteen minutes for these tasks, the effects of the different kinds of processing were likely to be insignificant. The tasks also differ in their use of verbal versus visual working memory. This means that subjects may be able to hold different numbers of objects in memory in shape tasks versus the text tasks (for example, see discussion in [88]). However, the text tasks often have additional complexities, for example many non-topic words, that limit the number of text objects that can be held in memory at once. Most importantly, it is expected that any difference in cognitive processing would affect all tools equally. This would likely mean a difference in the scale of the effects found between different kinds of data.

From previous experience, the hypothesis was developed that ZUIs would improve performance over other available interface paradigms for organization tasks. The reasoning behind this hypothesis is discussed in more detail below. Based on this hypothesis, the study below was designed based on the shape task to compare a ZUI to a folder-based interface. Only after a positive difference was found for the simplified shape task would the study be repeated for the original text task. It also anticipated that in studying this simplified version of the task, a better understanding of the text task could be developed.

2.6.1 The Core Question: Zooming versus Folder Interfaces

As previously mentioned, computer users often organize information using spatial arrangements. Yet most computer displays are only large enough to show the equivalent of one or two pages of information at normal size. Since this is a significant limitation relative to physical surfaces such as tables or whiteboards, computer interfaces to support organization usually implement mechanisms to increase the size of the available virtual space.

One type of interface used to increase the virtual space on a computer display is folders. Folders are an example of a more general technique that is analogous to "paging" in a physical book. Paging provides access to a large workspace that is divided into two-dimensional pages. On a computer, these pages are typically accessed by the user via an overview such as a hierarchy of labeled folders. Examples of tools that use such a paging mechanism are electronic books, presentation slide authoring tools, and file system explorers.

Another approach is to have one big space with a navigation mechanism. Perhaps the simplest navigation mechanism is scrollbars. Scrollbars are ideal for long one-dimensional workspaces such as paginated text documents. For these spaces, a scrollbar presents an overview of the entire space and facilitates rapid systematic navigation of all content. Scrollbars become much less effective when extended to two dimensions. Here the horizontal and vertical dimensions are each assigned their own independent scrollbars. As a result, the two individual scrollbars do not represent an overview of the entire workspace; instead they represent linear strips of the workspace. This means that to perform a systematic navigation of the entire workspace, the author must coordinate movements of both scrollbars. Beyond the cognitive complexity of managing the coordination, the independent scrollbars introduce additional pointing tasks to acquire and operate both scrollbars.

A second example that combines a large workspace with a navigation mechanism is ZUIs. In a ZUI, different views of the workspace are obtained by changing the position and magnification in the space through mouse or keyboard operations. These types of interfaces are often used in such settings as map-based systems, image editors, and games.

A third type of interface used to increase available space is distortion-based or fisheye techniques. For spatial arrangement tasks, distortion is often not desirable as it is likely to interfere with users' perception of spatial relationships. As result, distortion-based approaches were not explored here.

The two specific interfaces compared here are a folder-based interface and a ZUI-based interface. An informal analysis of users' presentation organization behavior is presented next to highlight the difficulties a user encounters in organizing information using folders. This analysis was suggested by the work of Henderson, et al. [51], who analyzed user behavior in a windowed computer environment using the language of working sets from the virtual memory literature.

The assumptions of most virtual memory strategies are that programs make use of limited size working sets and have locality of reference [51]. This basically means that only a limited amount of information is needed in memory at once. When these assumptions are violated, memory management strategies (not coincidently called paging) lead to thrashing where a large percentage of time is spent managing memory allocation rather than doing actual work.

Humans also manage memory when organizing information for a presentation. One of the primary reasons for using a workspace for authoring presentation content is because it functions as a large external memory. Yet the assumptions of virtual memory strategies described above do not necessarily hold for this task. Indeed, rather than locality of reference, the task often requires the user to access objects throughout the external workspace. Non-local access is needed to support making comparisons between objects in different parts of the workspace. These global comparisons allow a person to find similarities between objects in different groups and mentally experiment with alternate organizations.

Consequently, a user's ideal external memory for this task is a large two-dimensional workspace, such as whiteboards or tabletops. The large workspace reduces the cost of accessing items in the external memory since the entire workspace is visible through fast eye and head movements. This allows the author to quickly compare objects both locally within groups and globally between groups.

A folder-based interface seems to be less suited as an external memory for this type of task. Authors working with folders tend to associate one folder with one semantic group. Instead of comparing items between groups with a fast visual scan as they would in a large workspace, users of folders must initiate a page change, reorient to the new page, and then make a visual scan. As a result, it is expected that the time needed to compare items between groups would be substantially increased with folders over a single large workspace. A countervailing factor is that users have previous experience with folder-based interfaces for email and file systems. This raises the issue of whether prior experience will lead to superior performance with folders and mask out the expected benefits of a zooming interface. To resolve these questions, an empirical evaluation is presented below for the hypothesis that a ZUI-based interface will have shorter completion times than a folder-based interface for a shape organization task.

2.6.1.1 Method

2.6.1.1.1 Participants

Fourteen regular computer users participated in the study. Five of the subjects were female and nine were male. Subjects ranged in age from 18 to 50 and were given a gift certificate for their participation in the study.

2.6.1.1.2 Equipment

The study was run on Pentium 3 and Pentium 4 machines (ranging from 700 MHz to 2GHz) with at least 256 MB of memory and 1600x1024 flat panel displays.

2.6.1.1.3 Procedure

For our within-subjects study, subjects were asked to complete two shape grouping tasks using the two interface conditions of zooming and folders. The primary interface metaphor for grouping the shapes in both interface conditions was simple drag and drop of shapes.

The folder interface condition mimicked a file browser with a folder hierarchy in a resizable pane on the left side of the screen and a freeform 2D workspace in a pane on the right side of the screen. Each folder in the hierarchy provided access to its own (conceptually) infinite non-zoomable 2D space that was made visible in the right hand pane when the folder was selected. This 2D space allowed at least 10 shapes to be visible at once and provided scrollbars when shapes in the workspace extended outside the current view. The interface also provided facilities to create, name, rearrange, and delete folders. When objects were moved between folders, the tool automatically positioned objects as close to the center of the folder's space as possible without overlapping other objects. The folder interface is shown on the left in Figure 10.

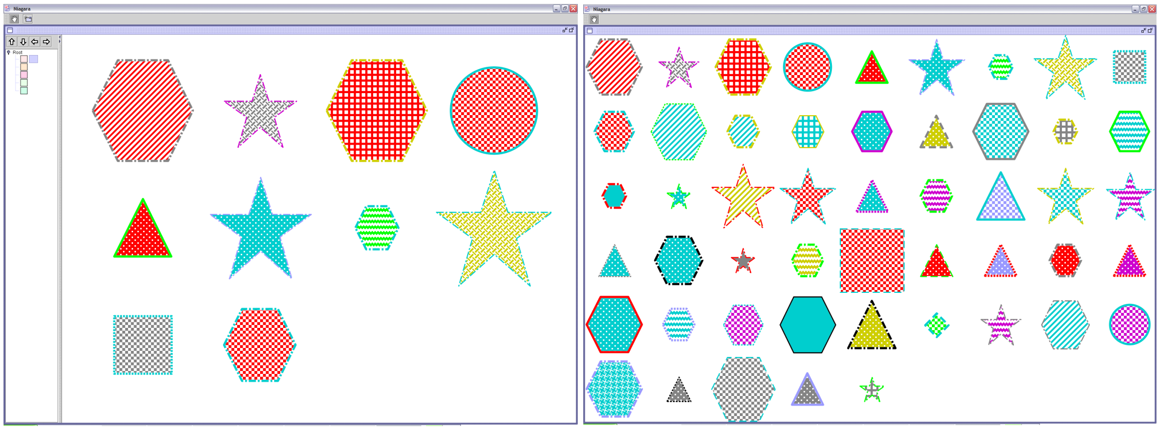


Figure 10: The starting state for the two interface conditions in the shape study. The folder interface is on the left and the ZUI interface is on the right.

The zoomable interface condition provided a freeform 2D workspace as in the previous condition, but instead of a folder hierarchy, it provided mouse interactions to zoom in and out in the workspace. Here again, scrollbars were provided when shapes extended outside the current view, for example when the view was zoomed in. The zoomable interface is shown on the left in Figure 10. The shapes in this zooming condition also implemented a type of semantic zooming (semantic zooming is summarized in [85]) that tried to visually preserve shape properties. So while the absolute size of the shapes was affected by zooming, the other properties, including a shape's size relative to other shapes, were largely unaffected. One primary exception was that background color and pattern sometimes became difficult to see as the view was zoomed out and border widths increased relative to the size of the objects. Similarly, as the view was zoomed out the relative differences in size between shapes became less apparent. This type of problem required participants to zoom in to resolve any ambiguities.

In both interface conditions, the system implemented a simple form of "bumping" to prevent objects from occluding one another. Because this kind of bumping is difficult to implement for arbitrary shapes, the system implemented an approximation of the optimal bumping behavior. Although there were cases where this algorithm defaulted to bumping with the bounding boxes, in practice it gave reasonable results for the most common cases. While occlusion is not particularly disruptive for shapes, it is likely to be more disruptive for more abstract object types such as text, web thumbnails, or pen input [93]. In addition, bumping effectively increased the demand for space, which is likely to highlight differences between the two interfaces. A similar effect could have been achieved by increasing the number of shapes in the tasks, although this would have also increased completion times.

The study task involved grouping shapes based on common properties. The shapes had six properties (see above) that were described to the subjects prior to completing the tasks. For each task, subjects were asked to identify groups of 10 shapes that had two of the six properties in common. These two properties were not specified to the subjects ahead of time as a primary component of the task was intended to be the discovery of the group membership criteria. A sample task is shown in Figure 9. Subjects were informed that a unique set of property pairs divided the objects into groups of 10 but that individual objects may fit into more than one group.

There were several considerations in choosing the size of the shapes used in the study. Since the study was ultimately simulating tasks involving text objects, the shapes were chosen to approximate the size of a typical text object. In Niagara, a fairly typical text object with 25 words at a 12 point SansSerif font with an average word length of 5 letters has a bounding box of 200 x 90 pixels. However, the shape task had an additional constraint since size was one of the properties that varied between shapes. As a result it was necessary for subjects to be able to clearly differentiate between differently sized shapes when zoomed in. This required particular attention for comparing the sizes of, for instance, diamonds and squares. Informal evaluation of different variations in shape sizes led to a difference of 40 pixels between sizes. A minimum size constraint of approximately 125 x125 pixels was enforced for the shapes so that subjects would be able to clearly identify the background textures for all the different background texture-border type-shape combinations when zoomed in. The combination of these different constraints led to an average bounding box of 205 x 205 pixels for the shapes in the study. It should be noted that although this results in a larger bounding area than with the text objects, several of the shapes (i.e. star, hexagon, triangle, circle, and diamond) covered only a fraction of the area in their bounding box. Ultimately, it is expected that the average shape area is comparable to a typical text object's area.

For each condition, subjects were first given a tutorial on how to use the interface. The task was then explained and the six shape properties were described. Subjects were told to indicate grouping through spatial proximity in the zooming condition and with folders in the folder condition. The experimenter demonstrated a sample task using the current interface, and the subjects were then given their own practice grouping task to complete. Finally, the subjects were given the actual timed task and were told to complete the task as fast as they were able.

At the beginning of each task, the shapes were distributed in a random order. For the zooming condition, the shapes were distributed in a single large grid. In the folder condition, the shapes were distributed in folders such that each folder initially contained 10 objects. The practice tasks were composed of 30 objects requiring 3 groups of 10 and the actual tasks were composed of 50 objects or 5 groups of 10. The ordering of the interface condition, practice task, and actual task were all counterbalanced to reduce ordering effects.

There was a tradeoff in the design of the starting condition for the folders task. Instead of distributing the shapes into the 5 folders, shapes could have been put into a single vertically-scrollable root folder to start with. This would have the advantage that all objects would be in a single conceptual space to start with. However, because the screen could show approximately 10 shapes at a time, the subject would have to scroll through approximately 5 screens worth of shapes at the beginning of the task. It was ultimately decided that the 5 folder icons provided faster and more consistent access to the shapes. Subjects could quickly see all 50 objects with 5 clicks on the folder icons. Moreover, they could directly jump to a consistent set of 10 objects with a single click.

It should also be noted that folders did not have to be used with the folder-based interface. Although subjects were asked to put their final group selections into individual folders, they were free to move all the objects into a single folder during the task, using the scrollbars instead of folder icons to navigate. Because the tool automatically positioned objects as close to the center as possible, subjects could quickly move the objects into a single folder and get a well-packed space in which to scroll around. The time needed to move all the objects into a single folder was nearly insignificant relative to the overall task time. In the experiment, one subject was observed to use this strategy.

The software logged all relevant operations, though this information was ultimately not analyzed. Following the two tasks, subjects were given a questionnaire regarding their experiences with the two conditions. The questionnaire is available in Appendix A.

2.6.1.2 Results

Paired two sample t-tests were used to analyze the results from the study. These were chosen over ANOVAs because only one independent variable and two samples were being compared.

One subject's results were discarded because of several significant cell phone interruptions during the timed tasks. The means for the dependent variables are shown in Table 4 for the remaining 13 subjects. A paired two sample t-test was performed for task completion times. This data indicated a statistically significant effect of interface type p < 0.03 with a 30% faster completion time for zooming. The mean completion times, shown in Figure 11, were 17.5 (σ 8.0) minutes for folders and 12.2 (σ 7.0) minutes for zooming.

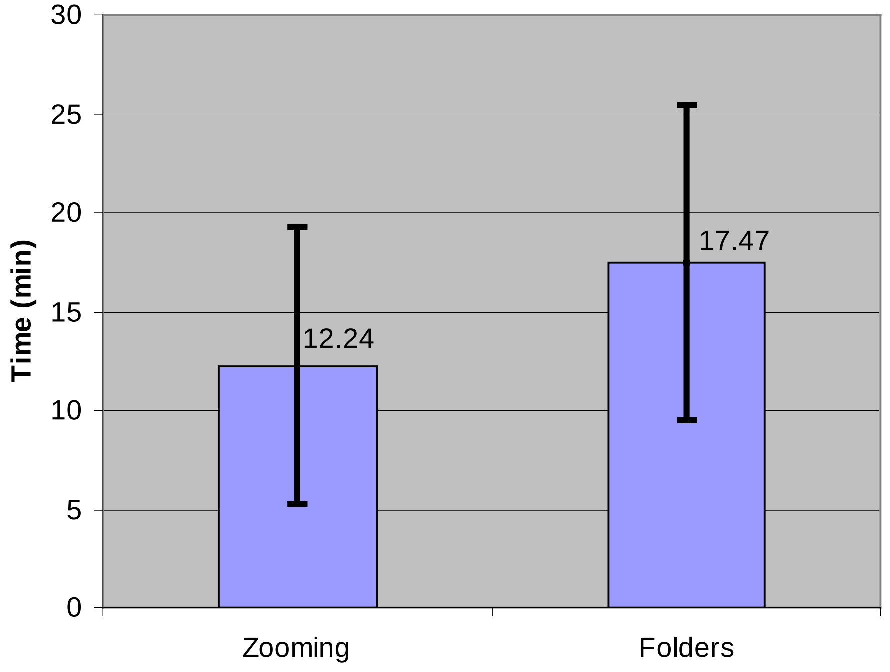


Figure 11: A graph comparing the mean completion times for zooming versus folders. The error bars indicate standard deviation.

There was also a statistically significant effect of interface type on two questionnaire items. For the question "What is your overall reaction to using this tool for the task?" (1=Frustrating,5=Satisfying) the means were 4.2 (σ 0.6) for zooming and 2.8 (σ 1.0) for folders with p < 0.001. For the question "How often did you feel like the software interrupted your thinking?" (1=Rarely,5=Often) the means were 1.8 (σ 0.9) for zooming and 3.4 (σ 1.3) for folders with $p < 0.001$.

Table 4: The means and standard deviations of dependent variables measured in the study.

	Zooming	**Folders**
Completion Time (min.)	12.2 (σ 7.0)	17.5 (σ 8.0)
Satisfaction (1=Frustrating, 5=Satisfying)	4.2 (σ 0.6)	2.8 (σ 1.0)
Interrupted (1=Rarely, 5=Often)	1.8 (σ 0.9)	3.4 (σ 1.3)

2.6.1.3 Discussion

The results indicate that a zoomable interface has a considerable advantage over a folder interface for the shape grouping task. A number of factors likely contribute to this advantage but the most notable seems to be the time involved in making comparisons between shapes. In the zooming condition, comparisons were usually made with rapid eye and head movements that took a few hundred milliseconds. In contrast, the folder condition often required subjects to look at objects in multiple folders. Moving between folders introduced a mouse pointing operation increased the overhead of many comparisons to more than a second. Because of the large number of comparisons involved, this relatively small difference in time became significant over the complete task.

A second possible contributing factor was the formality of folders versus spatial arrangement. Using the folder interface, subjects would often mark a folder with a descriptive name or unique position to indicate that the contained group was completed. Informal feedback from the study indicated that this behavior provided subjects with a sense of orderliness and progress. However, subjects were also observed to be hesitant to revisit groups that had been marked as complete. As a result, this behavior often delayed their completing the task when these groups contained an error. This problem was less frequently observed with spatial arrangement in the zooming condition, suggesting that the formal separation of groups in the folder condition is harmful for maintaining awareness of groups once the user considers them complete.

2.6.2 Scaling Effects and Other Parameters of Task Difficulty

The previous study demonstrated the benefits of ZUIs for novice users performing an organization task with a modest number of shapes and fixed organization parameters. It does not address how folders and ZUIs differ for more experienced users, how user performance changes as the number of objects to be organized increases, or what parameters most affect the task's difficulty. Performing additional controlled studies to examine each of these issues would be extremely time-consuming. As a result, what are presented in this section instead are multiple aggregated trials performed by a single expert user. Although these results must be regarded cautiously because of the small sample size, they can indicate potential trends and possible areas for further research.

Of course, understanding the difficulty of shape tasks is not the ultimate goal here, rather these results are intended to be suggestive for text based organization tasks. The implications of these results for text organization will be discussed in the next section following a controlled study of text organization.

There are many parameters that can influence task difficulty for the shape organization task. The task parameters explored here include:

1) Number of Objects – this determines the overall scale of the task. In addition, the number of objects is likely to interact with the kind of tool being used to perform the task. In particular, task completion times with folders seem likely to degrade rapidly since the percentage of objects that can be viewed at once in a folder becomes smaller and smaller as the number of objects increases.

2) Number of groups – given the number of objects in the entire task, the number of groups determines the number of objects in each group. The task described in the study had 5 groups; hence there were 10 objects per group.

3) Group membership criteria - the group membership criteria are the number of object properties that are used to define group membership. The task described in the study had two membership criteria, for example, circles with red backgrounds or objects with blue backgrounds and green borders. Increasing the number of membership criteria to three would lead to groups such as circles with red backgrounds and blue borders or squares with checkered backgrounds and thin-dotted borders.

4) Overlapping membership criteria - the number of overlapping group membership criteria indicates how many groups have the same values as membership criteria. For example, there is one overlapping set of membership criteria when one group has circles with red background and another group has circles with green borders. This parameter was not explicitly controlled for the tasks in the previous study. However, the two tasks that were actually used in the study were similar in this regard.

5) Total object attributes – the total number of object attributes is the number of attributes that vary in a set of objects. For instance, if a set of objects varies on the 3 attributes of shape, background color, and border color, then all the objects in the group would be identical except for these three attributes. The task in the previous study had 6 total attributes.

6) Possible values per attribute – the possible values per attribute is the number of different assignments there are for a particular attribute. For instance, if a color attribute has values of red, blue, and green, then there are 3 values for that attribute. In the previous shape task, the attributes each had 6 possible values.

As mentioned, an experienced shape organizer (specifically, the author) performed a suite of shape organization tasks that varied each of the above parameters one at a time. The shapes used in these tasks were randomly generated by a computer program within the constraints of the different parameter settings. Since the program randomly assigned the values for the different shape attributes, the organizer was not aware of the final answers prior to completing the tasks.

The organizer performed six trials for each combination of parameter settings. One exception was that only three trials were run for each of the tasks involving 120 objects since these trials often took more than 25 minutes to complete. The points on the graphs below indicate the means of the 6 trials and error bars indicate the standard deviations for these trials. Note that some trial sets were reused between comparisons as they had the same parameter settings.

The first set of trials shown in Figure 12 varied the number of objects to be organized. Shapes had to be put in to groups of 10 for each of these trials. The total number of attributes, number of membership criteria, and possible values per attribute were held constant for each of these trials at 6, 2, and 6 respectively. Overlapping between membership criteria was not controlled. The trials were performed for both zooming and folders.

The data in Figure 12 indicates that the completion times are increasing at a more rapid pace in the folder condition than in the zooming condition. Fitting a power regression equation to the data for the folder condition yields $y=0.0006x^{2.312}$ with $R^2 = 0.987$. Likewise for the data in the zooming condition, fitting a power regression equation to the data yields $y=0.0006x^{2.216}$ with $R^2 = 0.983$. In contrast, fitting an exponential regression equation to the folder and zooming data yields $y=0.712e^{0.035x}$ with $R^2 = 0.977$ and $y=0.518e^{0.0329x}$ with $R^2 = 0.934$ respectively. More data would be needed to accurately predict whether the completion times are increasing at an exponential or a polynomial rate. Regardless of the underlying functions, the existing data suggest that zooming is going to outperform folders for any number of objects.

Overall, the rapid increases in completion time are predicted by several inherent challenges in the task as the number of objects increases. First, the potential for unintended, random groups increases with the number of objects. These unintended groups lead to false starts and dead ends. For example, when a group of objects fits a set of unintentional membership criteria, putting them into a group is likely to prevent the organizer from completing 1 or more other groups. This means that the organizer must perform a substantial reorganization before the task can be completed.

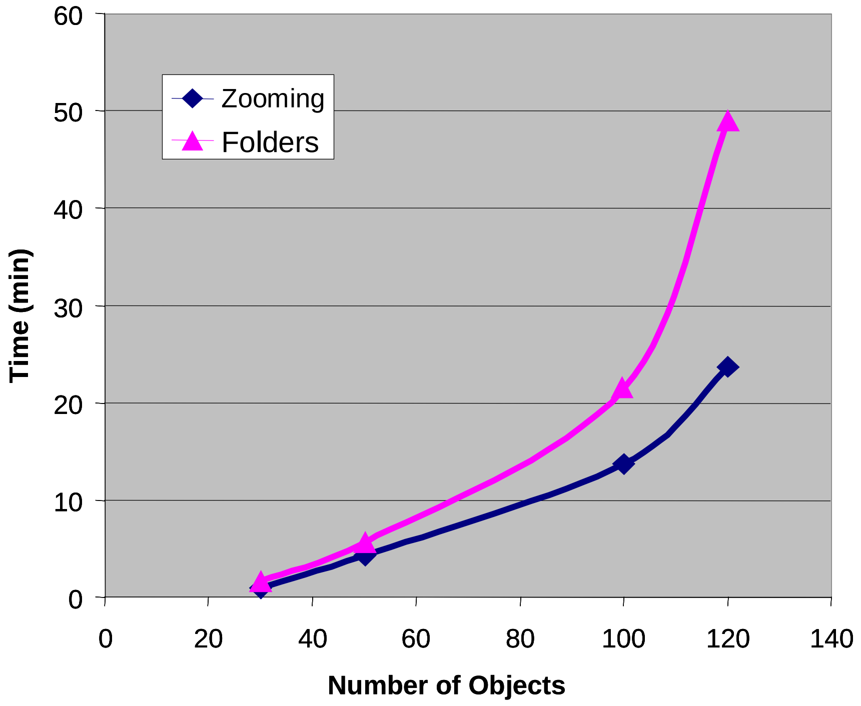


Figure 12: Graph of trials that varied in number of objects.

Similarly, increasing the number of objects also increases the chances that objects will unintentionally fit into more than one group. For example, one group might have membership criteria A and B, while a second group has membership criteria C and D. As the number of objects increases, the probability that there will be objects matching all four of these criteria also increases. When these ambiguities occur, the organizer will often end up with too many objects in one group and not enough in another group. In this case, the organizer must revisit groups and reevaluate membership criteria in order to find the ambiguous objects.

Another important factor that affects task completion times is the ratio of group size to overall number of objects. This ratio decreases as the group size is held constant (as it was here) and the number of objects increases. As the ratio decreases, the similarity of the group members is less prominent as the number of objects outside the group with overlapping properties increases.

Figure 13 shows the second set of trials that varied in the number of groups. The trials consisted of 50 shapes and all trials were performed under the zooming condition. The trials used 2 group membership criteria, 6 total attributes, and 6 values per attribute. Overlapping between membership criteria was not controlled for these trials.

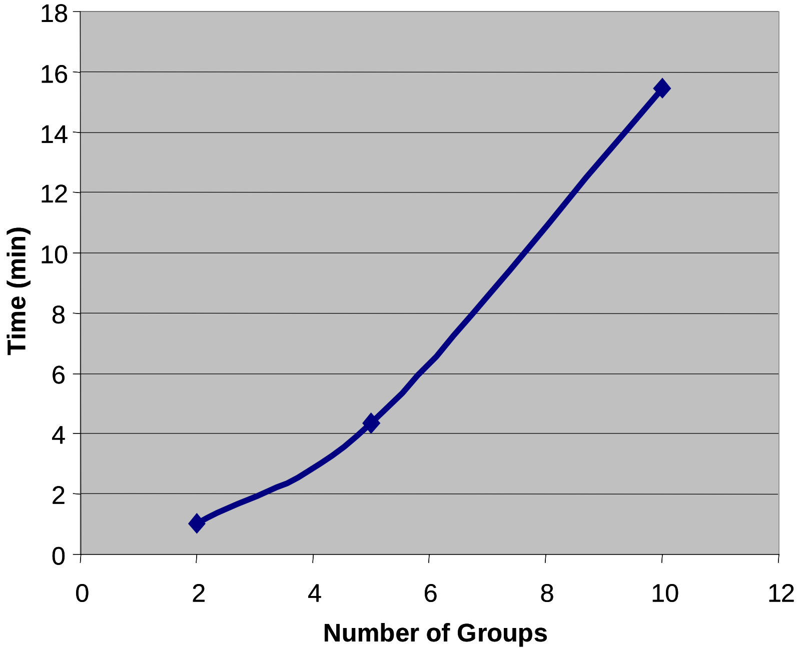


Figure 13: Graph of trials that varied in the number of groups.

The graph indicates that completion times are proportional to the number of groups. This is partially because as the number of groups increases, the group size decreases which increases the probability that a set of objects will unintentionally have the necessary number of properties in common. Similarly, as the number of groups increases, the probability increases that objects will unintentionally fit into more than one group. In fact, the number of group-to-group relationships is increasing quadratically with the number of groups that suggests a potential quadratic increase in completion times.

The third set of trials shown in Figure 14 varied in the number of group membership criteria necessary for each group. The trials consisted of 50 shapes which still had to be put into groups of 10 for each of these trials. The total number of attributes and possible values per attribute were held constant again for each of these trials at 6 and 6 respectively. Overlapping between membership criteria was not controlled. The trials were all performed under the zooming condition.

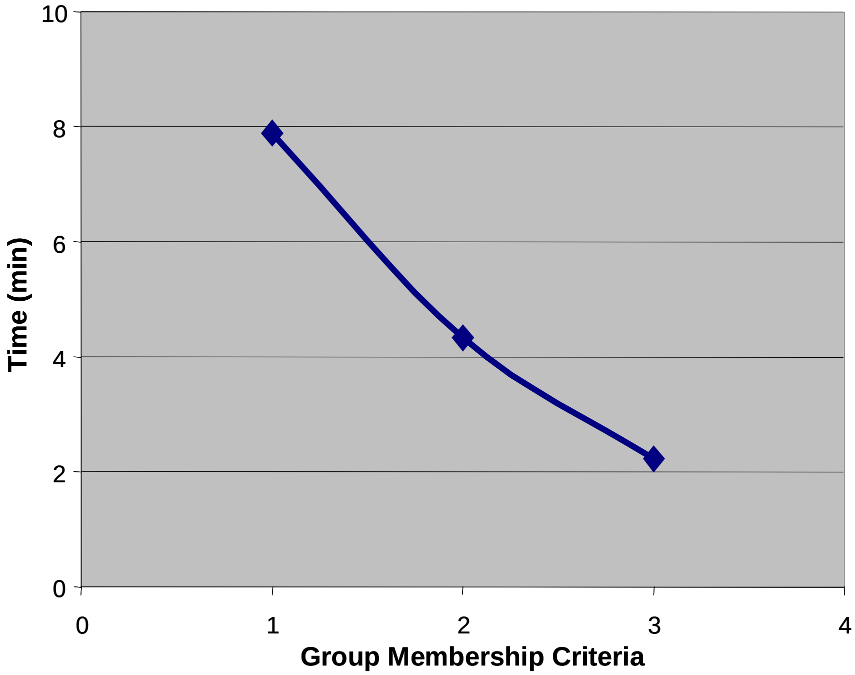


Figure 14: Graph of trials that varied in the number of group membership criteria.

The data in Figure 14 indicates that completion times are inversely proportional to the number of group membership criteria. This is because as the number of criteria increased, the probability that objects would unintentionally have all the criteria in common decreased. As a result, fewer similar objects were needed to identify the grouping criteria for a particular group since the number of similarities was greater. Trials were stopped beyond 3 criteria because the completion times had reached the minimum time required to move the objects into groups.

The fourth set of trials shown in Figure 15 varied in the number of overlapping membership criteria. For these trials, there were 50 shapes, the shapes needed to be put into groups of 10, the total number of attributes was held at 6, the number of membership criteria was held at 2 per group, and the number of possible values per attribute was held at 6. These trials were all performed under the zooming condition.

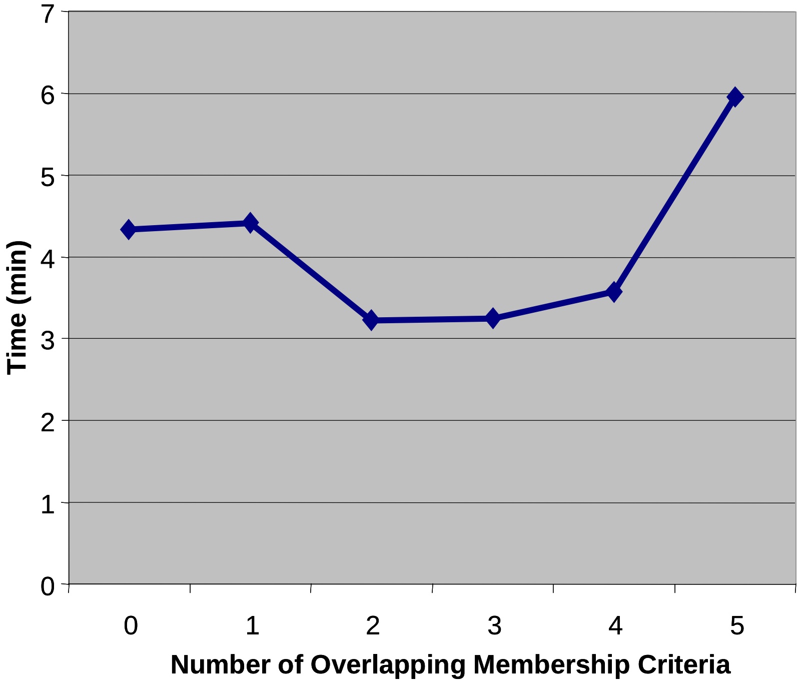


Figure 15: Graph of trials that varied in the number of overlapping membership criteria.

The data in Figure 15 suggests that the relationship between completion times and overlapping membership criteria is somewhat complex. The overlapping membership criteria seemed to interact with one of the initial challenges in the shape task, namely identifying possible group criteria. When the number of overlapping criteria reached 2, the search for promising group criteria became easier since several groups of objects shared a single criterion. However, this identification problem only occurs at the beginning of the task since as the task goes on, there are fewer objects in which to look for patterns. As a result, the advantage with 2 overlapping membership criteria was overwhelmed by the problems introduced by the overlapping groups. Specifically, having overlapping groups meant that many objects fit into multiple groups. These kinds of ambiguities required additional time to resolve.

Figure 16 shows the data from the fifth set of trials that varied the total number of shape attributes. These trials had 50 shapes, 10 shapes per group, 2 membership criteria, and 6 values per attribute. Overlapping between membership criteria was not controlled. These trials were all performed under the zooming condition.

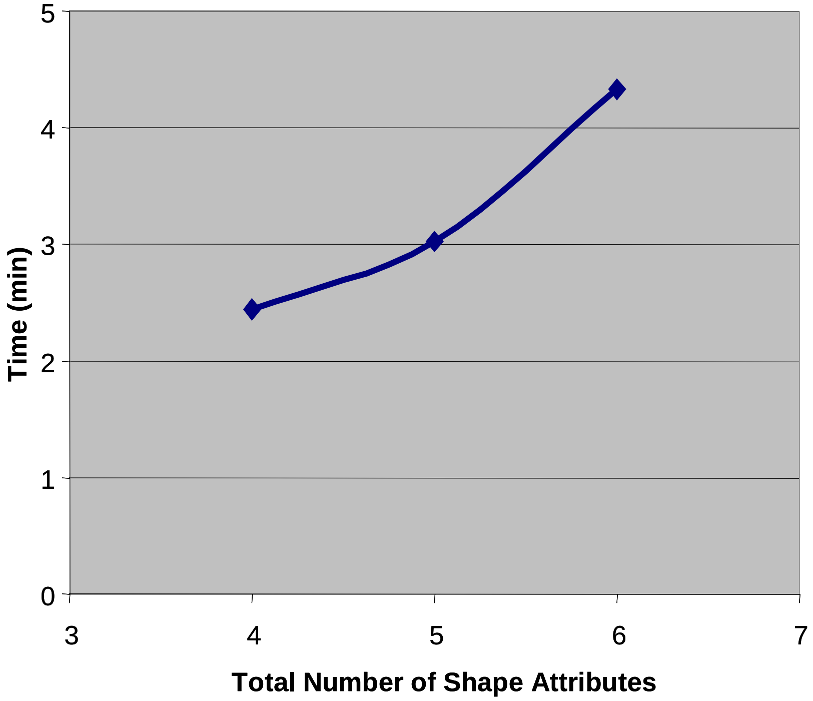


Figure 16: Graph of trials that varied in the total number of shape attributes.

The graph in Figure 16 indicates that completion times increase in proportion to the total number of attributes. This is perhaps the most intuitive of the results since the conceptual search space increases as the total number of attributes increases. Restated, as the conceptual space expands but the number of similarity group criteria stays the same, the relative similarity of the objects decreases. This ultimately makes finding the membership criteria more difficult for the organizer.

Figure 17 shows the final set of trials that varied in the number of possible values per attribute. These trials had 50 shapes, 10 shapes per group, 2 membership criteria, and 6 toal attributes. Overlapping between membership criteria was not controlled. These trials were all performed under the zooming condition.

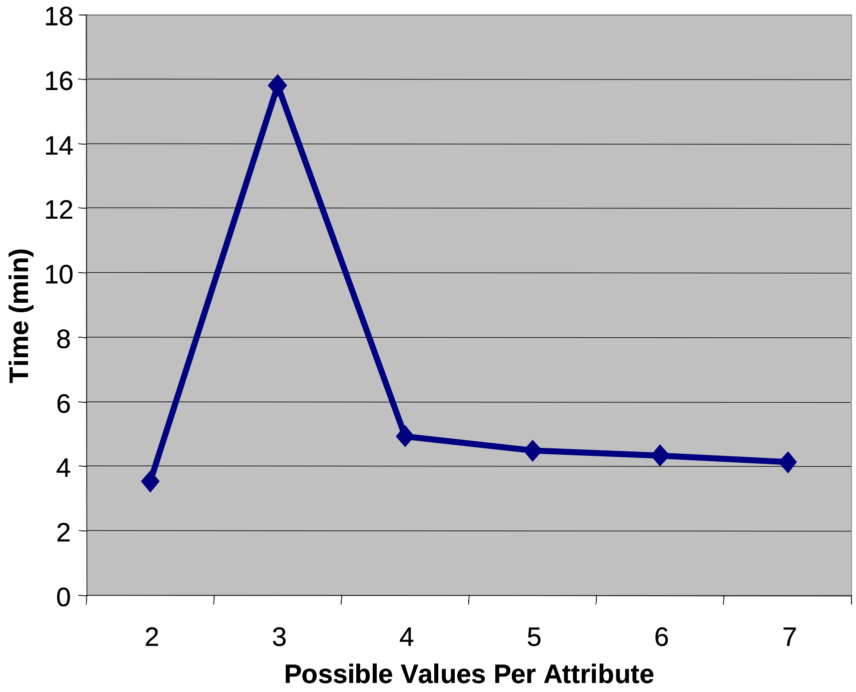


Figure 17: Graph of trials that varied in the number of possible values per attribute.

The graph in Figure 17 indicates the complex relationship between the values per attribute and completion times. When there are only 2 values per attribute, there are many unintentional groups. In fact, there are so many unintentional groups that it becomes possible to complete the task without finding the intended groups. With 3 values per attribute, the task becomes much more difficult. Although there are still many unintentional groups, it is typically no longer possible to complete the task unless the intended groups are found. As a result, the organizer creates many groups that they must later abandon when they prevent the task from being completed. At 4 attributes, the graph becomes more regular. Here, increasing the number of values decreases the probability that an object will be randomly assigned one of the values being used as group membership criteria. In an extreme example, if there are 1 million values for a particular attribute and one of the values is used as a group's membership criteria, then there is an extremely low probability that an object that is not a member of the group will be randomly assigned the value used by the group.

2.6.2.1 Discussion

Ultimately, each of the parameters explored above is affecting the similarity of group members relative to the other objects outside the group. As a result, the data seems to suggest that several of these parameters are not independent. For example, in Figure 12 the number of objects to be grouped is also increasing the number of groups since the number of objects per group is held constant. Comparing completion times between the group-size data in Figure 13 and the number of objects data in Figure 12 indicates similar results when the number of groups is the same. Specifically, completion times with 100 objects and 10 groups in Figure 12 averaged 13.69 min whereas completion times with

50 objects and 10 groups in Figure 13 averaged 15.47 min. This suggests that the number of groups may be a better predictor of task difficulty than the raw number of objects.

Similarly, increasing the total number of attributes in Figure 16 also increases the ratio of total attributes to the number of group membership criteria. Comparing completion times between the total attributes data in Figure 16 and the number of group membership criteria data in Figure 14 indicates similar results when the ratio of the number of membership criteria to total attributes is the same. In particular, completion times with 3 membership criteria and 6 total attributes in Figure 14 averaged 2.22 min and completion times with 4 total attributes and 2 membership criteria in Figure 16 averaged 2.45 min.

Taking into account these dependencies, the results suggest the following general relationships between task parameters and task difficulty:

1) Difficulty **increases** with the number of groups
2) Difficulty **increases** with the ratio of the total number of attributes to the number of membership criteria
3) Difficulty **decreases** with the number of values per attribute
4) Difficulty **increases** with the number of overlapping membership criteria

Following the study in the next section is a discussion of the extent to which these relationships apply to text-based organization tasks.

2.7 Extending Simplified Task Results to Text

Although the shape task simulates the text grouping task in many ways, it also differs in several important aspects. Consequently, the results in the previous section require further validation in order to be applied to text. The most significant interface difference between the two tasks is that shapes retain nearly all their value when they are viewed zoomed out. In contrast, text objects quickly become unreadable when they are viewed zoomed out. Automatic text reduction is an attempt to improve the representations of arbitrary text objects under zooming. Still, this technique is not going to preserve all of the value of a text object under zooming since it removes some content from the objects, sometimes missing the complications and subtly of human language.

The shape task further differs from the text grouping task in that text objects do not have well defined properties whose values can be compared. As a result, part of the task in grouping text objects is identifying the dimensions along which objects can be compared. This identification is further complicated because text objects have differently weighted topics. Topics are also hierarchical meaning that two different topics can be related if they share a common abstracted parent topic. In contrast, the dimensions for comparison in the shape task were un-weighted, well-defined, and provided to subjects at the task's outset.

Our previous studies have suggested many difficulties involved in studying a text based organization task in a controlled setting including: long completion times, large variability due to the task’s cognitive difficulty, and a lack of quantitative measures of results. In previous studies, it was also observed that the variability between subjects was further aggravated because the tasks did not have clear stopping conditions. Quality was often mediated by the amount of time subjects spent on the task.

In this section, a study of text organization is described that tries to replicate the results found in the previous shape study. It is expected that there are not solutions to many of the study-related problems described above since these problems are inherent to text-based tasks. However, the study task does provide clear stopping conditions by requiring subjects to create a specific number of fixed size groups. In addition, the familiar task involving the driver's handbook was chosen to reduce completion times.

2.7.1 Method

2.7.1.1 Participants

Fourteen regular computer users participated in the study. Five of the subjects were female and nine were male. Subjects ranged in age from 23 to 62 and were given snack food for their participation in the study. Two of the subjects in text study had also participated in the shape study.

2.7.1.2 Equipment

The study was run on Pentium 3 and Pentium 4 machines (ranging from 1.2 GHz to 2GHz) with at least 256 MB of memory and 1600x1024 flat panel displays. The mice on these machines were also equipped with a scrollwheel.

2.7.1.3 Procedure

As in the shape study, subjects were asked to complete two organization tasks using the two interface conditions of zooming and folders. All subjects used both interface conditions so the study again had a within-subjects design. The primary interface metaphor for grouping the objects in both interface conditions was simple drag and drop of shapes.

The interface conditions used in the study were primarily the same as the shape study. The folder interface condition mimicked a file browser with a folder hierarchy in a resizable pane on the left side of the screen and a freeform 2D workspace in a pane on the right side of the screen. Each folder in the hierarchy provided access to its own (conceptually) infinite non-zoomable 2D space that was made visible in the right hand pane when the folder was selected. This 2D space allowed at least 20 text objects to be visible at once and provided scrollbars when text in the workspace extended outside the current view. The mouse scrollwheel could be used to vertically scroll the workspace. The interface also provided facilities to create, name, rearrange, and delete folders. When objects were moved between folders, the tool automatically positioned objects as close to the center of the folder's space as possible without overlapping other objects. The starting state for the folder condition in the study is shown on the left in Figure 18.

The zoomable interface condition provided a freeform 2D workspace as in the previous condition, but instead of a folder hierarchy, it provided the ability to zoom in and out in the workspace using the mouse scrollwheel. Here again, scrollbars were provided when text extended outside the current view, for example when the view was zoomed in. As previously mentioned, the text in this zooming implemented a form of automatic text reduction. So while the bounding box of the text was affected by zooming, some of the text was kept readable. Automatic grouping, described above, was also provided in the zooming condition. This provided a means to delineate groups, move groups as a whole, and give groups a label. Lastly, the zooming condition also implemented tooltips, described above, that provided access to the full text of an object when the view was zoomed out. The starting state for the zoomable condition in the study is shown on the right in Figure 18.

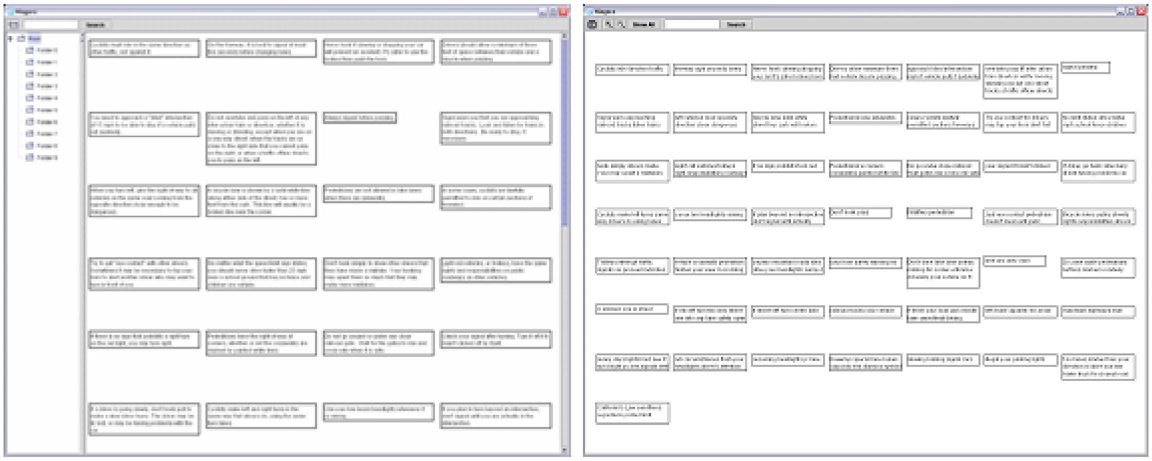


Figure 18: The starting state for the two interface conditions in the text-based study. The folder interface is on the left and the ZUI interface is on the right.

Because several unique features are included in the zooming condition, the study does not control for the effects of these individual features. This study is instead intended to identify any high order benefits of a well designed ZUI over a folder interface. Additional studies would be needed to determine which ZUI features have the biggest role in any overall benefits found.

As in the previous study, both interface conditions implemented a simple form of "bumping" to prevent objects from occluding one another. Here again, bumping effectively increased the demand for space, which was expected to highlight differences between the two interfaces. A similar effect could have been achieved by increasing the number of objects in the tasks, although this would have also increased completion times.

The study task involved putting text into groups based on common topics. In contrast to the shape study, subjects were asked to put objects into groups of 5 rather than groups of 10. The smaller groups were chosen because the large groups would be too easy for a text-based task and would not reflect the true difficulty of the task. The timed tasks again consisted of 50 objects resulting in 10 groups of 5.

The text objects used in the study were facts taken from the California Drivers handbook. This is expected to be a familiar topic for most of the subjects, though with enough details to be representative of an actual organization task. The tasks used in the study had an average of 21.3 words with an average of 5.5 letters per word. Text for the objects was shown at a 12 point SansSerif font.

Text describing the task and interface conditions was presented to the subjects in a 320x1024 window that was visible throughout the task. This meant that they completed the study tasks using the display's remaining 1280x1024 pixels. For both conditions, subjects were first presented with a tutorial on how to use the interface. Then, for the first condition only, subjects were given a single practice task consisting of 10 objects that had to be put into 2 groups of 5. Finally, subjects were given the actual timed tasks. Subjects were told that although the tasks were timed, they should try to find the best possible groups, keeping in mind that the objects in the groups did not have to be completely similar. Subjects were told to indicate their final groupings through automatic grouping in the zooming condition and with folders in the folder condition. The ordering of the interface condition and task were counterbalanced to reduce ordering effects.

At the beginning of the timed task, the text objects were distributed in a random order. For the zooming condition, the text objects were distributed in a single large grid. In the folder condition, the objects were distributed in single vertically-scrollable grid in the "root" folder. The folder condition also provided 10 pre-created folders in which subjects could put their final 10 groups. As in the shape study, there was a tradeoff here in the design of the starting condition for the folders task. Instead of being distributed in the “root” folder, objects could have been distributed evenly in folders as was done in the shape condition. The reason the scrolling condition was chosen here over the folder condition was because of the smaller group size for text rather than shapes. As a result, distributing the objects into 10 groups would unnecessarily segment the text objects, making comparisons unnecessarily difficult. Moreover, the text objects used here had a slightly smaller bounding box than the shapes allowing about 20 objects to fit on screen at once. This reduced the amount of scrolling needed in the single folder distribution, making it further preferable to the multi-folder distribution for this task.

One feature that was implemented in the zooming condition that could have also been implemented in the folder condition was counters in the group labels. Two subjects in the “labeled both” group specifically commented on this and manually added counters to several of their folder labels. Nonetheless, it is expected that this feature would have made little difference in the overall results since determining group counts was a relatively fast operation already.

Again, it should be noted that folders did not have to be used with the folder-based interface. Although subjects were asked to put their final group selections into individual folders, they were free to keep objects in the root folder during the task, using the scrollbars instead of folder icons to navigate. In the experiment, all subjects were observed to use this strategy to some extent.

The software logged all relevant operations during the timed tasks. These logs were analyzed to help explain results found from other measures. Following each of the two tasks, subjects were given a questionnaire regarding their experiences with the tool. One of the more unique questions, given immediately following each condition, asked subjects to estimate how long they thought the task took them. In a study involving various web browser tasks, Czerwinski et al. demonstrated a relationship between this so-called "subjective duration assessment" and task success rate [23]. This measure was collected here with the hope that it would provide an implicit measure of user satisfaction in completing tasks with each interface. The full post-task questionnaire is available in Appendix A.

2.7.2 Results and Discussion

Each of the subjects' final 10 groups was rated based on quality by 2 judges. Each of the groups was given a score from 1 to 5 based on the largest subset of related items in the group. Since all the facts were related to driving and driving safety, the groups had to have some additional topic in common in order to be related. Group labels were not counted in the rankings since not all groups had them. The final quality score was a number from 10 to 50. The average of the two judges scores were used in the comparison below. An analysis of inter-rater reliability between the two sets of scores found a linearly weighted Cohen's Kappa of 0.71. Since this was above the generally accepted reliability level of 0.70, the scored were not re-rated.

The means for the dependent variables measured are shown in Table 5. Preliminary paired two sample t-tests were initially used to analyze the results from the study. These were chosen over ANOVAs because only one independent variable and two samples were being compared.

The preliminary paired two sample t-tests were performed for the dependent variables of completion time, number of labeled groups, quality of groups, and the questionnaire items involving subjective duration, subjective satisfaction, and subjective confidence in groups. There was a highly significant effect of interface type on the questionnaire item "What is your overall reaction to using this tool for the task?" (1=Frustrating,5=Satisfying) at $p < 0.003$ with means of 4.21 (s 0.89) for zooming and 2.79 (s 0.89) for folders. There was also a significant effect of interface type on the number of groups labeled with $p < 0.05$. The mean number of labeled groups was 4.86 (s 5.05) for zooming and 7.50 (s 4.15) for folders. Graphs of these two measures are shown in Figure 19.

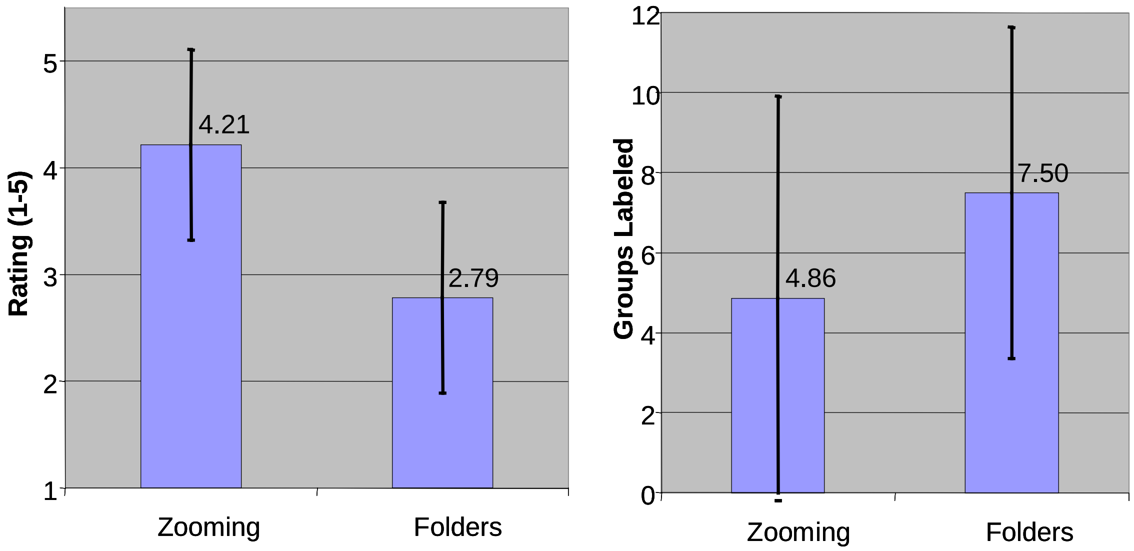


Figure 19: Graphs of the two measures on which interface type had a significant effect. The graph on the left shows the subjective rankings for the questionnaire item “What is your overall reaction to using this tool for the task?” The graph on the right shows the number of groups labeled in each condition.

It was not surprising that the difference in task completion times was not initially significant. Subjects appeared to have a wide range of computer skills, cognitive abilities, problem solving strategies, and engagement with the task. Also important was that completion time was not emphasized as heavily as it was in the shape study. This was done on purpose since the grouping criteria with text were less clear and less absolute. This meant that subjects could make a tradeoff between speed and quality in creating their groups. Since the interface was anticipated to have less of an effect if subjects were concerned with speed, we chose instead to emphasize quality in the task description.

Table 5: The means and standard deviations of dependent variables measured in the study, divided on the single within-subjects factor of tool type.

	Zooming	**Folders**
Completion Time (min.)	18.57 (σ 6.43)	21.20 (σ 8.35)
Group Quality (10-50)	46.71 (σ 2.71)	47.00 (σ 2.31)
Number of Labeled Groups (0-10)	4.86 (σ 5.05)	7.50 (σ 4.15)
Raw Subjective Duration (min.)	16.93 (σ 7.20)	19.50 (σ 9.35)
Subjective Duration Difference (min.)	-1.65 (σ 5.20)	-1.70 (σ 5.61)
Subjective Satisfaction (1=Frustrating, 5=Satisfying)	4.21 (σ 0.89)	2.79 (σ 0.89)
Subjective Confidence in Groups (1=Not Confident, 5=Confident)	3.36 (σ 0.93)	3.36 (σ 1.15)

2.7.2.1 Group Labeling Strategy

As mentioned above, several differences were observed between subjects in how they used the tools to solve the tasks. This was reflected in the preliminary analysis that showed an effect of tool type on the number of groups labeled. In fact, three clear categories of labeling strategies were evident in the data. Three subjects (two women and one man) fell into a first category that did not use any labels. Four subjects (two women and two men) fell into a second category that used labels only on their folders and none of their zooming groups. The remaining seven subjects (one woman and six men) fell into a third category that put labels on both their folders and their zooming groups. Table 6 shows the means for each of the dependent variables divided by labeling strategy.

Table 6: The means and standard deviations of dependent variables measured in the study, divided on the within-subjects factor of tool type and the between-subjects factor of labeling strategy.

	No Labels		**Labeled Folders**		**Labeled Both**	
	Zooming	**Folders**	**Zooming**	**Folders**	**Zooming**	**Folders**
Completion Time (min.)	15.15 (σ 2.70)	25.46 (σ 13.85)	13.77 (σ 6.08)	20.93 (σ 10.29)	22.79 (σ 5.16)	19.53 (σ 4.67)
Group Quality (0-50)	45.67 (σ 5.01)	44.83 (σ 2.57)	46.13 (σ 0.85)	46.25 (σ 1.85)	47.5 (σ 2.38)	48.36 (σ 1.68)
Number of Labeled Groups (0-10)	0.0 (σ 0.0)	0.0 (σ 0.0)	0.0 (σ 0.0)	9.50 (σ 0.57)	9.71 (σ 0.49)	9.57 (σ 1.13)
Raw Subjective Duration (min.)	13.33 (σ 7.64)	21.67 (σ 17.56)	11.50 (σ 5.07)	16.25 (σ 8.54)	21.57 (σ 5.38)	20.43 (σ 6.29)
Subjective Duration Difference (min.)	-1.82 (σ 6.07)	-3.80 (σ 5.37)	-2.27 (σ 2.45)	-4.68 (σ 3.42)	-1.22 (σ 6.55)	0.89 (σ 6.07)
Subjective Satisfaction (1=Frustrating, 5=Satisfying)	5.00 (σ 0.00)	2.67 (σ 0.58)	4.25 (σ 0.50)	2.50 (σ 0.58)	3.85 (σ 1.07)	3.00 (σ 1.15)
Subjective Confidence in Groups (1=Not Confident, 5=Confident)	3.00 (σ 1.00)	3.00 (σ 1.73)	3.50 (σ 0.58)	3.50 (σ 1.00)	3.43 (σ 1.13)	3.43 (σ 1.13)

As a result, the data was analyzed treating the number of labeled groups as a between-subjects factor with the three conditions described above. This led to a series of 3x2 ANOVAs with labeling strategy as a 3 condition between-subject factor and tool type as a 2 condition within-subjects factor. ANOVAs were used here in preference to t tests since the results involved a complex combination of factors. These analyses indicated a significant effect of tool type on completion time with $F(1,11)=6.39$, $p < 0.03$ and on the questionnaire item "What is your overall reaction to using this tool for the task?" (1=Frustrating,5=Satisfying) with $F(1,11)=18.24$, $p < 0.001$. There was also an interaction between tool type and labeling strategy affecting completion times with $F(2,11)=5.76$, $p < 0.02$.

It is notable that all 3 groups of subjects showed a significant preference for zooming 4.21 (σ 0.89) over folders 2.79 (σ 0.89) with the same average ratings as in the shape study. In the shape study, the zooming condition was fairly intuitive and similar to existing tools for working with visual data, such as map browsers or photo editors. In this study, subjects were presented with the two completely unfamiliar interface techniques of automatic text reduction and automatic grouping. Although this novelty itself may account for some of the difference in preference, it seems unlikely that subjects would show such a strong preference if the techniques did not provide some benefit for the task.

A cursory look at the data across subjects is revealing. shows the patterns for the four dependent variables of completion time, subjective duration assessment, subjective satisfaction, and group quality. For each labeling strategy, the four graphs plot the difference between measures for the two tool types. Although not all of the differences shown in the graphs are significant, they are indicative of a consistent tendency between the three labeling conditions.

In general, it seems very unlikely that the different labeling strategies are the cause of the differences seen here. Looking in particular at the graph of completion times, it seems unlikely that teaching users *not* to label their groups will make them more efficient at the task with zooming. Instead, a more plausible explanation of these results is that the labeling strategies reflect three different kinds of users. Because this task heavily exercises short term memory and different spatial abilities, the labeling strategies could be indicative of individual differences on these dimensions.

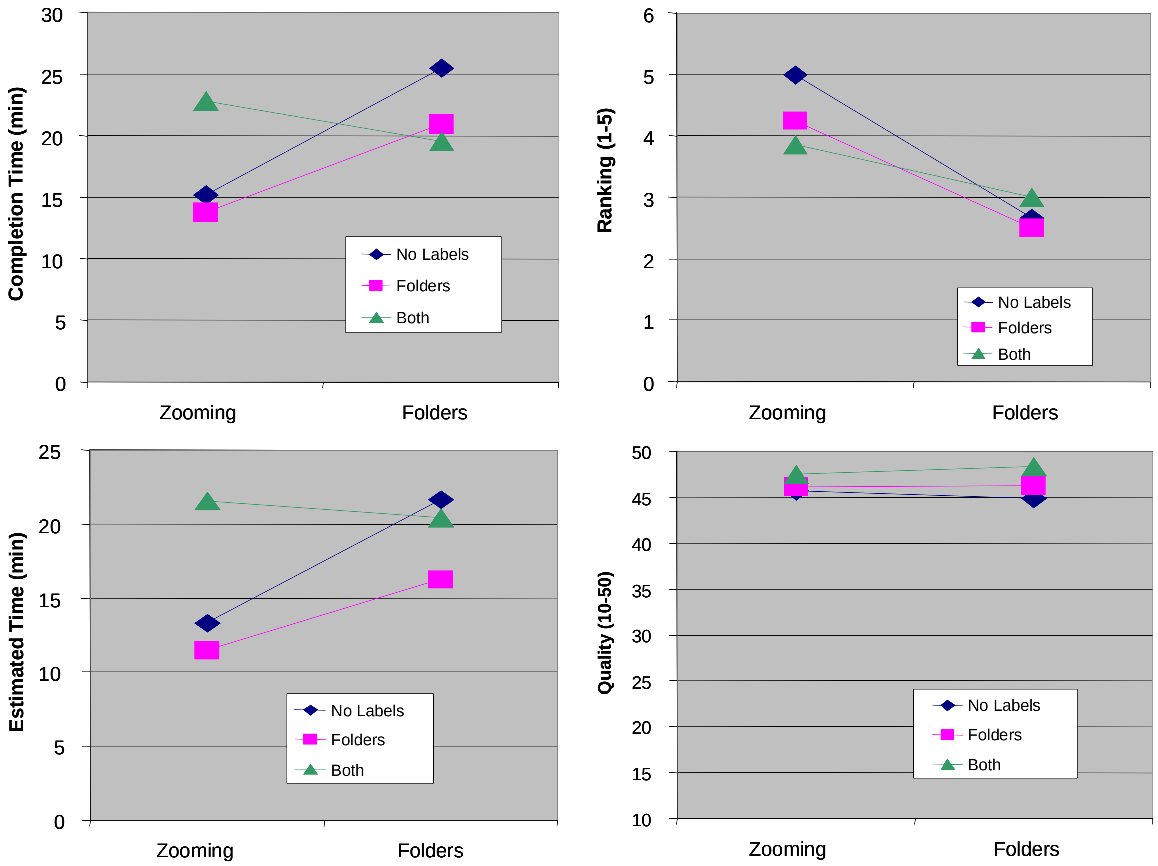


Figure 20: Graphs of the differences between four measures for each of the 3 labeling strategies. The top left graph depicts completion time. The bottom left graph depicts subjective duration assessment. The top right graph depicts subjective satisfaction. The bottom right depicts group quality. Positive values represent higher values in the zooming condition, while negative values represent higher values in the folder condition.

Our informal observations offer further evidence to suggest individual differences between these three categories of users. Subjects who labeled in both conditions seemed to be generally more comfortable with the computer and hence, more willing to explore the nonstandard navigation techniques provided in the zooming interface. An analysis of the operation logs from the software used in the study indicates that there is indeed a difference in the number of navigation operations performed by these users under zooming. This navigation measure combines zooming, panning, and scrolling and is shown in Figure 21. A single factor between-subjects ANOVA indicates that these results are not significant. Nevertheless, the general pattern further indicates that individual differences, rather than labeling strategy, may be at the root of the performance differences between the groups.

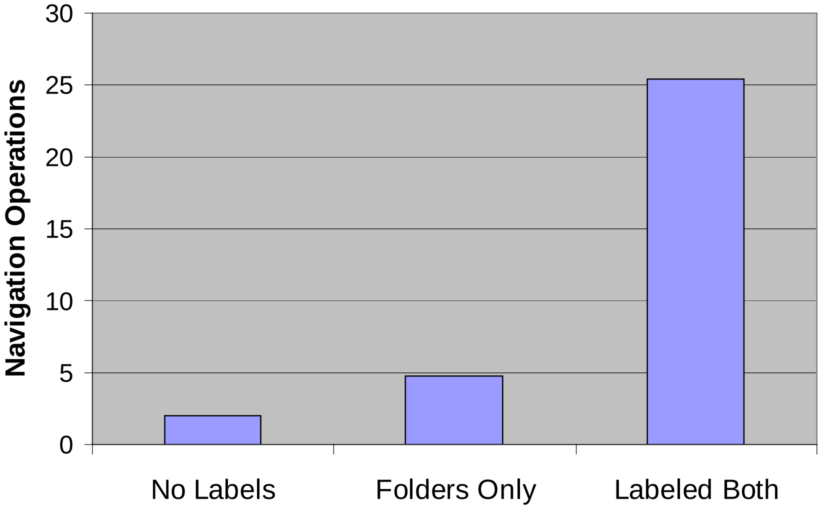


Figure 21: A graph of the number of navigation operations between the different labeling strategies. The difference between strategies was not significant.

Users were also asked directly about their use of zooming and automatic text reduction on the follow-up questionnaire. Ten of the fourteen subjects indicated that they rarely or never used zooming during the task. This may at first glance seem surprising since this was, after all, a *zoomable* user interface. However, this was actually a desirable result since automatic text reduction and full text popups were added to increase the value of zoomed out text objects and reduce the need for zooming. The responses to the automatic text reduction question further support this finding. Only three of fourteen users indicated that they rarely or never made use of the automatically reduced text. Interestingly, two of the three users who did not use the reduced text indicated a heavy reliance on zooming. It is also possible that these three users were not aware that they were using the reduced text since it was always visible and did not require any effort to invoke.

Additional studies are needed to determine the specific cognitive mechanisms that are driving these different labeling strategies. Regardless, the results indicate that some categories of users are significantly faster at the task with zooming than with folders and with no significant difference in quality. Since ZUIs are relatively novel, additional training and experience with zooming could yield an even greater benefit for these categories of users. Of course, since the remaining users were faster with folders than with zooming, the interface should be flexible in providing access to both mechanisms for managing screen space. Here again, the performance of this category of users may also improve under zooming with additional experience and training.

It was somewhat surprising to see that tool type did not affect subjects' confidence in their final groups. Feedback from several of the subjects on these confidence ratings indicated that the task may have been too difficult to reflect any differences here. In particular, the most difficult portion of the task seemed to come towards the end when there were a few objects left that did not fit into existing groups. This often meant that groups had to be reorganized to accommodate the outliers. However, very few subjects were ultimately able to incorporate all the outliers to their satisfaction. As a result, subjects seemed to give equally low confidence ratings based on their frustrations at the end of the task. A more revealing measure might have been how many different organizations they tried before giving up. However, this data would be difficult to judge for the folder condition since many of the groups were initially represented by informal spatial clusters in the “Root” folder.

Lastly, the effect of tool type across labeling strategies on the raw subjective duration assessment values collected in the study approached significance in a 3x2 ANOVA with $F(1,11)=4.818$, $p < 0.06$. However, the goal behind this metric is to take the difference between the raw subjective duration and the actual task duration to indicate task difficulty. Tasks that are perceived as being easier to complete have a subjective duration that is lower than the actual duration and tasks that are perceived as being harder or interrupted have a higher subjective duration than actual duration. For the data above, comparing these subjective duration differences did not reveal any significant effects. One reason for this might be that there really is no difference between the two interface conditions for this metric. Another possibility is that the study tasks have completion times that are too long and are therefore beyond the useful range of the metric. The paper that introduces this metric had average task completion times of around 3 minutes [23]. In contrast, the tasks in this study had completion times of around 20 minutes. These longer completion times are likely to introduce random noise into the estimates that may hide any effects of the subjective duration assessment.

2.7.3 Revisited: Scaling Effects and Other Parameters of Task Difficulty

The parameters of difficulty found for the shape task in the previous section require some adjustment to be applied to text. Perhaps the biggest difference between the two object types is that the text objects do not have clear attributes. Instead, they have a more amorphous notion of topics. Text objects also don’t have a discrete number of values for each attribute as with shapes. Instead, the values for each topic are continuous weights, from 0 to 1 for example, indicating the extent to which the object is about that topic. This is illustrated by a sample text object from the previous study that contained the text “Pedestrians are not allowed in bike lanes when there are sidewalks.” This object has at least three potential topics including pedestrians, bikes, and lanes. However, the weightings of these topics are not equal. “Pedestrians” is likely to be weighted more heavily for this object since it is the subject of the sentence.

Topics take longer to discover than the perceptual attributes of shapes. This is because topics are often represented by only a small percentage of the words in an object. In some cases, the topic is implicit and topic related words may not appear at all. This means that the actual number of objects in a task plays a bigger role with text than with shapes. In the shape task, the data suggested that the number of groups, rather than the number of objects, was the primary scale factor since each individual object took minimal time to process. For text objects, the number of groups also affects completion times, but the actual number of objects is also important since the time to process the individual objects is much longer.

A second difference between shapes and text is in defining membership criteria for groups. The shape tasks described above had membership criteria that were well defined and absolute. This is not necessarily the case for groups of text objects. Topics for text objects can be related through hierarchical abstraction. As a result, a group may have membership criteria involving topics that do not explicitly appear in any of the group members. For example, in the tasks in the previous study, all of the facts were on the general topic of "driving" though few objects actually contain this word.

Overlap between membership criteria is likely to have a similar effect on the difficulty of the text task as it had in the shape tasks. Of course, overlap is somewhat more ambiguous for text objects since topics are weighted and hierarchical. The weightings on objects mean that objects can have partial overlap on membership criteria. For example, a text object containing the text "Cyclists must ride in the same direction as other traffic" contains some degree of overlap with the "pedestrians" object mentioned above since they both implicitly mention bicycles. However, since the two objects have different weightings on this topic, they do not have complete overlap on this criterion. In addition, overlap can also have different weightings at different levels of abstraction since the topics are hierarchical. For instance, the "pedestrian" object and "cyclist" object would have greater overlap under membership criteria such as "Non-motorized locomotion."

Another potential similarity between the text and shape tasks is that the difficulty increases with the ratio of the total number of topics to the number of group membership criteria. Just as in the shape condition, as the number of topics in a group of objects increases, the number of dimension on which the objects can be grouped also increases. In this case, if the number of topics used to group the objects remains constant, then the ratio of similar topics to non-similar topics decreases for the group.

These differences and similarities with the shape task suggest the following revised relationships between task parameters and task difficulty:

1) Difficulty **increases** with the number of objects to be grouped

2) Difficulty **increases** with the number of groups needed

3) Difficulty **increases** with the ratio of the total number of topics to the average number of topics used for membership criteria in each group

4) Difficulty **increases** with the number of overlapping membership criteria

2.8 Niagara in Use

As part of our future work, we intend to distribute Niagara to a wider audience to get feedback in using it for actual presentation authoring and other writing tasks. In the meantime, we have used Niagara ourselves for a number of authoring and bookkeeping tasks including: organizing 3 presentations, organizing 3 chapters in this dissertation, creating 6 demos, organizing a movie script, and keeping track of 2 to-do/bug lists.

These initial uses have largely confirmed our hypotheses regarding the utility of various Niagara features. We have found Niagara's unconstrained 2D layouts, bumping behavior, and facile text manipulation to be extremely natural and effective for these kinds of tasks. Automatic grouping has also met our expectations in providing a low-effort transition from informal spatial clusters to more formal structures. The zoom controls provided in Niagara have also proved much more valuable than scrolling or folders as mechanisms for acquiring more space. Certainly, zooming was not needed to complete all of these tasks since they varied in the amount of information to be organized. However, zooming seems to become more important as tasks last longer and more information needs to be organized.

Chapter 3:
Niagara Features and Implementation

Niagara was designed as a stand-alone java application that stores its data in a custom XML file format. The first two versions of Niagara described in Chapter 2 were built on top of Jazz [10], a Java toolkit for building ZUIs. The final version of Niagara was built on top of Piccolo [8], a ZUI toolkit designed to replace Jazz. The final Jazz version of Niagara contained 55 source files representing 183 classes. These classes contained around 23K lines of code not including comments and 28K lines of code including comments. The final Piccolo version of Niagara contained 86 source files representing 328 classes. These classes contained around 13K lines of code not including comments and 17K lines of code including comments.

3.1 Unique Features in Niagara version 1

3.1.1 Objects and Interactions

There are three object types in Niagara version 1 including text objects, lists, and subspaces. The text object provides a simple text editor that can wrap at a specified width or grow to fit all the text. The text object is the foundation of the Niagara workspace and can contain no other objects. The list object is a collection object that enforces a vertically-ordered left-aligned layout. Lists can contain text objects, subspaces or other lists. Subspaces are also collection objects that can contain text objects, lists, or other subspaces. Subspaces have their own conceptually infinite 2D workspaces in which objects can be arranged with no constraints on layout. Lists and subspaces both have instances of a title object that is used to move the object and displays an editable text label. The top level Niagara workspace has all the properties of a subspace but is special in that it does not have a title and can not be deleted.   The three object types are illustrated in Figure 22.

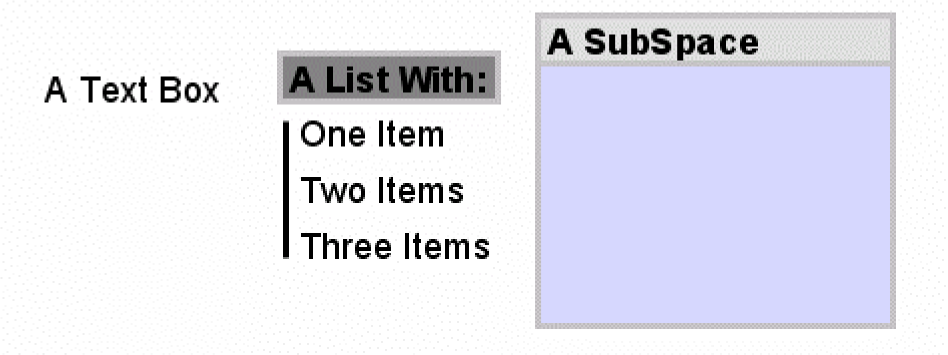


Figure 22: A screenshot of the three object types in the first version of Niagara. The objects from left to right are a text object, a list, and a subspace.

These object types are fairly typical of many authoring tools and can be found in systems such as VKB [100], Tivoli [84], and Storyspace [107]. Niagara's uniqueness comes primarily from object interactions. For instance, lists provide interactive previews as shown in Figure 23. Previous systems have used markers to indicate the point of insertion in a list, but this technique does not give a good representation of what the list and the overall workspace will look like once the insertion is completed.

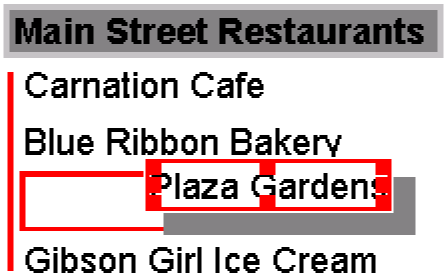


Figure 23: A screenshot of an item being added to a list. The list provides a preview of what the list will look like when the object is inserted.

Subspaces also provide a unique interaction shown in Figure 24. At each level of nesting, subspaces implement a reduction in scale. This scale reduction is also reflected during objects movements so that objects over a subspace are immediately reduced in scale to reflect their appearance when the move is completed. Figure 25 shows the result of moving an object over nested subspaces.

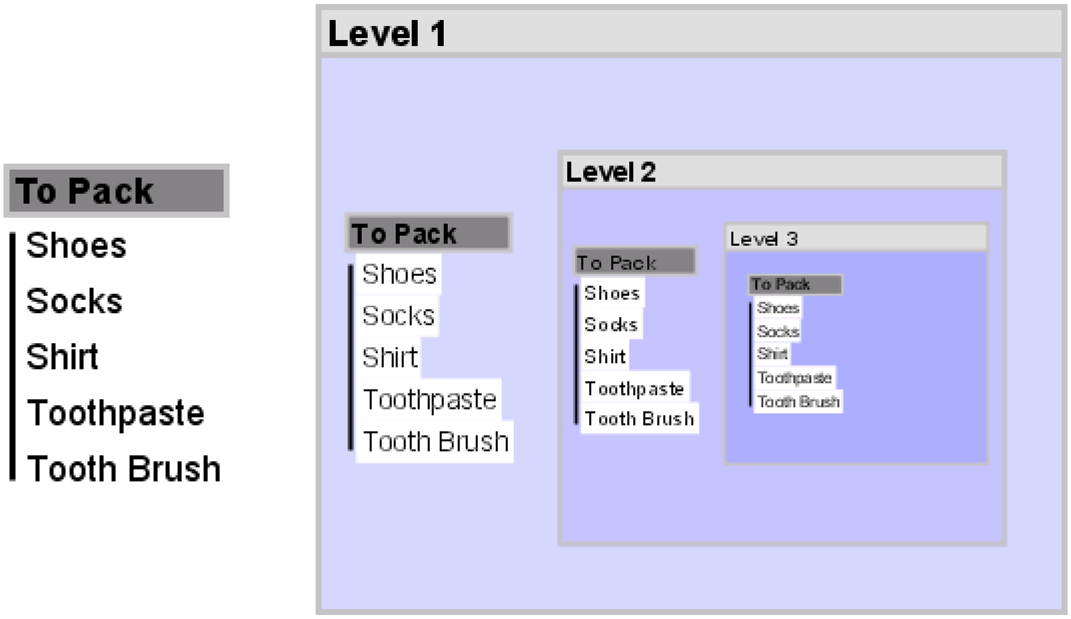


Figure 24: A screenshot of a list shown in three different levels of nested subspaces. The scale of objects is gradually reduced as the nesting depth of the subspaces increases.

In addition to reducing scale, the subspaces shown above clip objects that extend beyond the subspace' rectangular boundaries. Other subspace behaviors that were considered include reducing the scale of the subspace to fit all contained objects or growing the subspace to fit all contained objects. Ultimately the current implementation was chosen because it provides a consistent scale reduction across subspaces while allowing the author to control the size of the space.

In addition to the above features, subspaces behave much like normal operating system windows. They can be resized, maximized, and restored. In order to indicate their nesting level when maximized, subspace backgrounds are painted a progressively darker shade of blue. This is shown in Figure 24. Techniques such as those described by Furnas et al. [38] could be used to increase the number of levels of nesting possible.

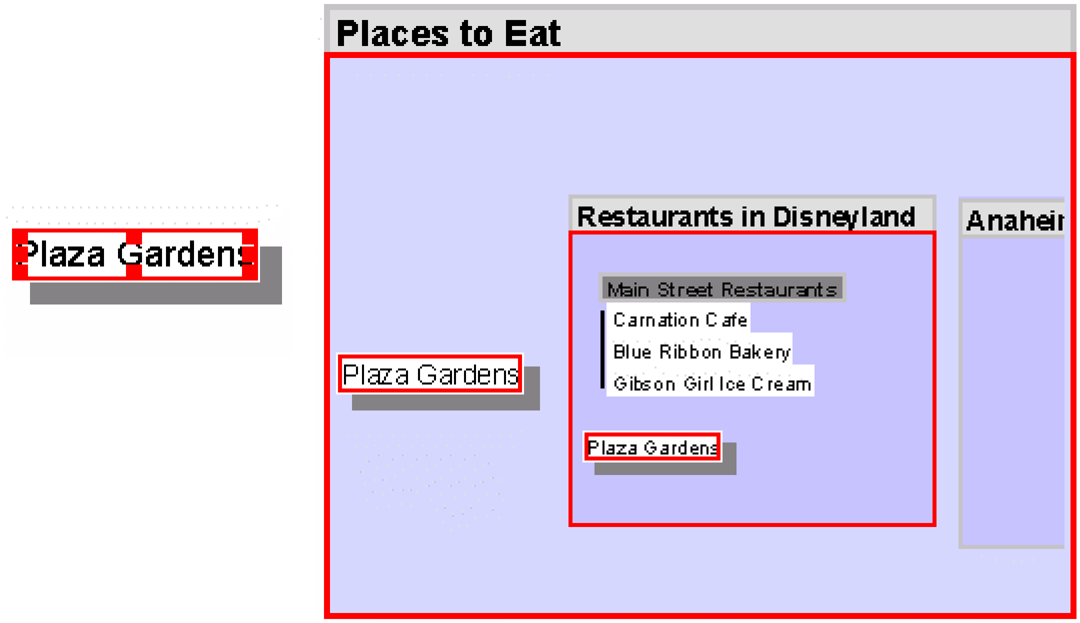


Figure 25: A screenshot of an item being moved over nested subspaces. The item reduces in size to preview the size it will take when dropped in the space.

Another of Niagara's distinguishing interactions is its no-occlusion contract. This specifies the amount that objects are allowed to overlap and was inspired by applications such as Data Mountain [93] and Flatland [74]. For example, Figure 26 shows a text object being dropped between two subspaces. Here, the subspaces are moved apart to make room for the text object. In this specific figure, no overlap is allowed, though this could also be adjusted to allow the occlusion of a certain number of pixels.

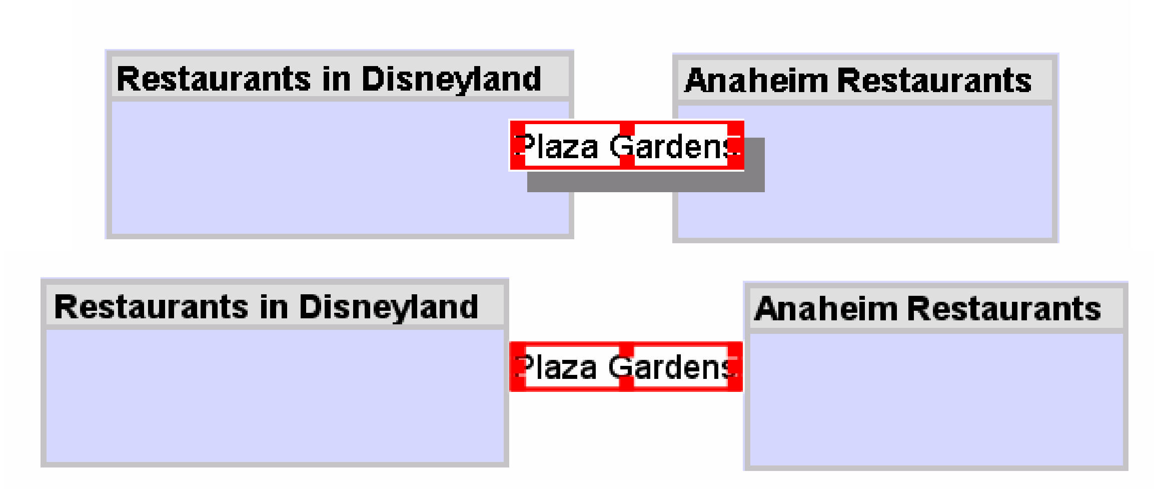


Figure 26: A screenshot of a text object being dropped between two text boxes. When the text object is dropped it "bumps" other objects out of the way. Prior to being dropped the text object is in "flying" mode.

The overlap avoidance described above is also useful for interactively positioning objects. It allows an author to use any object as a tool to make space or align a set of objects. An example interaction is shown in Figure 27. Nonetheless, this kind of interaction can also be disruptive in certain situations. Sometimes the author needs to move an object across the workspace without disturbing existing arrangements. Consequently, Niagara also implements a flying mode that allows an author to move objects without disturbing objects on the workspace. This mode change is portrayed to the user through via the addition of a shadow, shown in Figure 26. The shadow is meant to be suggestive of the object flying above the workspace' surface.

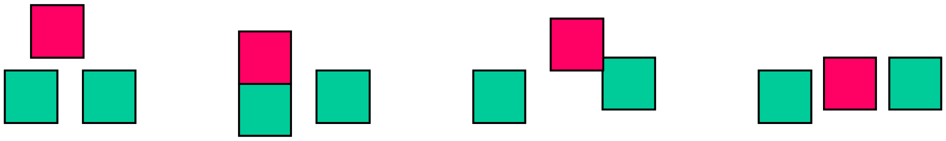


Figure 27: The magenta object is used as a tool to make space between the green objects.

3.1.2 City Lights

A fundamental challenge in the design of ZUIs or 2D workspaces like Niagara is to support detailed authoring operations while still providing context or awareness of the larger containing space. For abstract content, these workspaces often lack good facilities for portraying objects that are currently not in view. The two traditional techniques used to address these problems are overview plus detail and fisheye. However, existing instantiations of these types are often problematic for information work. Overviews do not necessarily work well for textual content and the overview window intrudes on the primary workspace. Similarly, most fisheye techniques introduce distortion that can be harmful to the user's understanding of spatial location.

City Lights is a simplified fisheye technique for displaying contextual information in clipped views of 2D information workspaces. The name "City Lights" was chosen in keeping with the metaphor of information clusters as cities [26]. Just as the lights from a physical city are visible at night from greater distances than the city itself, our technique makes properties of information cities visible when those cities are not in view.

City Lights uses a small, fixed amount of screen space on the frames of windows for peripheral awareness indicators. The primary advantage to limiting the space allocated to City Lights is that the majority of the screen can be left undisturbed. This frees users from having to adjust to a novel presentation of data in the focus region. More importantly, users do not have to learn new techniques for authoring in a warped or distorted space. Instead, systems utilizing City Lights can function for the most part as if the City Lights were not there. In their most rudimentary form they can use a one pixel line on the window's border.

One way to understand City Lights is as meaningful decorations on the frame of a window over an infinite canvas. Any visual information objects on the canvas that are not directly under the window are clipped. As shown in Figure 28, that part of the canvas on which there are visual information objects is called "populated space" and that part which can be seen at a given time is its "viewed space." Restated, elements that are in populated space but not the viewed space are clipped. Panning operations make it possible to move the viewed space around in the populated space – analogous to moving the "window" over the "canvas."

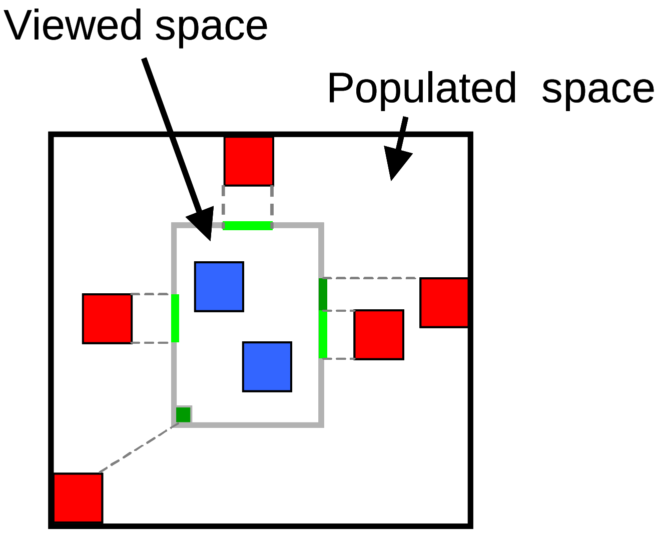


Figure 28: A simplified portrayal of a City Lights technique.

To provide peripheral awareness of the populated space, City Lights provide information about clipped objects. Such information about a clipped object could include many things, such as its direction from the boundary, its size, its distance, its type, its recency or time since last edit, or a summary of its information content.

The initial implementation of City Lights used an orthogonal projection to map clipped objects' profiles onto the window borders. Figure 28 shows an abstracted depiction of the technique with blue boxes representing visual information objects that are inside the viewed space and red boxes indicating clipped objects in populated space. The city lights are shown as green colorings on the outer bounding box of the viewed space. An orthogonal projection from the clipped objects onto the frame bounding populated space delineates the edges of the city lights. Because this orthogonal projection ignores objects in the corners, decorations were also added to indicate when objects were in the corner regions.

The need for peripheral awareness also often exists simultaneously at multiple levels in a hierarchy. *Nested* hierarchy levels can themselves introduce problems in the management of focus and context. The user needs to maintain regions of focus at several levels simultaneously. Our approach enables clipping of the populated spaces around each viewed space at any level, and manages the competition for screen space for each window. Figure 29 shows City Lights on Niagara subspaces at multiple levels.

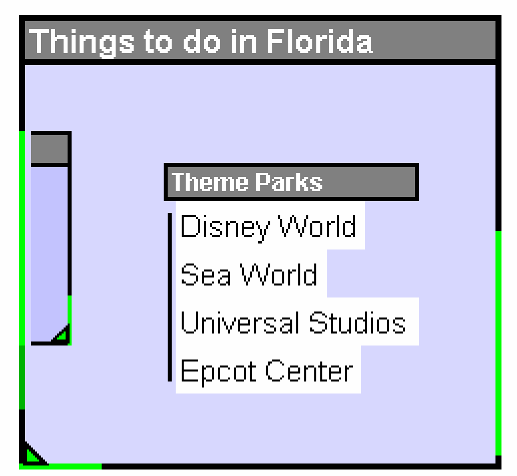


Figure 29: Nested Niagara subspaces with City Lights at multiple levels.

City Lights can also be interactive. The current implementation allows users to click on a City Light visualization to navigate to the clipped object. Future implementations can support more sophisticated interactions such as the fisheye behavior implemented by the Macintosh OS X taskbar.

The left screenshot in Figure 30 shows the initial Niagara City Lights implementation. These City Lights try to convey information about the size of off-screen objects. However, as the number of objects in the populated space increases, the window borders quickly become cluttered by this kind of representation. An alternative space-conserving approach, shown on the right in Figure 30, portrays only the object centers in the City Lights.

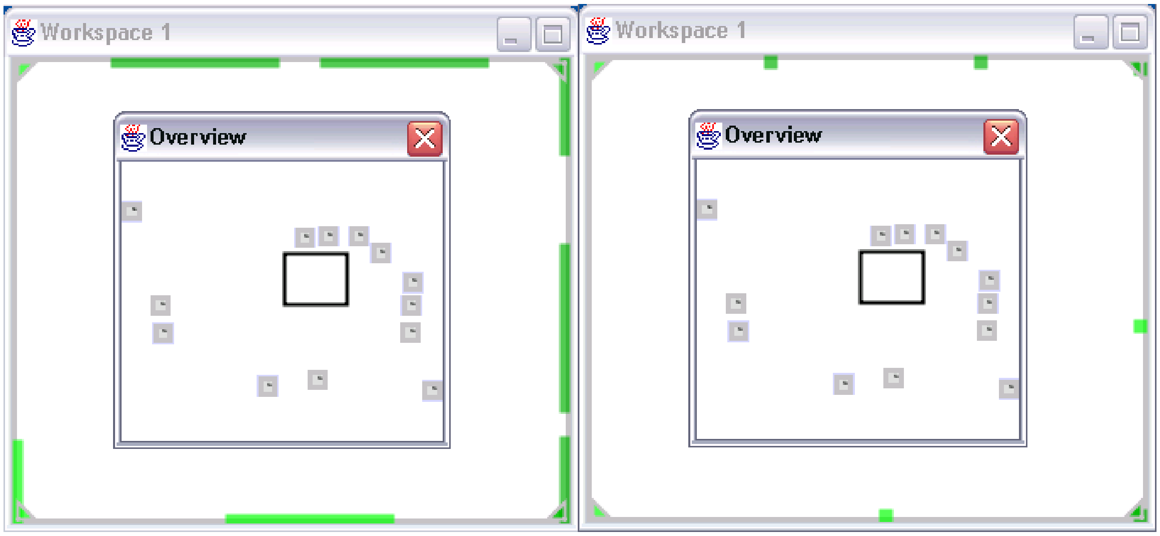


Figure 30: The first in a sequence of screenshots showing Niagara workspaces with 13 off-screen objects. To show these off-screen objects, overview windows are displayed on top of the primary workspace windows. The overview windows show the view rectangle in black and the unseen objects outside this view. These two screenshots show two versions of City Lights along the larger windows' border using an orthogonal projection. The left screenshot shows a line projection of the object bounds and the right screenshot shows a point projection of the object centers.

The orthogonal projections used in the original City Lights implementations are roughly based on the physical metaphor of light. This simple metaphor is particular intuitive since most windows are Manhattan rectangles and panning takes place in a Cartesian space. However, this projection requires special considerations for objects in the corners. In contrast, a radial projection can be more consistent in certain situations by eliminating the special case for the corners. Figure 31 compares these two projections side-by-side.

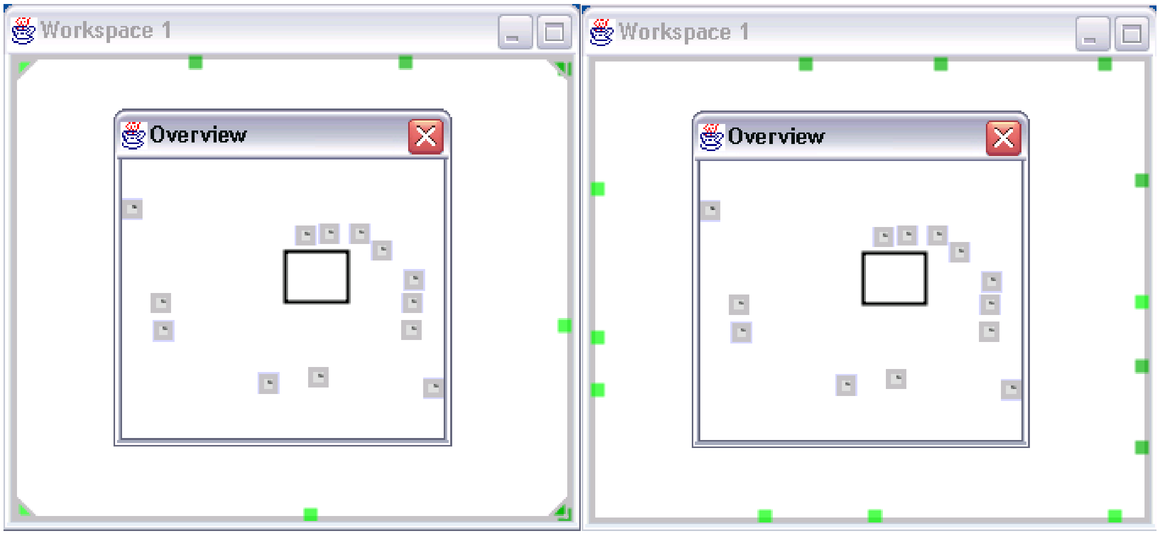


Figure 31: Two versions of City Lights are shown along the larger windows' border. The left screenshot shows an orthogonal projection of object center points and the right screenshot shows a radial projection of object center points.

A variant of City Lights was also developed that uses ellipses to represent off-screen objects. This is shown next to a radial point projection in Figure 32. Baudisch et al. named these circular projections "halos" and demonstrated their effectiveness for different kinds of peripheral awareness tasks [4]. This technique seems to be most suited for applications with a small number of relevant off-screen objects where distance judgments are important.

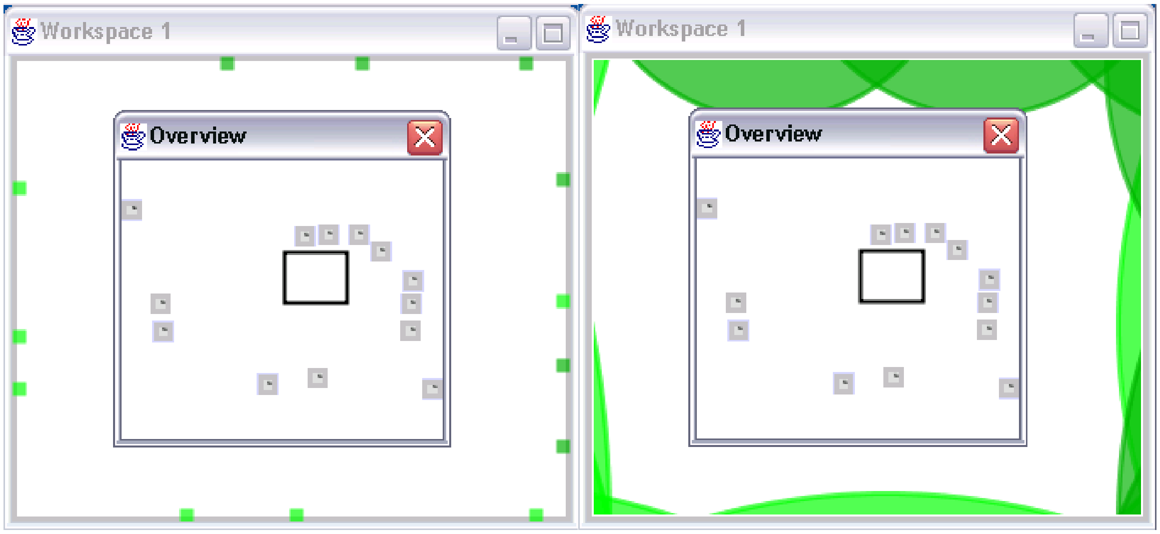


Figure 32: Two versions of City Lights are shown along the larger windows' border. The left screenshot shows a radial point projection of object centers and the right screenshot shows "halos" centered on the objects' centers.

Just as physical light attenuates with distance, City Lights can represent distance by manipulating contrast or color. In our current implementations, near objects receive a higher contrast color and far objects receive a lower contrast color. These color changes are shown in the screenshot on the left of Figure 33. City lights for objects that are nearer than a near boundary are colored dark green. City lights for objects that are farther away are colored light green. Indicators for near objects occlude indicators for far objects. This color difference is more apparent in the screenshot on the right of Figure 33, which assigns blue to near objects and red to far objects. In addition, City Lights can also use color gradients to indicate a wider range of distances.

More City Lights details, variations, and usage scenarios are described in our CHI short paper [121].

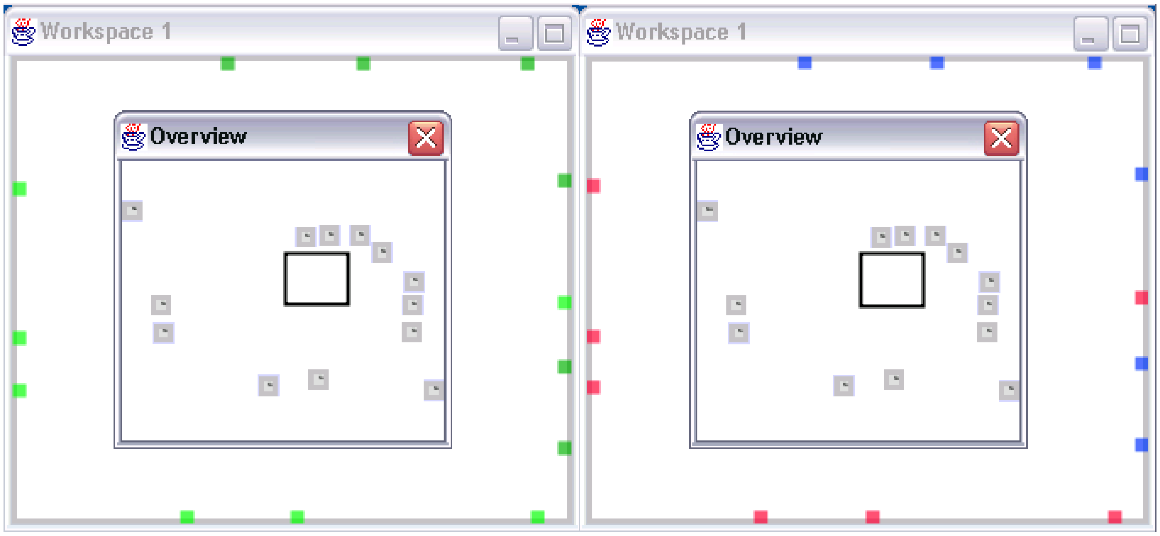


Figure 33: Two versions of City Lights are shown along the larger windows' border that use a radial projection of object center points. The left screenshot uses a binary color system with near objects in darker green and far objects in lighter green. The right screenshot makes the difference clearer by using blue for near objects and red for far objects.

3.2 Unique Features in Niagara version 2

3.2.1 Spatial Overview Features

One of the primary changes to the Niagara version 2 prototype was the addition of a spatial overview. Although spatial overviews are a frequently used interface technique, there seem to be few design guidelines for their use with abstract data. As a result, several designs were tried in our initial implementation. Figure 34 shows three of these designs that vary from the most accurate on the left to the most abstract on the right. The rightmost design is similar to the overview available in the VKB system [100].

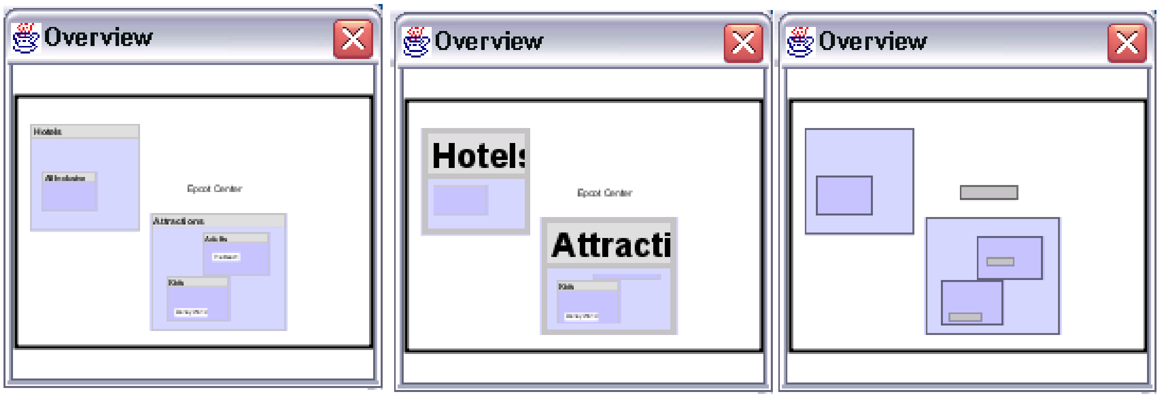


Figure 34: This figure shows three versions of the spatial overview. The leftmost version shows a standard geometrically reduced representation of the entire space. The center version shows an overview with semantic zooming on collection titles. The rightmost version shows an abstracted version of the objects in the workspace.

By default, Niagara used the center design in Figure 34. This design portrayed the overall geometric properties of the space while still preserving some of the high level text content. This design also facilitates the common operation of moving an item from the current workspace to a known collection. To support this interaction, the overview implemented drag and drop functionality. This meant that a user could drag an object from the main workspace and drop it into an off-screen subspace via the overview. However, we also noted that this functionality only worked for a single level. As result, we implemented a drill-down pop-up interface shown in Figure 35. When the user paused the mouse over a subspace in the overview during a drag and drop interaction, the system would pop-up a window that displayed an overview of the target subspace. This worked recursively so the author could drill down to access arbitrarily nested subspaces.

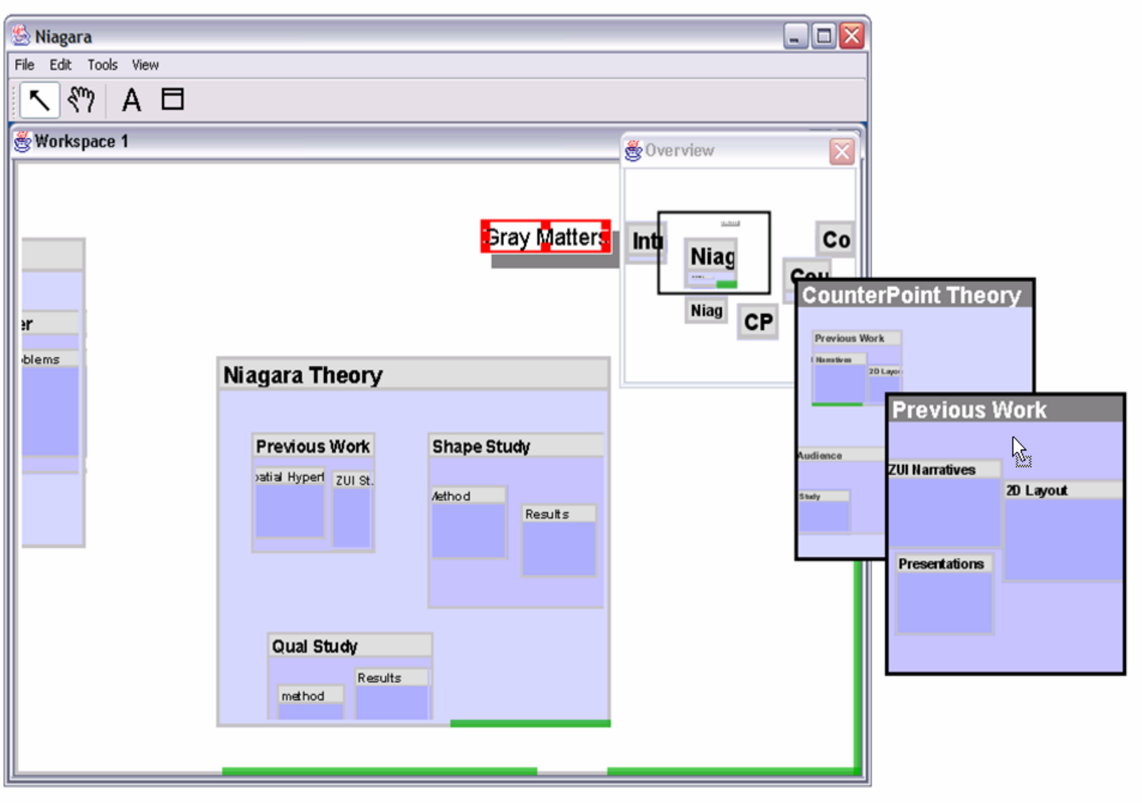


Figure 35: A screenshot of an interactive spatial overview. When the user pauses while dragging an item onto a subspace in the overview, the system pops up a window to show an overview of the current subspace.

The author does not always want to decide an object's exact position when moving the object into a subspace as in Figure 35. Instead, the author sometimes just wants to move the object into a subspace for later processing. In these cases, Niagara tries to automatically position the object as close to the center of the subspace as possible without overlapping other objects. This strategy leads to a good spatial packing of objects, reducing the size of the workspace that the author needs to manage later. The algorithm Niagara uses to find empty space and position objects is based on the work of Bell and Feiner [11].

Another feature in Niagara provided an overview of a subspace's entire contents when the subspace was selected. This is illustrated in Figure 36. This allowed an author to get a preview of subspace without having to maximize or manually resize the space.

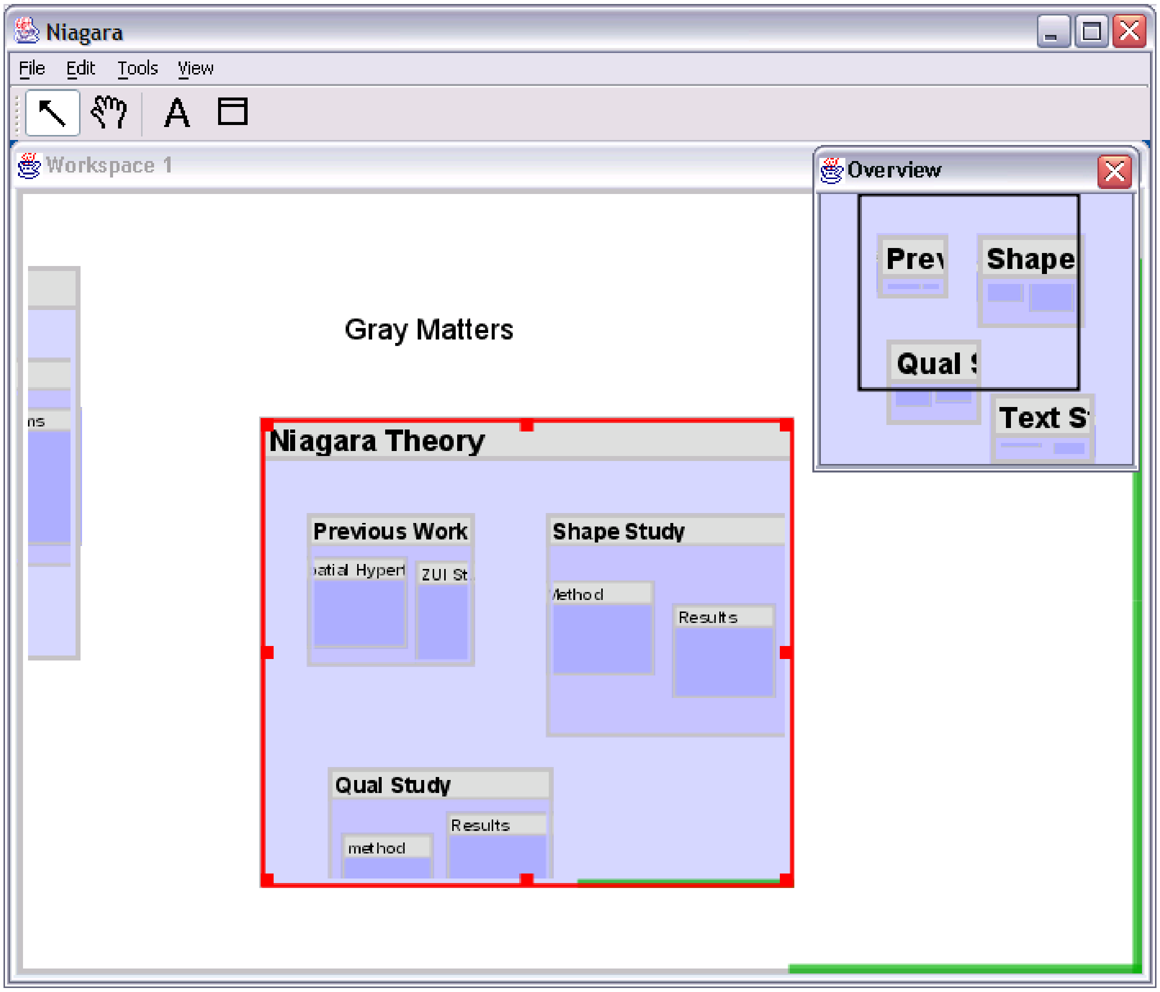


Figure 36: The overview switches to show the currently selected workspace.

An additional non-standard feature in the Niagara overview was the ability to move to non-populated space. Traditional spatial overviews restrict the view rectangle's movement based on the current workspace size. Niagara allows the user to move the view rectangle to arbitrary points in space, including those outside bounds of the currently populated workspace. This feature may be important for authoring workspaces like Niagara's since the author will often need a way to get more room to work.

Niagara's spatial overview was also adapted to work with the Focus Plus Context screen [3]. The Focus Plus Context screen uses a standard LCD flat panel for the high resolution focus and a video projector for the low resolution periphery. Figure 37 shows

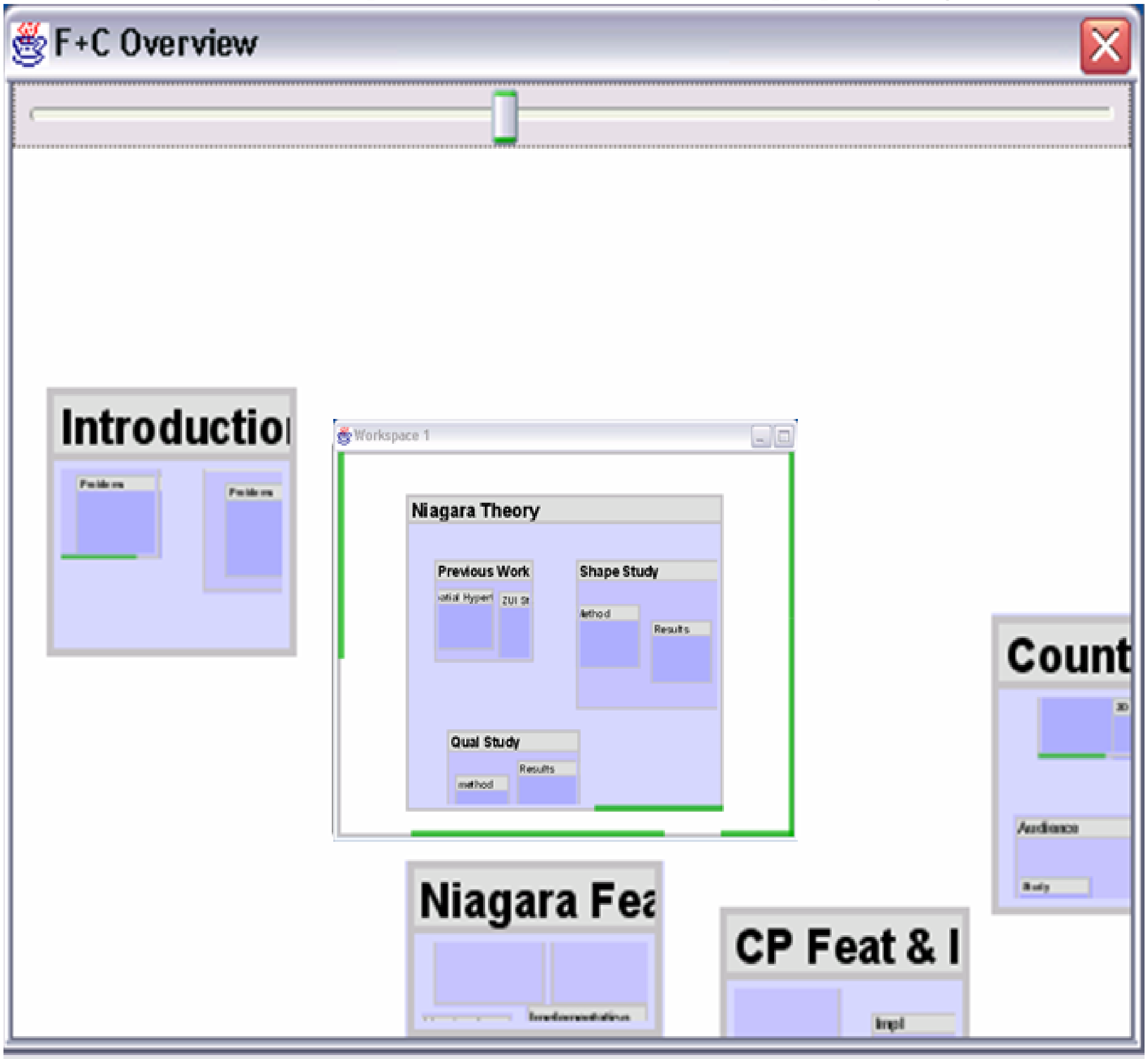


Figure 37: A screenshot of a prototype overview developed for use with the Focus Plus Context display.

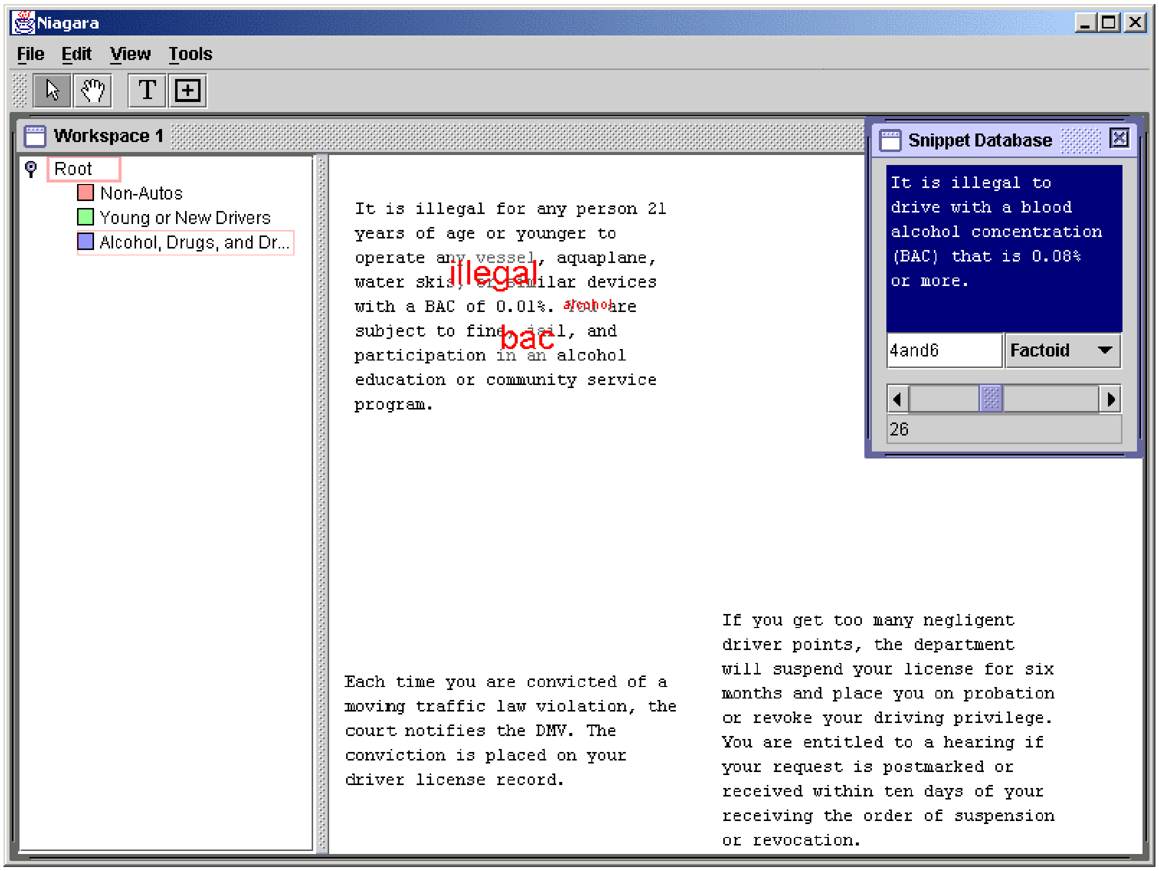
3.3 Unique Features in Niagara version 3

Three of the most important features added in the third round of Niagara design were automatic text reduction, automatic grouping, and full text popups. These features were described in Section 2.5. Several other interesting Niagara version 3 features are described below.

3.3.1 Similarity Matching

One additional feature in the third version of Niagara was similarity matching. This feature was intended to address a recurring behavior observed in Niagara where authors would group facts into collections based on the appearance of similar important words in each of the facts. In Figure 38, the similarity indicators can be seen as red overlay text (e.g. “illegal” and “bac”) on the text objects in the workspace. The current fact for which the similarity indicators are being computed is shown with a dark background in the upper right corner of the figure. These similarity indicators are calculated using different linguistic techniques, such as TFIDF, to identify important words in the entire Niagara workspace that also occur in the current fact.

Although these similarity indicators will sometimes be incorrect, having a high threshold for similarity will reduce the chances of inaccurate results. Interestingly, an analogous type of similarity indicator was implemented in the Data Mountain to help subjects organize web favorites [22]. This study suggested that the similarity indicators encouraged a larger number of category creations, longer organization times, and shorter retrieval times.



It is illegal for any person 21 years of age or younger to operate any vessel, aquaplane, water skis, or similar devices with a BAC of 0.01%. You are subject to fine, jail, and participation in an alcohol education or community service program.

It is illegal to drive with a blood alcohol concentration (BAC) that is 0.08% or more.

If you get too many negligent driver points, the department will suspend your license for six months and place you on probation or revoke your driving privilege. You are entitled to a hearing if your request is postmarked or received within ten days of your receiving the order of suspension or revocation.

Each time you are convicted of a moving traffic law violation, the court notifies the DMV. The conviction is placed on your driver license record.

Figure 38: A screenshot of similarity matching cues shown in red. The user can request these cues when moving an item in the workspace or dragging in a new item from one of Niagara's information sources.

3.3.2 Overview and Group Representations

Based on our experiences in Niagara version 2, the spatial overview was replaced with a structural overview (also called a tree or outline view) shown in Figure 39. Since the overview was largely intended to provide access and readability of group labels, the structural overview offered a more useful layout for text labels than a spatial overview. In particular, the structural view had the advantage that it could show an overview of multiple levels for nested hierarchical collections.

Subspaces in previous versions of Niagara were replaced by folders in the structural overview. Folders act like subspaces in Niagara version 2 in that they each have their own infinite 2D workspace. They behave in essentially the same way as folders used in the file browsers on many operating systems. Figure 39 shows a Niagara workspace with a single folder labeled "Root."

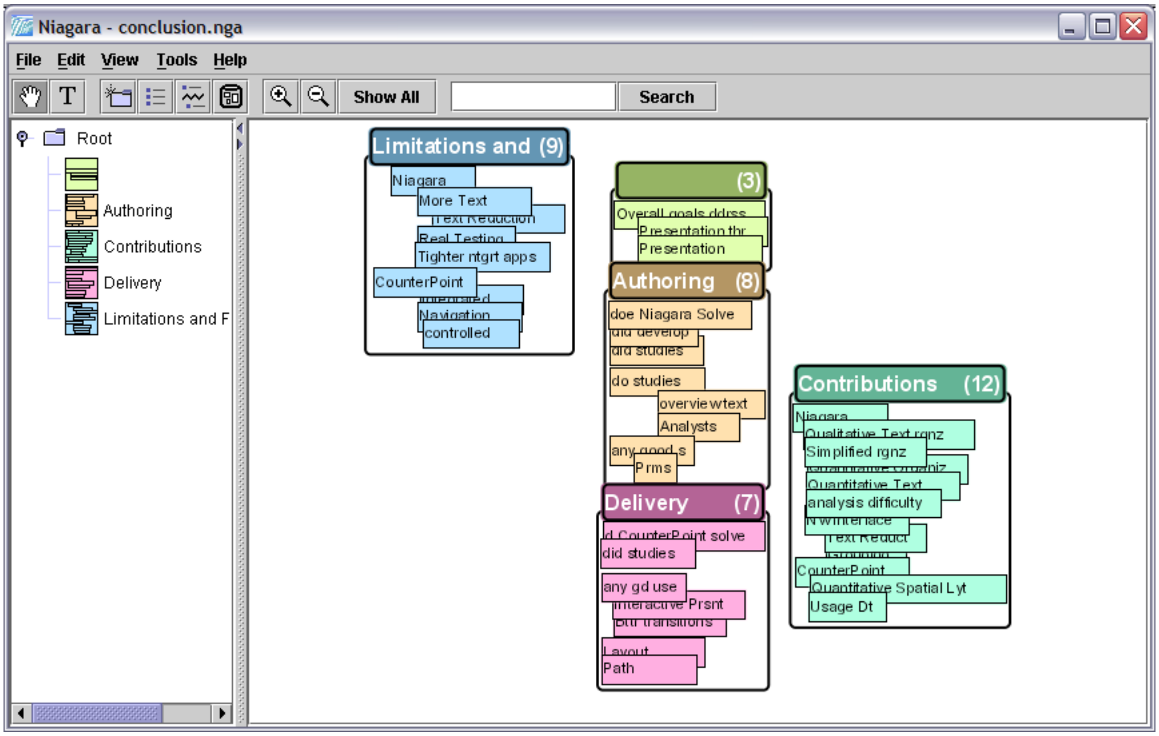
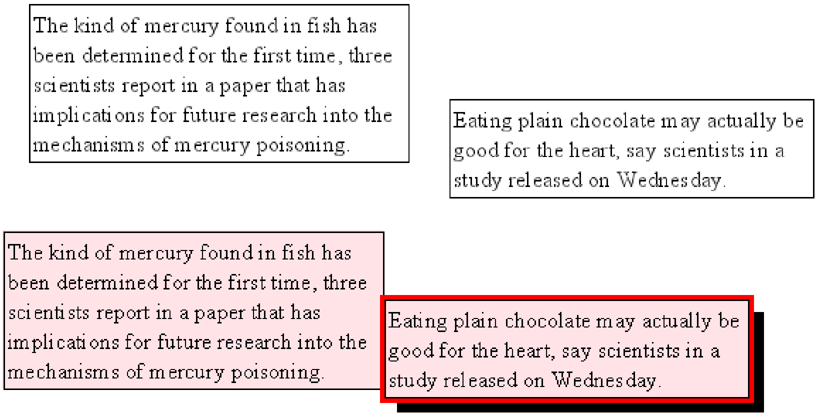


Figure 39: In the structural/folder view, the system represents automatically created groups with a label and an iconic representation of the group's spatial layout.

There was also an interesting crossover between the structural overview and automatic grouping. The automatic groups were represented in the overview with an icon and a text label if one was given. The group icon portrayed a miniature version of the group's spatial layout using the group's automatically assigned color. Several of these icons are portrayed in the overview in Figure 39. The author could double click on these icons to navigate to the group in the main view or drag the icon onto other groups or folders for quick reorganization.

Since the structural overview represented a linearization of a 2D workspace there were also choices in how groups were sorted. Some potential sorting criteria included spatial position from top-left to bottom-right, alphabetical order of group labels, and creation times. Folders could also be separated from or interleaved with automatic group labels. By default, Niagara separated folders and group labels at each level then sorted first alphabetically then by creation time.

In addition, several alternative automatic grouping representations were also considered. The current implementation shown in Figure 39 added a persistent title and rectangular border. However, we also considered variations of an implementation, shown in Figure 40, that only changed the color of grouped objects. Moving the mouse over the group would then invoke a group handle that allowed the author to move the entire group.



The kind of mercury found in fish has been determined for the first time, three scientists report in a paper that has implications for future research into the mechanisms of mercury poisoning.

Eating plain chocolate may actually be good for the heart, say scientists in a study released on Wednesday.

The kind of mercury found in fish has been determined for the first time, three scientists report in a paper that has implications for future research into the mechanisms of mercury poisoning.

Eating plain chocolate may actually be good for the heart, say scientists in a study released on Wednesday.

Figure 40: One implemented version of automatic grouping that changed object colors but did not display a title bar or rectangular outline.

3.3.3 Zooming

As described above, an author could move objects into subspaces for limited control over scale in previous versions of Niagara. However, prior to version 3, Niagara disabled manual zooming of the entire workspace. This restriction was partly due to the lack of good semantic zooming for text objects and also partly to avoid the complexities of independent zooming in multiple nested subspaces.

Niagara version 3 eliminated the previous notion of subspaces while introducing automatic text reduction. These changes facilitated manual zooming of the entire workspace by avoiding the complexities of multiple independent zoom levels and by providing more meaningful views of zoomed out text objects.

A number of different zooming interactions were also considered in Niagara. To avoid many of the navigation problems described in previous ZUI applications, the Niagara implementations completely eliminated unconstrained zooming. Instead, all variants of zoom interactions implemented in Niagara limited zooming to both a minimum and maximum level. It should also be noted that all objects were located at the same baseline scale to further constrain the space and reduce the chances of misplacing objects or getting lost.

Niagara's first zooming interactions were based on those found in PhotoMesa [6] and Nested User Interface Components [86]. This type of interaction computes a discrete number of zoom levels based on the workspace contents that the user can access through clicking. For example, Niagara initially allowed the user to zoom in on an object by left clicking and then progressively zoom out by right clicking.

This approach provided a highly intuitive interface for zooming but did not offer enough control for organizing text. In particular, we found that different users often wanted to zoom out in different amounts to get more room to work. However, users were also often balancing the amount of text they could see in each text object at a particular zoom level, which was highly dependent on the size of the window and the layout of the objects in the workspace. Of course, more discrete zoom levels were added, but this tended to make the technique more inconvenient and less useful.

The second interaction implemented in Niagara tied zooming to the mouse scrollwheel (or middle mouse button). Rather than discrete zoom levels, the scrollwheel allowed the user to rapidly zoom in and out through a series of small increments. While this approach provides more control than the previous one, it raises the issue of defining the point around which zooming would occur. Zooming in around the mouse location is fairly intuitive since you basically just point where you want to go. However, zooming out around the mouse location is fairly unintuitive since the object under the mouse does not move to the center of the window. One possible way to make zooming more intuitive is to modify zooming out to zoom around the center of the view. However, this loses the zooming interaction's idempotent quality, which can also be confusing. The solution currently adopted in Niagara instead is to zoom in and out around the center of the current view. However, future versions of Niagara should probably provide the user with options to control these aspects of the zoom control.

Several redundant zoom controls were also added in Niagara to further simplify zooming. Right clicking on the workspace background would zoom out to see the entire workspace. Right clicking on an object brings up a popup menu with the option to zoom in on the object or zoom out to see the entire workspace. Several toolbar buttons also provided the user with the ability to zoom in and out in increments and to zoom out to see the entire workspace.

3.4 Implementation Details

3.4.1 Bumping

One of Niagara's relatively unique features was its bumping or overlap-avoidance policy. In bumping interactions, the user moves an object so that it overlaps another object or objects and the system must reposition the bumped objects to resolve the overlap. Of course, there are a number of possible ways to resolve the overlap, but the goal in Niagara is to give the interface a physical quality. This means that overlap avoidance schemes like those described by Bell et al. [11] are not suitable since they do not enforce continuous object movements. Their algorithm instead allows bumped objects to jump to the nearest open space to avoid the problem of bumped objects bumping other objects.

Instead, the algorithm needed for the more physical behavior must move objects continuously and iterate from the original bumped objects to the objects bumped by these objects, etc. This helps preserve the illusion of solidity suggested by Change et al. [18]. Our algorithm keeps a queue of bumped objects starting with the initial moved object. Then, while the queue is not empty, the head of the queue is removed and tested for collisions with other objects. If necessary for efficiency, this collision testing can make use of techniques such as interval trees [21, 96]. When collisions are found, the bumped objects are moved to resolve the collision and added to the tail of the queue. The algorithm then loops. As long as objects are convex and collisions move the objects in a consistent horizontal and vertical direction, this process will terminate since it is monotonic. It should also be noted that minor unexpected behaviors can result from discretely sampled mouse motions. This is illustrated in Figure 41 for a diagonal movement. Although it would not be difficult to add, the bumping algorithm in Niagara did not interpolate mouse movements.

Bumping in Niagara also interacted with semantic zooming in an unexpected way. Niagara implemented semantic zooming to enforce a minimum font height for group title bars and one line text objects when viewed zoomed out. Since objects had larger relative sizes when viewed zoomed out, bumping often resulted in large spaces between objects that were only visible when the workspace was viewed zoomed in. For example in Figure 39, if the group labeled “Delivery (7)” were to bump based on it’s current title height, then there would be a space between the “Delivery (7)” title bar and the “Authoring (8)” group when viewed zoomed in. To address this problem, Niagara performs bumping based on an object’s native size rather than its current rendering size.

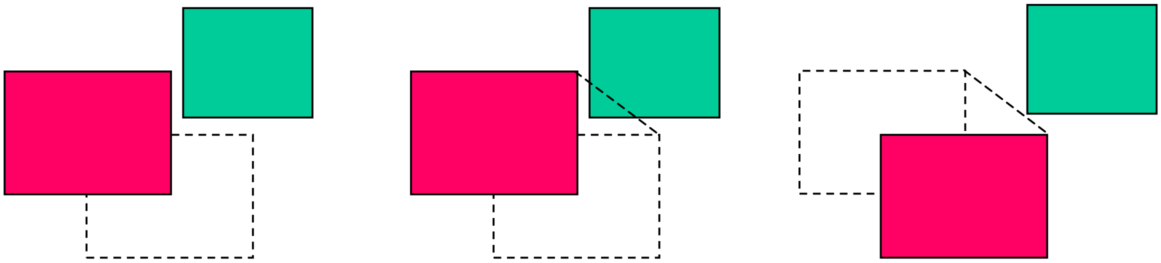


Figure 41: Discrete movements must be interpolated to obtain the ideal bumping behavior. In the movement shown above, the red object does not bump the green object unless the movements are interpolated.

3.4.2 City Lights

An important implementation concern with City Lights is scale. Figure 28 demonstrates the use of orthographic projection for displaying object distance and size in the first implementation of City Lights. In this approach, objects in the “corners” of populated space receive a disproportionately smaller space for conveying their information. The corners are akin to “blind spots” in driving a car – that zone where the use of mirrors gives little or no coverage for viewing other automobiles.

How does the number of objects in the blind spot change with scale? As shown in Figure 42, as the ratio of the size of populated space to viewed space increases, the percentage of objects in the blind spot increases. For square windows where the bounding side of populated space is three times the length of the viewed space, already half of the objects in the populated space are in the blind spots. The use of a radial rather than orthogonal projection addresses this issue by eliminating the blind spot. Consequently, the radial projection performs more consistently when the populated space is much larger than the viewed space.

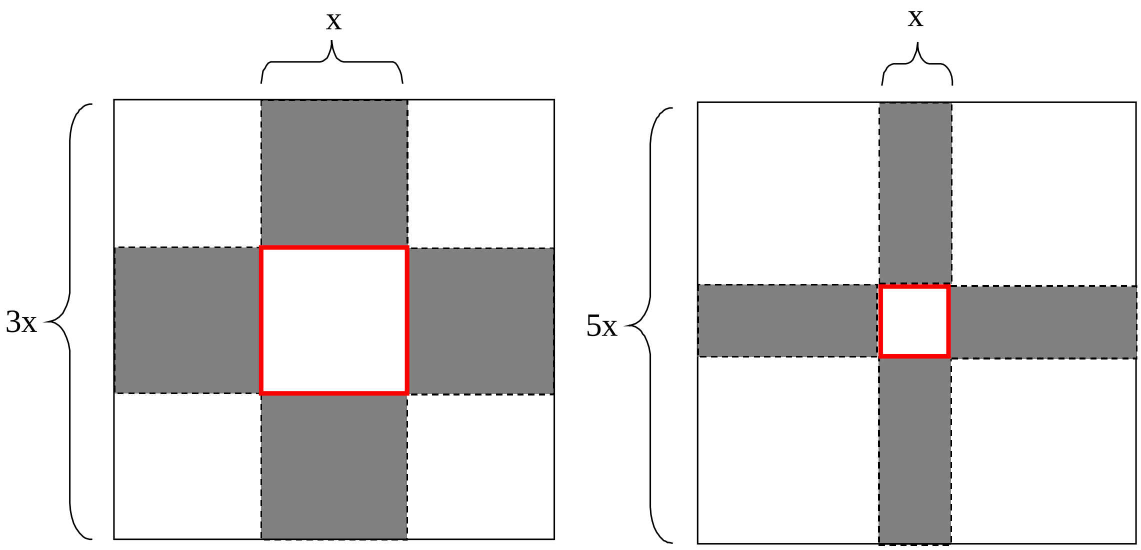


Figure 42: The red rectangle in both drawings represents the viewed space and the outer rectangle represents the populated space. The shaded areas represent the set of objects projected onto the window borders by an orthogonal projection. As the ratio of populated space to viewed space increases the percentage of the objects in the orthogonal projection's blind spots also increases, as can be seen by the growing corner rectangles above.

3.4.3 Automatic Text Reduction

Automatic text reduction, described in Section 2.5, makes use of a text reducing function to generate different length texts based on the given size requirement. Niagara's automatic text reduction focuses on condensing small collections of text snippets or paragraphs rather than larger texts. This violates the assumptions of many traditional linguistic techniques [67]. In addition, Niagara's text reduction has a different goal than many linguistic techniques. Since users are familiar with the textual content, the purpose of text reduction is primarily to allow users to recognize text elements they created or selected earlier. Therefore the reduced text need not carry full meaning or be comprehensible and strictly correct.

We have informally experimented with combinations of several simple text reducing functions including rankings based on: universal word frequencies, word length, word position, syntactic role, and TFIDF. A common trend we found in several of these rankings is that they eliminate short words. This seems to be a result of these words' frequent occurrence, their appearance in stop word lists, and their use in limited syntactic roles. Because of these similar results, we found that many of these reduction techniques work sufficiently well for our purposes. Further user testing is needed to determine whether Niagara requires more complex language models for optimal text reduction.

The current implementation also reduces text length by shortening individual words. The techniques used to shorten words include stemming, removing vowels, and truncation. In the current implementation, these techniques are only employed when complete, unmodified words will not fit in the remaining available space. Of course, future implementations could use more sophisticated models for combining word elimination and intra-word reductions.

The current text reduction implementation follows the general pseudocode procedure below:

```
Initialize PrevWordList to contain all words in full text
For each level of reduction
    Initialize LevelWordList as an empty list
    Compute available space at current level
    Rank words in PrevWordList using a combination of TFIDF and global
        word frequencies
    For each word in PrevWordList
```

```
          If space available, respecting original text order, and
              LevelWordList does not already contain word
                    Add word to LevelWordList
          If word added
                    Remove word from PrevWordList
For each word in PrevWordList
          If space available, respecting original text order, and
              LevelWordList does not already contain word
                    Add stemmed word to LevelWordList
          If stemmed word added
                    Remove word from PrevWordList
For each word in PrevWordList
          If space available respecting original text order, and
              LevelWordList does not already contain word
                    Add word with vowels removed to LevelWordList
          If vowel-removed word added
                    Remove word from PrevWordList
If LevelWordList is emtpy
          Truncate top-ranked word to fit available space
          Add word to LevelWordList
Set PrevWordList equal to LevelWordList
```

Automatic text reduction introduces a tradeoff between font size and content length. As the font size is reduced, more content can be shown in an object. However, reading the text at the reduced size becomes correspondingly more difficult. Niagara chooses a balance between these two, shown in Figure 43, that alternates between reducing font size and reducing text length. A subtle problem that arises in applying this alternating strategy across objects is that it can lead to different font sizes for different objects. This results in a patchwork display of font sizes that is aesthetically unappealing and even potentially distracting. Niagara deals with this problem by coordinating the two kinds of reduction across objects at predefined zoom levels. For example, Niagara implements content reductions on all objects when the scale is a multiple of 0.84 (or $0.5^{0.25}$). At these points of content reduction, the font size is reset to maximum and then gradually decreases with scale until the next content reduction point. Other applications using automatic text reduction will likely need a similar coordination, though they can choose different transitions through the graph in Figure 43 based on the requirements of the task.

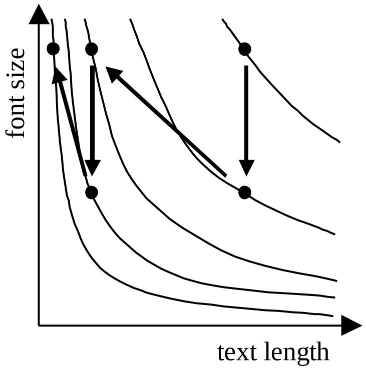


Figure 43: A graph of text reduction in terms of font size and text length. The black dots indicate the states represented in Figure 6. The curves indicate objects of the same size.

It should be noted that the curves in Figure 43 are only approximate. In actuality, text cannot be smoothly reduced but instead must be reduced in discrete increments due to letter, word, and line boundaries. Discrete line boundaries are particularly limiting since each line can contain a potentially unlimited amount of text. As a result, each line that is eliminated in automatic text reduction can lead to a significant reduction in the quality of the text. The number of lines that can be shown in a text object is highly dependent on the font size and aspect ratio of the object. For example, Figure 44 demonstrates how two text boxes with equivalent area, line spacing, insets, and font sizes can show different numbers of lines and hold different amounts of text.

The size of the text object itself also has a strong effect on automatic text reduction. Just as with non-text objects in zoomable space, smaller text objects become less recognizable than bigger text objects the further you zoom out. However, this is intrinsic in the nature of zoomable space so it is not necessarily something that should be fixed. That being said, the current implementation does try to delay this effect in Niagara. In particular, Niagara implements a one line minimum height on all text objects. Figure 39 shows a zoomed out Niagara workspace with a number of single line text objects that have grown beyond their original size to enforce the minimum height. The figure also illustrates the primary tradeoff to this approach, namely that objects begin to overlap the more you zoom out. However, without this approach, font size would be greatly reduced leaving essentially none of the text readable.

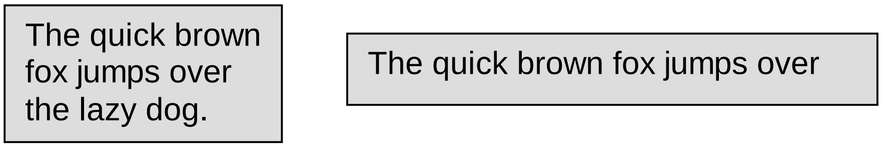


Figure 44: Two text boxes with equal area and equal font size. The right text box holds less text because its height is between multiples of the font height.

In an interactive application like Niagara, speed is a primary concern in implementing automatic text reduction. Niagara often has many text objects, where each text object has many levels of reduction. This makes it difficult to compute text reduction on the fly for each object during interactive zooming. Instead, it is often necessary to pre-compute and cache all necessary text reduction levels as each text object is created and edited. Fortunately, the need for caching text reduction levels integrates nicely with the need to coordinate font sizes across objects. In order to coordinate font size in Niagara, the font is reduced until the scale reaches a power of a given ratio *r*, then the content is reduced and the font returns to its original size. The text lengths of these cached levels form a geometric series with a ratio of *r*, so we can predict that the total memory required to cache these levels will be less than 1/(1-*r*) times the memory of the original text. In Niagara, *r* is set to 0.84 so that text objects with a 12 point font will not fall below a 10 point font. This value of *r* means the memory is limited to 6.25 times the original text. In practice, the actual memory requirement is much lower since we implement a minimum text length rather than continuing the geometric series to infinity. In addition, other memory saving techniques can be used such as using text references rather than text copies for the cached levels to limit the memory requirements of caching even further.

Chapter 4:
Presentation Delivery and CounterPoint

4.1 Related Work

The work in this chapter builds heavily on the slide show metaphor. As a result, it also further develops many of the ideas found in the numerous software slide presentation tools including Corel Presentations [91], Freelance Graphics [34], Hancom Presenter [47], Harvard Graphics [48], Impress [57], Keynote [63], KPresenter [64], Persuasion [87], and PowerPoint [89]. These commercial tools primarily provide a software interface for mimicking physical presentation media such as 35mm slides or overhead transparencies. CounterPoint, a ZUI presentation tool described in this chapter, extends the techniques and metaphors employed in these tools.

4.1.1 Navigating Presentations

Because of their demanding conditions, slide show presentations have long made use of extremely simple navigation controls. The most basic slide show presentations provide only two navigation controls, namely moving forward one slide and backward one slide. These simple techniques work fine until the presenter wants to jump out of their linear sequence. At that point, more advanced controls are needed.

One extension to this purely linear traversal through a collection of information was Zellweger's Scripted Documents [119, 120]. Scripted Documents allowed the author to define timed traversals through a collection of documents with specifiable actions performed at each stop in the traversal. Conditional and customizable paths were also described that allowed paths to be modified to fit the users current needs. The

‘Audio-visual presentation’ application of scripts described in Zellweger’s earlier work closely resembles the use of scripted paths described in this chapter.

Trigg's Guided Tours and Tabletops also described a more dynamic version of linear paths [112]. The system described in the paper provided tools for creating a collection of "tabletops," each of which contained a spatial arrangement of notes or documents. An author could then define arbitrary paths through these tabletops with any number of available branches at each point in the path. Here again, the scripted paths described in this chapter resemble Trigg's paths in that they can combine both scripted and dynamic components. Both also follow the traversal defined by the scripted path unless special actions are taken. However, CounterPoint paths differ from these tours in that they traverse data in a single continuous space instead of sets of disjoint spatial arrangements.

World Wide Web-style hyperlinking also provides an interactive extension to traditional linear paths. These hyperlinks can be found as a navigation aid in several commercial presentation tools including PowerPoint [89]. One practical application of hyperlinked paths in the presentation setting was Moore’s use in teaching an undergraduate Computer Science course [73]. Moore found that the use of this kind of traditional hypertext facilitated hierarchical organization and also allowed for interconnection of related material, both of which potentially improved navigation. However, hyperlinks have the drawback that they require an author to create them prior to giving the presentation. As a result, the presenter must anticipate all potential branches that might be required during a presentation.

Another tool suggested for improving navigation in slide presentations is Hyper Mochi Sheet [111]. Hyper Mochi Sheet employs a multi-focus distortion-oriented view to display a hypertext network. During a presentation, the system automatically resizes nodes in the network based on the presenter's current focus. While the multi-focus views allow it to show focus and context, its dynamic nature makes it harder for the presenter to predict. Thus it is often less desirable for presentations where layouts and object sizes are parameters of primary concern.

Several recent techniques address presentation navigation through extensions to the existing slide show metaphor. The Palette system [75] allows presenters to navigate a standard slide presentation using a barcode reader with paper copies of slides. The barcode on the paper slide acted as a reference to the PowerPoint file and slide number of its corresponding virtual slide. This allows presenters to dynamically combine slides from multiple PowerPoint files into a single presentation and to improvise slide orderings at presentation time. The paper slides also allowed presenters to search for a particular slide by flipping through a physical stack or spread the slides out on the desk or podium. Of course, this approach also has drawbacks. In particular, it requires users to manage both the physical and virtual representations of their slides.

In contrast, Dieberger et al. describe a pure software technique for presentation navigation [27]. Their technique uses a visualization along the left edge of the current slide to facilitate movement within the presentation. This visualization also displays information about which slides have been visited and various other timing statistics.

4.1.2 Navigating in ZUIs

Simplified interaction techniques are more important for slide presentation tools than traditional desktop applications since strict time limitations, concerns about appearance, and the potential for audience confusion often increase the cost of errors. In this regard, it would seem that ZUIs are not a natural fit for the presentation setting since they are notoriously difficult to control. Navigation and spatial awareness have been noted as particular problems in ZUIs [36, 61]. Nevertheless, recent advances in ZUI navigation have helped alleviate these problems in many cases.

Navigation in 2D workspaces has traditionally been achieved through panning and zooming or dedicated widgets such as scrollbars or overviews. However, these techniques do not take into consideration the specific challenge of ZUIs that information can reside at multiple zoom levels. One particular problem using traditional techniques in large multiscale spaces is disorientation. Furnas et al. propose the space scale editor as a technique to address disorientation in multiscale ZUI spaces [37]. The Space Scale Editor is an additional view that augments a traditional 2D workspace with a visualization of multiple zoom levels. This view offers awareness of object positions at different zoom levels while displaying a visual representation of the primary view's position in the multiscale space. An alternate technique proposed by Jul et al. addresses disorientation through visual cues integrated into the primary 2D workspace [61]. These visual cues enhance the standard workspace view by indicating interesting regions containing objects too small to be seen at the current zoom level. CounterPoint tries to avoid the problems addressed by these techniques simply by limiting the range of scales at which objects can reside and the amount the user can zoom in or out.

A second problem with traditional ZUI navigation techniques is that manual panning and zooming often require a lot of attention and coordination. One technique used to reduce the attention and coordination needed in large documents is speed dependent automatic zooming [56]. This approach automatically combines zooming and rate-based scrolling to maintain a constant visual information flow as the scrolling velocity varies. An even more constrained approach is used in PhotoMesa to simplify navigation in a 2D layout of photos [6]. Here the user primarily navigates through animated step-wise zooming; left clicking to zoom in and right clicking to zoom out. A similar type of simplified navigation was also employed in Nested User Interface Components [86]. A version of this kind of step-wise zooming has been implemented for navigating in CounterPoint presentations.

4.1.3 ZUI Narratives

Presentations tell a kind of story. As such, the work in this chapter builds on previous work in using ZUIs for narratives. One early exploration in using ZUIs for narrative was a work titled Gray Matters [114]. Gray Matters laid out pictures and text in a Pad++ workspace that a user could then navigate through panning, zooming, and animated hyperlinks. This work was primarily artistic in nature, hence no evaluations were performed.

Using ZUIs for narrative has also been explored in the context of a children's story-authoring tool called KidPad [30]. Kidpad provides simplified tools to allow children to create primarily vector based drawings that can be linked together in a zoomable workspace. These links then support the child in dynamically navigating the workspace to tell different stories based on the illustrations. Boltman et al. looked at these kinds of ZUI stories in a study of 72 children [13]. The study involved three versions of a wordless children's book included a paper-based version, a traditional computer-based hyperlinked version, and a computer-based ZUI version with animated panning, zooming, and fading transitions between images. The study used a between-subjects design and found that the ZUI stories provided significant improvements over the other two conditions in different elements of elaboration and recall.

ZUIs were also studied for readers of nonfiction. Paez et al. looked at using ZUIs for a document reading task [80]. They performed a between-subjects experiment with 36 participants comparing a document laid out in a ZUI workspace to a hyperlinked document in a traditional web browser. The task in the study required the user to find the answer to 5 questions within the document using one of the two interface types. They found no significant differences between interface types in completion times, comprehension measures, disorientation measures, or subjective satisfaction. Subjects in the study did report that the ZUI provided good skimming and overview capabilities and was easy to learn. This study is particularly interesting for the current research since the interfaces are fairly similar to those discussed in this chapter. In particular, the hyperlinked document is similar to a traditional slide show presentation and the ZUI document is similar to a ZUI presentation. Of course, a slide presentation is different on several levels from document reading task, yet we expect that ZUI presentations will have many of the same properties suggested in this work.

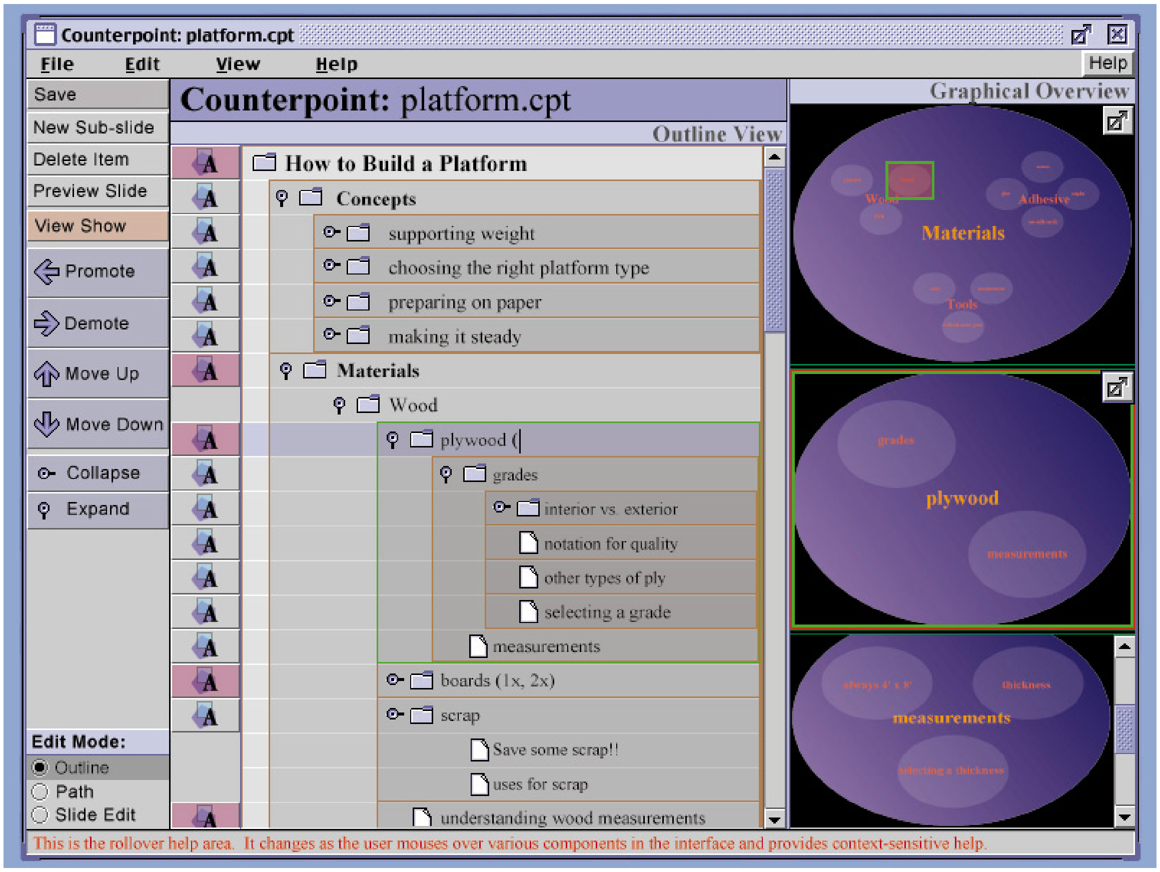
4.2 CounterPoint

Although ZUIs are a relatively new, a number of ZUI presentations have been created prior to this work. These presentations were originally created and delivered using Paddraw [9], a general purpose ZUI authoring tool built on top of Pad++. More recently, these presentations were created using Hinote [10], another general purpose authoring tool built on top of Jazz. Creating presentations with these tools can be likened to creating slides for a traditional slide show with a general purpose drawing program. The tool provides tremendous freedom and all the necessary functionality, but it does not offer any specific shortcuts for common presentation tasks. This trade-off in specificity versus generality has actually been explored for slide authoring with general drawing tools versus specific slide authoring software [59]. The authors identified a number of tradeoffs between the two techniques but suggested that the presentation-specific tools are likely to become more desirable as the number of authored presentations increases. Many of the tradeoffs described in this work are likely to apply to ZUI presentation authoring tools as well.

This work comes out of many years of ongoing research into ZUIs and their actual use for presentations. As such, it builds on the experiences described above in authoring and delivering presentations with existing ZUI authoring tools. Based on these experiences, this chapter introduces a presentation-specific ZUI authoring tool called CounterPoint that is primarily intended to address the complexities involved in authoring ZUI presentations. This chapter also provides evidence for the various benefits and limitations in using ZUIs for presentation. Additional information can also be found in our published work on CounterPoint [42].

4.2.1 Early Mockups

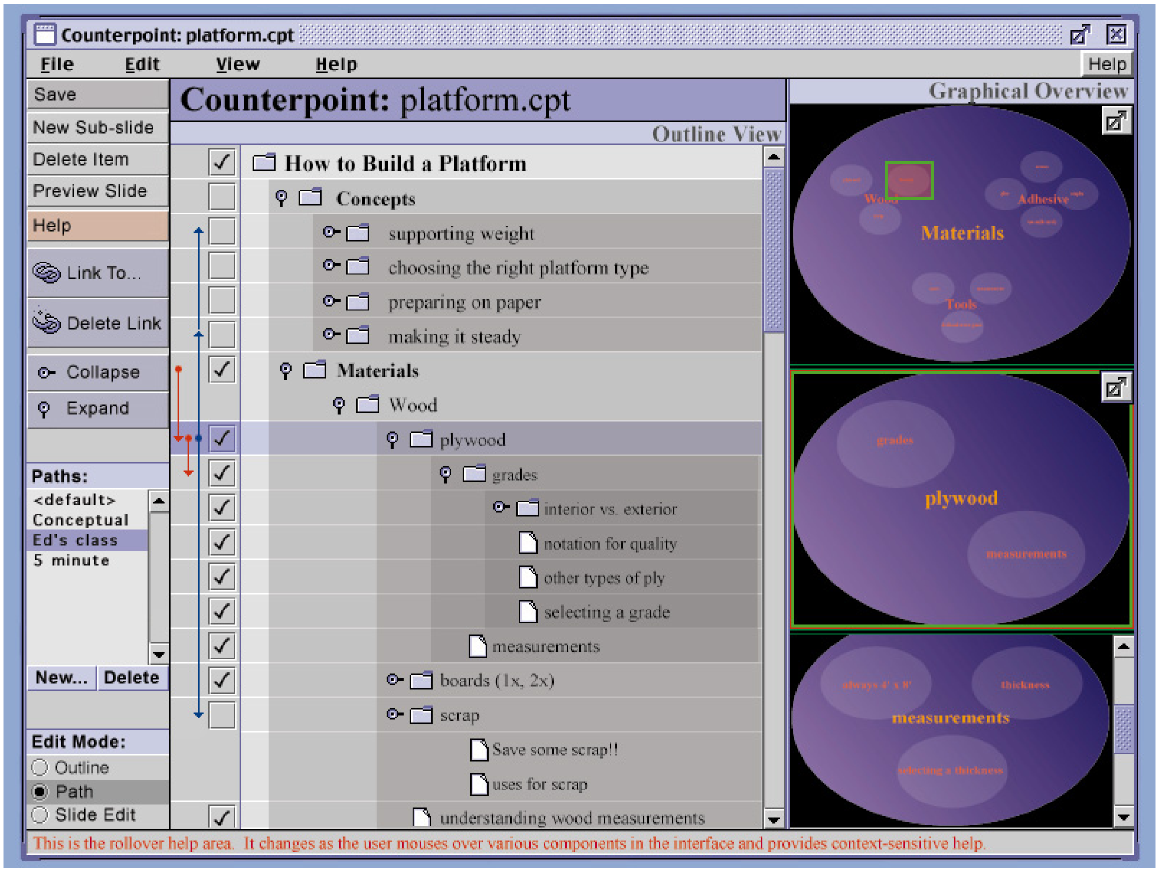
Prior to actually implementing CounterPoint, David Feldman (www.interfacethis.com) created a specification and several mockups describing the expected functionality in CounterPoint. The mockups portray a hierarchical editing mode for authoring textual presentation content (Figure 45) and a path editing mode for authoring traversals through the content hierarchy (Figure 46).



This is the rollover help area. It changes as the user mouses over various components in the interface and provides context-sensitive help.

Figure 45: A mockup of a hierarchical editor for authoring textual presentation content.

As previously mentioned, text authoring tasks often organize information using a hierarchical structure. These mockups are therefore a natural starting point for CounterPoint since they are designed around hierarchical editing tools. They were also a natural extension of previous work in ZUIs, such as the Pad++ directory browser [9], that mapped levels of a hierarchy to zoom levels in the zoomable workspace.



This is the rollover help area. It changes as the user mouses over various components in the interface and provides context-sensitive help.

Figure 46: A mockup of a path editor for authoring paths through the content hierarchy.

4.2.2 A Functional Prototype

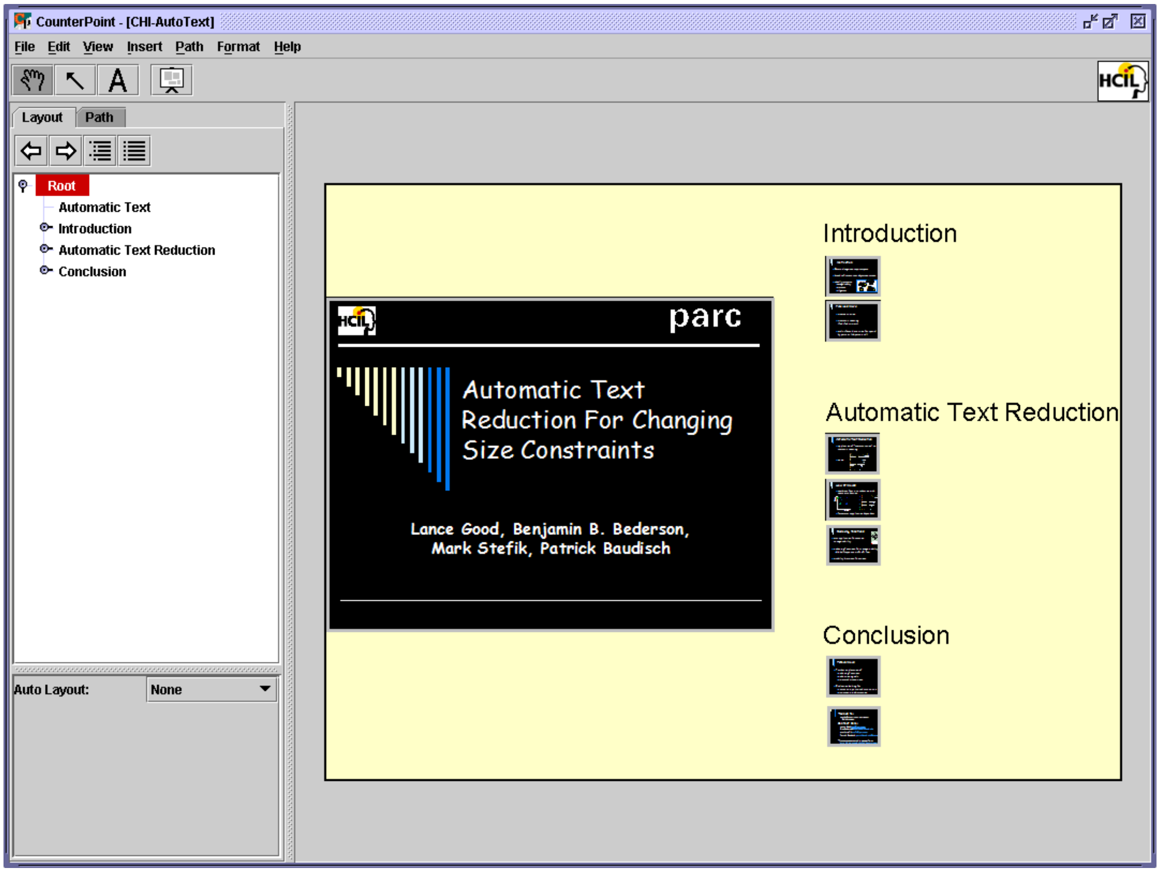
The hierarchies shown in the previous mockups had an intuitive appeal for presentation authoring since all ideas were given equal representation. In particular, all ideas had an icon and a label. This allowed them to be displayed using existing tree representations and interactions. However, in practice presentations are often not purely hierarchical at the level of individual ideas or bullets. For example, in existing presentation software, presenters often create an introductory slide containing overview information for a topic and then additional slides containing the details. In this case, the overview information on the introductory slide is not a subcomponent of the topic in the same sense as the detail slides.

This suggests that a hierarchical decomposition at the level of individual bullets and ideas is the wrong granularity for presentation slide authoring. Instead, presenters tend to think about their presentations in terms of view-sized chunks or slides. This allows them to maximize the value of individual views and limit the number of transitions needed between views. Ultimately, this strategy minimizes the amount of attention needed to control the presentation tool as well as any distractions caused by the superfluous transitions.

An additional practical benefit of using slides or discrete views as the unit of information in ZUI presentations is that authoring tools can leverage users' existing presentation authoring skills. Because the notion of slides remains essentially the same, ZUI presentation authoring environments can provide similar objects and operations as in traditional slide authoring tools.

Based on these early lessons, a CounterPoint prototype was implemented, shown in Figure 47, that provides a hierarchical decomposition using slides as the level of granularity. The prototype also integrates with PowerPoint, an existing slide authoring tool. This allows the tool to take full advantage of presenters’ existing slide authoring experience since many presenters are already using PowerPoint for presentations. In addition, it also allows presenters to reuse existing slides that have been authored and saved in PowerPoint’s specific format.

Much of the benefit of CounterPoint’s ZUI interface for presentations comes from having structure on top of the slides. As a result, CounterPoint requires additional effort beyond individual slide authoring to combine these slides into a single zoomable structure. While creating arbitrary spatial layouts can be difficult in a zooming environment, this process can be simplified by separating out the conceptual structure from the spatial structure. This allows the conceptual structure, which is often well defined, to automatically determine the spatial structure, which can be more subjectively or aesthetically defined. Of course, the automatically defined spatial structure can also act as a starting point, allowing the presenter to tweak the spatial layout to emphasize certain ideas, improve memorability, or customize the aesthetics.



Auto Layout: None

Automatic Text Reduction For Changing Size Constraints

Lance Good, Benjamin B. Bederson, Mark Stefik, Patrick Baudisch

Figure 47: A screenshot of the CounterPoint interface in the hierarchical layout editing mode for a presentation on Automatic Text Reduction [43].

As the mockup in Figure 46 suggests, CounterPoint also supports multiple paths through a single presentation hierarchy. As a result, an additional authoring challenge is creating and editing these paths. The automated approach to defining these paths is to use the presentation's structure. In this case, the spatial layout can be used to define a default ordering of slides, such as a depth first search. In addition to spatial order, the conceptual structure can be used to inform the ordering based on group membership and to define reasonable overviews of groups within the path. As in authoring layouts, this default path can also serve as a starting point for more customized paths.

In addition to helping layout the space and define a path, the hierarchical structure defined in CounterPoint can assist the presenter in navigating the presentation when answering questions or responding to unexpected time constraints. For example, in jumping from a slide in the conclusion to a slide on an earlier topic, the presenter can simply move up the conceptual hierarchy to find the high level representative for the topic then drill down in that topic to find the appropriate detail slide. Restated, the conceptual hierarchy facilitates a search with logarithmic complexity. Because of these efficient navigations, scripted paths may become less important in CounterPoint as presenters begin to rely on the customized navigation controls available in presentation mode. Still, some presenters are likely to rely on scripted paths because of the reduced chances of error.

More details on the CounterPoint features and implementation can be found in Chapter 5, titled “CounterPoint Implementation and Features.”

4.3 The Audience’ Perspective

Previous research in HCI, hypertext, cognitive psychology, and educational psychology suggests mechanisms by which CounterPoint presentations may offer cognitive advantages to the audience over traditional slide show presentations. In many of the studies below, the conditions and tasks used differ from the presentation setting.

However, there are also enough similarities to justify some level of comparison. Moreover, informal experience with CounterPoint presentations offered some evidence that the audience benefits from these presentations.

Of course, individual differences undoubtedly influence the cognitive utility of ZUI presentations for the audience. Indeed, animations [50] and spatial arrangements [1], two elements of ZUIs, have been shown to support greater recall in users with low spatial abilities than in users with high spatial abilities. Fortunately, these studies give some further evidence that animations and spatial arrangements do not negatively impact recall in users with high spatial abilities. Likewise, research into concept maps (see Figure 48) suggests that subjects with low verbal ability may benefit more from a spatial display than from a textual display [83]. Here again the type of display did not adversely affect the high verbal ability students. With the caveats in mind, listed below are some potential cognitive benefits of ZUI presentations and supporting evidence from related work.

Following a discussion of the potential benefits, a preliminary user study is presented to explore the extent to which these benefits appear for an actual slide presentation.

4.3.1 Potential Cognitive Benefits

4.3.1.1 Dual Encoding in Memory

Presentation tools are typically accompanied by a presenter's oral discourse. Hence, the audience receives simultaneous visual input from the presentation tool and verbal input from the presenter. Therefore, an interesting question is whether humans learn differently from these two streams of data. Cognitive hypotheses suggest that human memory does encode spatial information distinctly from verbal information (for example, see [2, 81] and discussions in [1, 95]). As a result, a presentation tool may exercise a larger portion of the memory resources of the audience if it employs a spatial, visual display in combination with the verbal discourse.

Robinson et al. performed research into this phenomenon by comparing graphic organizers and concept maps with linear lists and outlines [94, 95]. Graphic organizers are simply graphical layouts of information, for example tables and flowcharts. This research suggests that the information in the graphic organizers and content maps is encoded more spatially than the information in linear lists and outlines. In an earlier study, Robinson et al. also investigated the benefits of adjunct displays in a setting more comparable to that of presentations [94]. Subjects were shown different visual displays while a related text was presented aurally. As in the other study, graphic organizers and concept maps facilitated a more spatial encoding of the information than the textual displays.

The spatial organization of data in ZUIs, though unconstrained, lends itself to structures similar to graphic organizers and concept maps. As a result, ZUI presentations likely allow for spatial memory encoding of the presentation layout. Combining this spatial data with the conceptual encoding of the oral discourse may both reinforce the conceptual structure and help reduce the audience' verbal load, ultimately increasing the retention of certain presentation data.

One specific instance of this dual memory encoding is that, in certain situations, the memory for data and the spatial location of that data are correlated (summarized in [28]). For presentations, this implies that more memorable spatial layouts may improve the memorability of the underlying presentation content. As a result, the potential advantage of CounterPoint presentations over traditional slide shows is the ability to spatially organize presentation structure in two dimensions at different magnifications. This spatial layout may provide the audience with a strengthened memory pathway with which to recall the presentation content.

4.3.1.2 Improved Overview Support

ZUIs naturally support overviews by providing the capability to zoom out for more context. Good overviews may also benefit the audience by increasing the memorability and possibly the comprehensibility of high-level presentation concepts. In a hypertext reading study, spatial, hierarchical overviews of hypertext networks were demonstrated to improve recall of overview titles when compared to both hypermedia with alphabetical linear overviews and hypermedia without overviews [24]. Similarly, overviews have been shown to improve the understanding of concept maps (see Figure 48) over several disconnected views in subjects with low spatial abilities [116].

Overviews in CounterPoint presentations also reveal spatial landmarks in the presentation layout. Landmarks have been identified as a fundamental component in the development or orientation knowledge [28]. Landmarks indicate a position in the global surroundings based on salient or memorable objects in the local surroundings. Consequently, CounterPoint overviews may also serve the purpose of helping the presenter and audience stay oriented during a presentation.

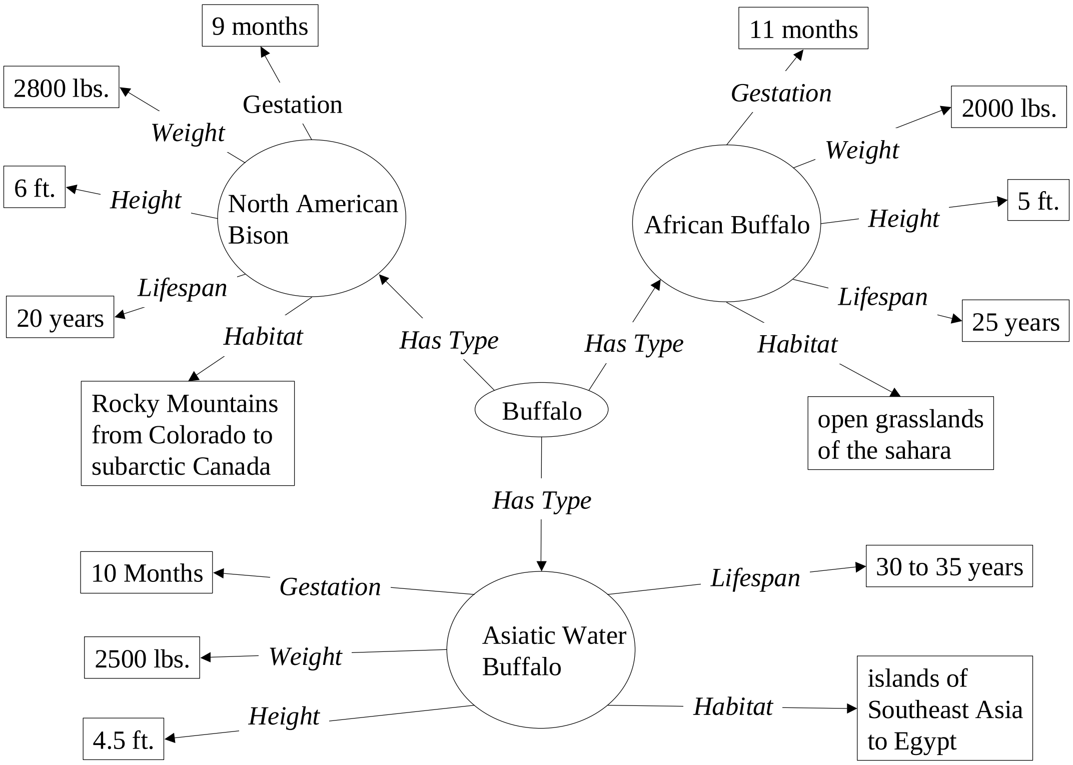


Figure 48: An example of a concept map taken from Robinson et al. [95].

4.3.1.3 Structural Visibility

CounterPoint presentations allow the logical organization of the presentation to be incorporated into the spatial layout of the data. Then, because the transitions between viewpoints in CounterPoint presentations animate through the space, they reveal this logical structure to the audience through the normal course of the presentation. Designers have recognized the benefits of portraying the structure of an object to its users in this manner. Norman describes the principle of revealing the structure of an object in its design as "visibility" [77]. Similarly, Thüring et al. suggest that presenting a hypertext document's structure to the audience is a necessary component "for reducing the mental effort of comprehension" [109]. A similar sentiment is offered by Rivlin et al. for improving hypertext navigation [92].

Structural visibility can also be important when a presentation shifts from one topic to another. In this case, two adjacent slides in the presentation sequence may contain no semantic relationship. In the slide show metaphor, the presenter was responsible for orienting the audience to this kind of topic change. In contrast, CounterPoint presentations can indicate the semantic difference between two slides by the visual distance in the presentations spatial layout. This spatial separation can then be indicated to the audience through the overviews and spatial animations. A similar example of semantic distance explored in hypermedia is the "warp coefficient" suggested by Kaplan and Moulthrop [62]. Here a number is associated with each link on a hypermedia page to indicate the semantic difference between the content of the current page and the link's destination page.

4.3.1.4 Incrementally Revealing Content

One problem mentioned in the use of concept maps, a specific type of spatial display shown in Figure 48, is map shock [12]. Map shock occurs when map-readers feel overwhelmed, confused, or unmotivated by the size or complexity of a map. To solve this problem, map content can be incrementally revealed to help make information appear less intimidating.

An early technique used for incrementally revealing content is the stacked map [116]. Stacked maps divide a large or complex concept map into smaller cross-referenced maps. One study compared these stacked maps to un-segmented whole maps to determine if they improve comprehension of map content [116]. The results of this study suggest that the utility of different map formats may depend on subjects' individual differences. Subjects with low spatial abilities performed better with whole maps while subjects with high spatial abilities performed better with stacked maps.

A more recent technique, applicable to computer-based displays, that tries to address some of the limitations of stacked maps is using animation to incrementally reveal map content. Here, the map presents a subset of the map content and incrementally animates more details into the display. A recent study compared these animated maps against plain text, animated text, and static maps [12]. The results of this study indicate that animated maps may facilitate better recall of high-level points than the other three displays.

Because they use viewpoint animations, CounterPoint presentations can similarly mediate between stacked and whole map displays by incrementally revealing content. Meaningful chunks of presentation content can be arranged at different spatial locations to achieve an effect similar to disjoint, stacked concept maps. Then, spatial animations can be used to navigate between these disjoint maps in the 2D space. And because they support zooming, CounterPoint presentations can display overviews of the collection of stacked maps to also support whole-map displays.

4.3.1.5 Animated Slide Transitions

Although animated transitions are available in many slide show based presentation tools, these transitions are mainly used for visual effect and usually do not attempt to give any insight into the underlying data. CounterPoint presentations implement slide transitions as animated viewpoint navigations through the presentation space. As such, these animations are able to display the changing spatial context as the system transitions from one point in the 2D space to another. Although the actual benefits of viewpoint animation still require further investigation, initial research indicates that these animations are beneficial for learning spatial organizations and data relations [7]. This study further suggests that viewpoint animations allow for a more constant understanding of object positions and relationships than viewpoint transitions without animation. Research also indicates that animation may improve long-term understanding of various kinds of presented material [50, 65]. This improvement was most profoundly observed in those with low spatial abilities [50].

One of the biggest risks associated with animations is the time consumed by presenting extra intermediate frames during a transition. However, research also indicates that the extra time spent on animation does not result in longer task completion times [7], which relates directly to comprehension time.

Of course, animated transitions also pose more subjective risks as well. For instance, some users may find these animations distracting or otherwise undesirable. However, there is some evidence to suggest a subjective preference for animated systems over non-animated systems [29]. Although user preference is a recognized quantitative measure of software usability [105], these preferences require further study to determine their importance, both in general and specifically for presentations.

The animated transitions in CounterPoint presentations may also benefit from additional dual-encoding redundancy. Research has shown that animations accompanied by explanatory audio can improve understanding of abstract concepts over static graphics combined with explanatory audio [65]. This animation-audio combination also led to better long-term retention than its static counterpart [65].

It is also important to note that there are a number of different types of animations (see [7] for a partial listing) that can make comparisons difficult across tasks. Similarly, whether a system's animation is automated or manually controlled can affect its cognitive utility. Therefore, these differences must be considered when comparing the results from other animated systems to the ZUI presentation setting.

4.3.2 An Audience Study

Based on this previous research and our own experience, we hypothesize that CounterPoint presentations offer advantages to the audience for recalling information from the presentation. In particular, we anticipated that these presentations would help the audience remember the main ideas and the high level structure.

In designing a study to test this hypothesis, it was noted that CounterPoint presentations differentiate themselves from the traditional slide show metaphor through the two features of spatial layout and animation. The spatial layout is distinguished from spatial layouts on individual slides in that all of the information on all of the slides is positioned in a single continuous space. Animations are then used in moving between points in this space to visually reinforce the sense of physical motion. Consequently, these features were evaluated in the user study below, comparing their effects on subjective preferences and perceptions as well as recall for content and structure.

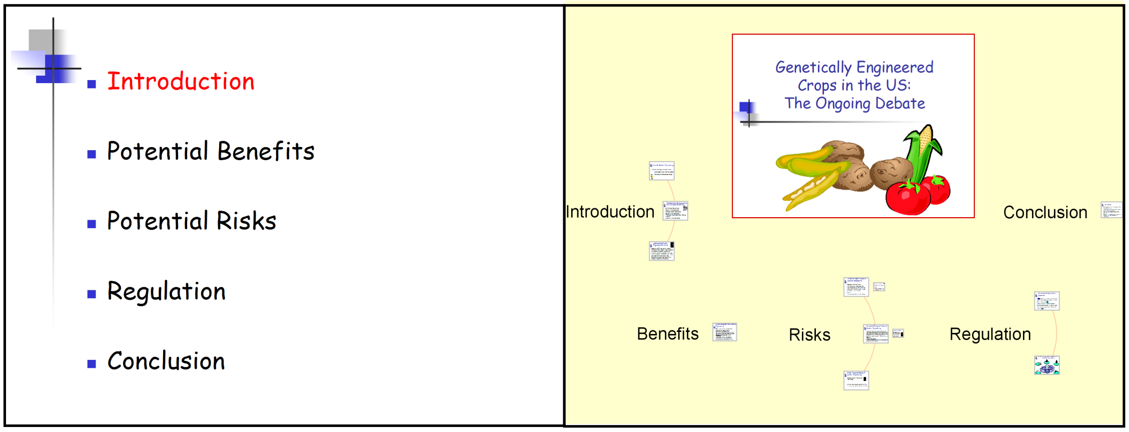
4.3.2.1 Method

Participants in the study consisted of 96 undergraduate and graduate students in the University of Maryland computer science department. Seventy-eight of the students were male and 18 were female. Participants were all enrolled in one of three upper level undergraduate Human-Computer Interaction courses. Each of the 3 classes was assigned to one of the following 3 conditions: 1) a traditional slide show, 2) a traditional slide show plus spatial layouts and overviews, and 3) a traditional slide show plus spatial layouts, overviews, and animations between spatial positions. These conditions are shown in Table 7 below:

Table 7: Conditions explored in the study. Animation and no spatial layout were not explored since the animations were not expected to contribute to non-spatial presentations.

	No Animation	**Animation**
No Spatial Layout	Standard PowerPoint	These animations indicate only forwards or backwards progress
Spatial Layout	CounterPoint without animated transitions	Standard CounterPoint

In the above table, the condition employing a traditional slide show and animations but not spatial layout was not studied because it did not seem like a distinguishable difference from an ordinary slide show. In addition, to make the traditional slide show comparable to the spatial layout conditions, linear overviews were added at each point where a spatial overview was shown in the spatial conditions. Examples of the top level overviews are visible in Figure 49. Each of the 3 classes was then shown a presentation using the features of the assigned condition. The actual slides for each condition were identical. The slides were shown in the same order and at the same times in all conditions. Similarly, the visual presentation was accompanied by a 10 minute pre-recorded audio track that was identical for each condition. The topic of the presentation for all conditions was genetically modified crops.



- Introduction
- Potential Benefits
- Potential Risks
- Regulation
- Conclusion

Figure 49: A top level overview for the conditions in the study. The left image is the overview in the no spatial layout condition. The right image is the overview for both of the spatial layout conditions.

Following the presentation, participants were asked to fill out a questionnaire measuring subjective ratings of the presentation as well as the participants recall for content and structure. In the next class period, participants were also given a second questionnaire that again measured the participants recall for content and structure. In each condition, the follow-up questionnaire was administered following a 2 day break.

The questionnaire had two content related sections. The first section asked them to recreate an outline of the presentation. The outline section was scored using two scores of content and structure. The content score awarded 1 point for each correct item and 1 point if the item appeared in the correct position in the outline. The structure measure awarded 1 point for each correct outline position that was filled, regardless of the content of the item in that position. Overall, there were 18 possible outline items. This resulted in a maximum possible outline content score of 36 and a maximum possible outline structure score of 18. This question had the same point value in both the initial and follow-up questionnaires.

The second content section asked questions about specific facts in the presentation. Each of these questions had a different point value based on the number of possible responses. The total number of possible points in the initial questionnaire was 20 while the total in the follow-up questionnaire was 10.

The questions regarding subjective ratings of the presentations used a scale of 1 to 9 and covered four aspects of the presentation including: the overall visual portion of the presentation, the presentation's organization, the presentation's ability to convey the main areas covered, and the presentation's ability to convey the current progress of the presentation. The full questionnaires for both the initial and long-term sessions can be seen in Appendix A.

4.3.2.2 Results

One-way ANOVAs with independent measures was used to analyze the results in preference to a t-test since the study compared 3 study conditions. Post hoc tests were then used to reduce the chances of Type I error in performing multiple comparisons. In particular, Scheffé tests were used as they are conservative yet more sensitive for complex (in addition to pairwise) comparisons between conditions than Tukey tests.

The data for 28 of the 96 participants involved in the study was discarded because the subjects did not complete the content portions of one or both questionnaires. Of the remaining subjects, 36 received the no spatial layout, no animation condition, 14 received the spatial layout, no animation condition, and 18 received the spatial layout, animation condition. The remaining subjects consisted of 55 males and 13 females. The means for dependent variables measured in the study are shown in Table 8.

The one-way ANOVAs on the remaining 68 subjects showed a significant effect of presentation type on the questionnaire items regarding the overall visual portion of the presentation ($p < 0.01$, $F(2,65)=6.03$) and the presentation's organization ($p < 0.05$, $F(2,65)=3.24$). Post hoc Scheffe test indicated that the spatial layout conditions were significantly higher ranked than the traditional slide show condition on both questions with $p<0.01$ and $p<0.05$ respectively. Figure 50 compares the means and standard deviations for the responses to these two questionnaire items.

Table 8: The means and standard deviations for the dependent variables measured in the study.

	No Space, No Anim	**Space, No Anim**	**Space, Anim**
Liked Visuals (1-9)	4.81 (σ 1.82)	6.21 (σ 1.25)	6.17 (σ 1.50)
Organization (1-9)	7.14 (σ 1.29)	7.86 (σ 0.77)	7.83 (σ 1.04)
Main Topics (1-9)	7.81 (σ 1.26)	8.00 (σ 1.41)	8.22 (σ 0.73)
Progress (1-9)	7.28 (σ 1.26)	7.71 (σ 1.07)	7.44 (σ 1.38)
Specific Content (0-20)	10.33 (σ 2.89)	9.96 (σ 3.63)	8.44 (σ 2.89)
Outline Content (0-36)	11.14 (σ 4.09)	12.29 (σ 3.67)	12.56 (σ 2.77)
Outline Structure (0-18)	6.14 (σ 2.55)	7.14 (σ 3.23)	7.11 (σ 2.14)
LT Specific Content (0-10)	3.75 (σ 1.46)	3.68 (σ 1.73)	3.91 (σ 1.48)
LT Outline Content (0-36)	10.94 (σ 4.88)	13.14 (σ 5.49)	11.44 (σ 1.76)
LT Outline Structure (0-18)	5.81 (σ 2.87)	8.14 (σ 4.00)	6.72 (σ 2.52)

ANOVAs did not indicate a significant effect of presentation type on the questionnaire measures for recall of content or structure. However, as seen in Figure 51, there were emerging patterns in the outline data suggesting higher values for the spatial layout conditions. An inverse tendency appeared to hold for the specific questionnaire items shown in Figure 52. Because no significant effect was found for these scores, inter-rater reliability was not calculated for these questions.

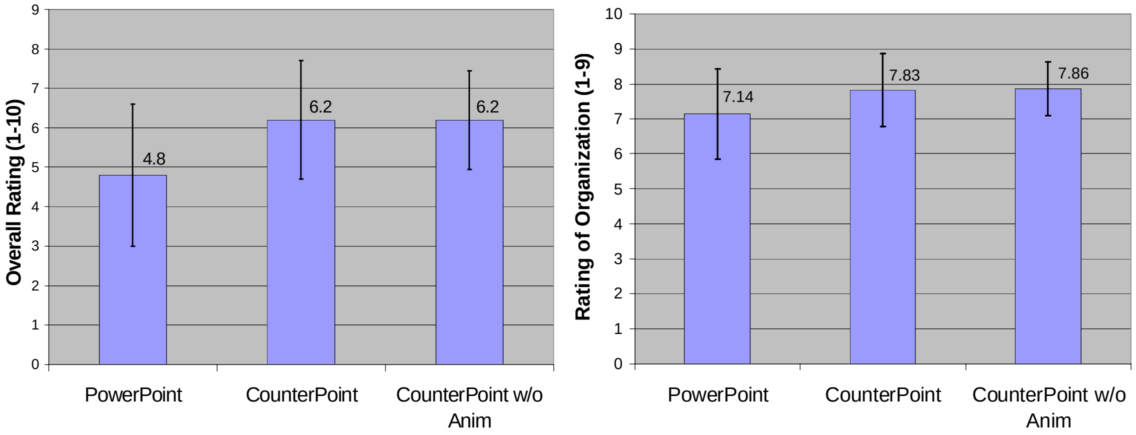


Figure 50: The graph on the left compares the means and standard deviations for responses to the questionnaire item “Rate how much you liked the visual portion of the presentation?” The graph on the right compares the means and standard deviations for responses to the questionnaire item “How would you rate the presentation’s organization?”

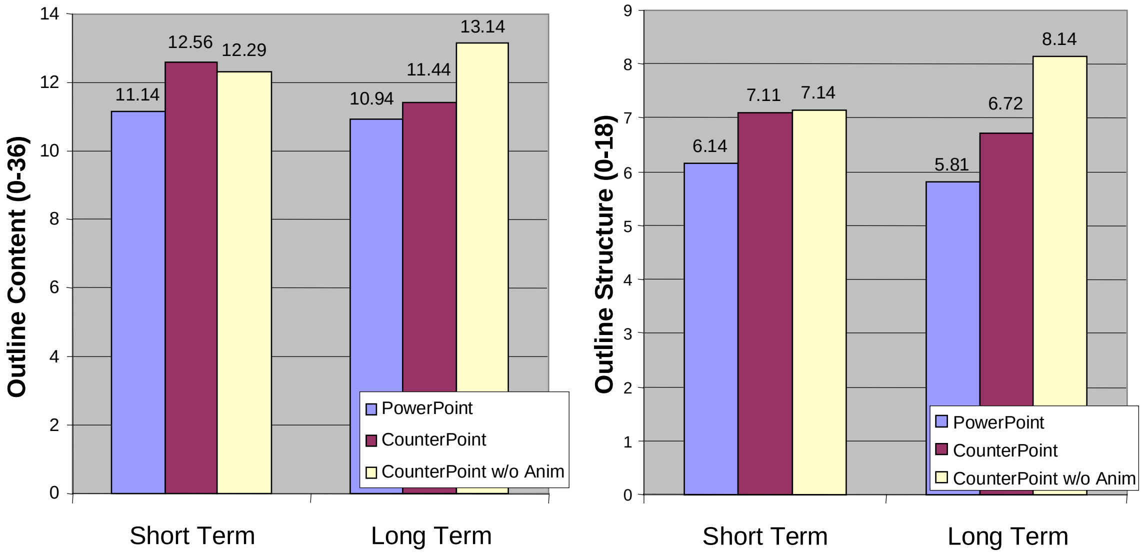


Figure 51: The graph on the left shows the short and long term recall of outline content. The graph on the right shows the short and long term recall of outline structure. These results indicate a possible pattern in the data but they were not significant.

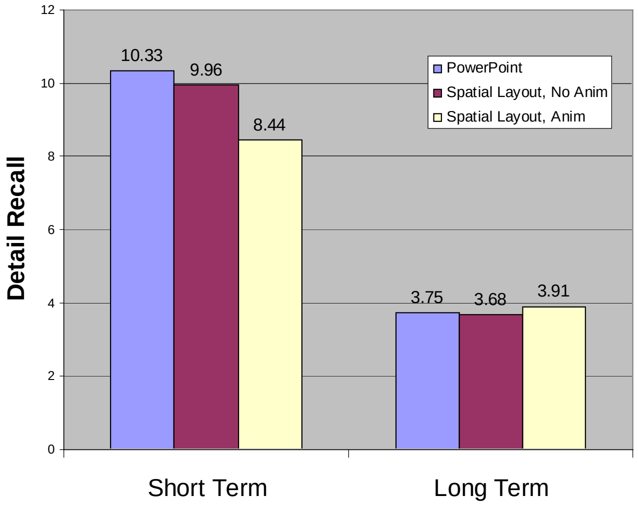


Figure 52: A graph of short and long term recall of specific content questions. These results indicate a possible pattern in the data but were not significant.

4.3.2.3 Discussion

The study revealed a general preference for the presentations with spatial layout over the more traditional slide show presentation. This could be an indication of the prevalence and resulting monotony of slide show presentations. Indeed, two-thirds of the participants indicated having seen 20 or more PowerPoint-style presentations in the last year. However, this novelty effect may also reflect in part the novelty of the spatial layout. One of the ancillary benefits of spatial layout over a linear slide show is the ability to have unique spatial layouts for each presentation. In either case, the between-subjects conditions helped reduce novelty effects as much as possible since subjects could not make comparisons between conditions.

The data analysis also indicated a significantly higher ranking of the organization of the spatial layout conditions. While this again may be related to a novelty effect, it is also supported by the pattern shown in Figure 51 in recall for information in the outline. Interestingly, this pattern seems to appear in recall for both content and structure of the outline. This lends support to our hypothesis that the audience may be benefiting from a kind of dual memory encoding in recalling high level presentation information.

The previous result is particularly significant given the artificiality in the number of overviews shown in the no spatial layout and no animation condition. Although the number of overviews was constant between all the conditions, overviews often felt unnatural when they were shown in this more traditional slide show context because they appeared as repetitive outline slides. In contrast, the overviews were more natural in the ZUI conditions since they were revealed as part of the navigations through the presentation space.

The pattern shown in Figure 52 for recall of specific content may be yet another artifact of the novelty effect. The fact that the lowest scores occurred for the spatial layout and animation condition is consistent with this explanation since this condition was the most different from a traditional slide show. A similar effect has also been noted by several presenters in using CounterPoint for real presentations. These presenters have noted an initial shock among audience members when using CounterPoint, which causes them to miss the content in the first few minutes of the presentation. This kind of extreme reaction is likely to diminish over time as CounterPoint style presentations become more common.

In general, it was not surprising that the study did not reveal a significant effect of presentation type on the recall measures. One of the biggest reasons was because the presentation was extremely short at only 10 minutes long. Further, the overviews and transitions between slides comprised only a small fraction of this already short time. This meant that the conditions in the study differed for only about a minutes worth of time. An additional reason for the lack of effect was the large variability between subjects. Subjects varied in their prior knowledge of genetically engineered crops and also in their spatial and verbal abilities.

4.4 The Presenter's Perspective

Beyond the above cognitive benefits, CounterPoint presentations also offer a number of more concrete advantages for presenters over traditional slide show presentations. In fact, CounterPoint's greatest advantages are likely to be for the presenter rather than the audience. Below, the novel affordances of CounterPoint presentations are discussed for some common presentation authoring and navigation challenges.

4.4.1 Enabling CounterPoint Features

4.4.1.1 Hierarchical Support

One of the problems in using the slide show metaphor is breaking a complex set of ideas into a linear list of slide-sized chunks. Presenters often prefer instead to organize their presentations using a hierarchical structure [55]. The CounterPoint's ZUI metaphor provides natural support for hierarchies by facilitating a spatial portrayal of hierarchies that mimics a 2D projection of a physical tree (for examples, see [52, 85]). In addition, ZUIs allow for visually distinguishing hierarchy levels by placing them at varying levels of scale or magnification. This change in magnification can decrease uniformly between levels of detail to indicate the level of nesting consistently across the presentation.

Of course, this support for hierarchies is also useful because the structure can be shown to the audience during the presentation. This frees the presenter from having to create and maintain overview slides. It can also make it easier to portray complex overviews at multiple levels of detail.

4.4.1.2 Spatial Navigation for Improvisation

CounterPoint's ZUI metaphor is based on physical space. As a result, CounterPoint offers navigation that mimics physical navigation. This type of navigation directly addresses one of the primary weaknesses in the slide show metaphor, namely the ability to jump to arbitrary points in a presentation with minimal interruption. These kinds of jumps can become necessary to address audience questions, unexpected time constraints, or other impromptu situations.

While this type of navigation is possible in the slide show metaphor, it incurs a fairly high interruption cost for presentations because of the linear nature of the navigational search. Specifically, presenters are often forced to sequentially step through a subset of their slides when searching for a specific bit of information in a traditional slide show. This becomes particularly problematic in presentations with a large number of slides. In contrast, CounterPoint provides mechanisms for organizing information at multiple levels of detail so that the number of possible navigation targets at any point in the presentation is reduced, logarithmically in the case of a hierarchical organization.

This problem can also be described in terms of the structure of the presentation. In the slide show metaphor, no structure is imposed on the slides other than their linear ordering. The slides are primarily discrete, self-contained units with no knowledge of relationships between slides. In contrast, CounterPoint allows the presenter to use the presentation's structure to organize information beyond slide boundaries. As a result, the presenter's navigation decisions in the zooming space are informed by spatial and structural cues from the inter-slide relationships. The overall result is that the complexity of navigation decisions is reduced.

Zooming navigation also supports the presenter's desire to project a consistent and professional appearance. This is accomplished by eliminating presentation-irrelevant visual controls for navigation, such as traditional GUI widgets. Because the ZUI paradigm is based on navigating a single continuous space, it is able to avoid dialogs, menus, and scrollbars in favor of consistent navigation cues, gestures, and keystrokes. These navigation controls are also compatible with controls for traditional rehearsed scripting navigation, such as those found in existing software presentation tools.

4.4.1.3 Separation of Path and Content

A common presentation task is to reuse an existing presentation for impromptu or otherwise unanticipated settings. This typically means adapting the presentation on the fly to audience interest or different time constraints. The slide show metaphor is not well suited to this type of spontaneity as the content of the presentation is directly tied to the order of presentation. Restated, the slides represent both the information a presentation will contain and the rhetorical structure the story will follow. In contrast, CounterPoint presentations separate the content of the presentation from the path through that content. This means that the presentation material remains constant independent of the specific routes through that material. This parallels the physical world in that multiple paths or tours are available through the same physical locations. The separation allows for creating traditional scripted paths and tours but also facilitates improvised storylines through dynamic navigation in an existing presentation space.

Making additional unvisited information visible in the structure of the presentation has an additional benefit for many presentations in that it can help spark audience discussion or inform audience questions. Specifically, it can suggest to the audience where additional information is available with further probing.

4.4.1.4 Visual Support for Transitions

One failing of the slide show metaphor is in providing visual support for the presenter during transitions. The ZUI metaphor addresses this issue through the use of animated spatial navigations between locations in the zoomable space. Because the zoomable space is used by the presenter to portray the structure of the presentation, this structure is revealed to the audience during the animated navigations through space. As a result, the presenter's transitions between points in the logical structure of the presentation are supported visually by the animated transitions between points in the spatial structure of the presentation.

4.4.1.5 Conveying Presentation Progress

The slide show metaphor has no inherent notion of presentation progress. One common technique presenters use to compensate for this deficiency is to manually add text specifying, "Slide n of N," to convey an overall progress value. However, such a display does not indicate more localized progress, such as the number of slides remaining in the current topic.

Through the use of a spatial metaphor, CounterPoint presentations have a natural indication of presentation progress. In particular, the spatial metaphor can indicate

CounterPoint can also provide a more explicit indicator of progress by visually altering visited slides. This concept builds on the concept of visited hyperlinks in a web browser. In our own use of CounterPoint, we have found the combination of these implicit and explicit progress indicators generally effective at conveying progress.

4.4.1.6 Creative Control

Because they support the arrangement of presentation content in a two and half dimensional space, ZUIs offer an additional degree of creative freedom over the discrete boundaries inherent in the slide show metaphor. One example that demonstrates the creative potential of the medium is described in the original Pad research [85] that overlaid text passages on an image of a tree. Another example is described in Gray Matters [114] in which information is laid out in relation to various images of body parts.

In addition, unlike other novel user interface approaches such as Hyper Mochi Sheet [111], CounterPoint presentations also support deterministic control over presentation layouts and transitions. This type of direct control ensures predictability, which authors are likely to expect for presentations.

4.4.2 Real World Experience

As discussed above, CounterPoint offers a number of potential benefits to the presenter. However, the effect of these benefits can be subtle because they are only exhibited in the transitions between slides, which are often a small part of a larger presentation. Consequently, any improvements in these transitions can be difficult to measure through controlled evaluations. Nonetheless, in our own use of CounterPoint, we have found these benefits to be compelling. In order to see if these benefits compelled others, we made CounterPoint available to the public for use in actual presentations. A version of CounterPoint has been available for download from the web since February 2001.

4.4.2.1 External Feedback

We collected feedback from 7 presenters who had used CounterPoint to deliver from 1 to 50 presentations. The total number of presentations delivered was 73, resulting in an average of 10 presentations per presenter and a median of 3. The presenters were employed primarily at educational or research institutions although some of the institutions had military and intelligence ties as well. All seven presenters either worked outside of the University of Maryland and the Palo Alto Research Center or were not directly involved in developing CounterPoint. Four presenters were interviewed in person or over the phone and the other three filled out a questionnaire on the web.

Presenters were asked to describe the features and benefits that had motivated their use of CounterPoint or that they had discovered through using CounterPoint. The most common response, given in some form by all 7 users, was the ability to portray overviews or multiple levels of detail. In describing this capability, six of the presenters specifically appreciated CounterPoint's hierarchical or "clustered" organizations and its ability to portray this structure to the audience. Related but less frequently noted benefits included portraying progress to the audience, keeping the audience oriented to the current location in the talk, and indicating the number of slides at each level of the talk.

Two of the more experienced CounterPoint presenters also valued CounterPoint's ability to keep the audience oriented during unscripted navigations. They noted that when jumping out of order in CounterPoint to answer questions, the audience is given more visual cues, through the overviews and animations, as to where the presenter is going in their talk. More of the presenters probably did not comment on this aspect of CounterPoint since they indicated that they had not discovered or made use of the navigation controls available during presentation delivery.

Presenters also indicated a number of significant challenges in authoring and delivering presentations with CounterPoint. In fact, it is a testimony to the power of ZUI presentations that presenters used CounterPoint in spite of these difficulties. Some of the authoring challenges noted by presenters included tuning layouts, compensating for missing animation features from PowerPoint, and synchronizing between PowerPoint and CounterPoint. Several presenters also found flexibility problems in adapting CounterPoint to their presentation style. In particular, two presenters wanted to be able to hierarchically nest one slide inside of another, which was not supported very well in

CounterPoint. Many of the presenters also described problems in either finding or using CounterPoint's navigation controls during presentation delivery. A common fix for navigation suggested by several presenters was to have navigation icons on the screen during delivery so presenters would not have to remember mouse gestures or key commands.

4.4.2.2 Our Experience

We have also used CounterPoint extensively ourselves for delivering presentations. Much of our experience echoes the experiences described by external presenters above. However, we have found that one of the most promising applications of CounterPoint is for more interactive presentations with a lot of information. These kinds of presentations can be characterized as "briefings" since the presenter is condensing a large body of work under limited time constraints. Often the presenter is not be able to predict or doesn't have time to prepare the specific subset of information that the audience will be interested in from the large collection.

In this vein, Mark Stefik, the author's supervisor and a lab manager at Palo Alto Research Center, has created a presentation with 101 slides that describes various organizational aspects, research areas, and business issues related to the Information Sciences and Technology lab. Mark estimates that he has delivered this large presentation over 50 times on various road trips and for visitors in the lab. Since there are a lot slides and each audience has only limited interests, Mark customizes the presentation on the fly based on what he knows of the audience and the feedback they provide. An overview of a version of his presentation is shown in Figure 53. It should be noted that although the visual design could be improved, it serves its purpose since it is combined with animated transitions, multi-level overviews, and Mark's ongoing commentary.

We have found that the use of CounterPoint in briefing scenarios resonates with intelligence and military personnel. These organizations seem to be particularly in tune with this use of CounterPoint since they have documented problems in managing the glut of presentation slides [58]. One member of our lab describes CounterPoint's benefits for these organizations in terms of the "bionic analyst." The bionic analyst creates a broad, comprehensive CounterPoint presentation with all potentially relevant information and then smoothly navigates the expanse in response to audience interests and questions.

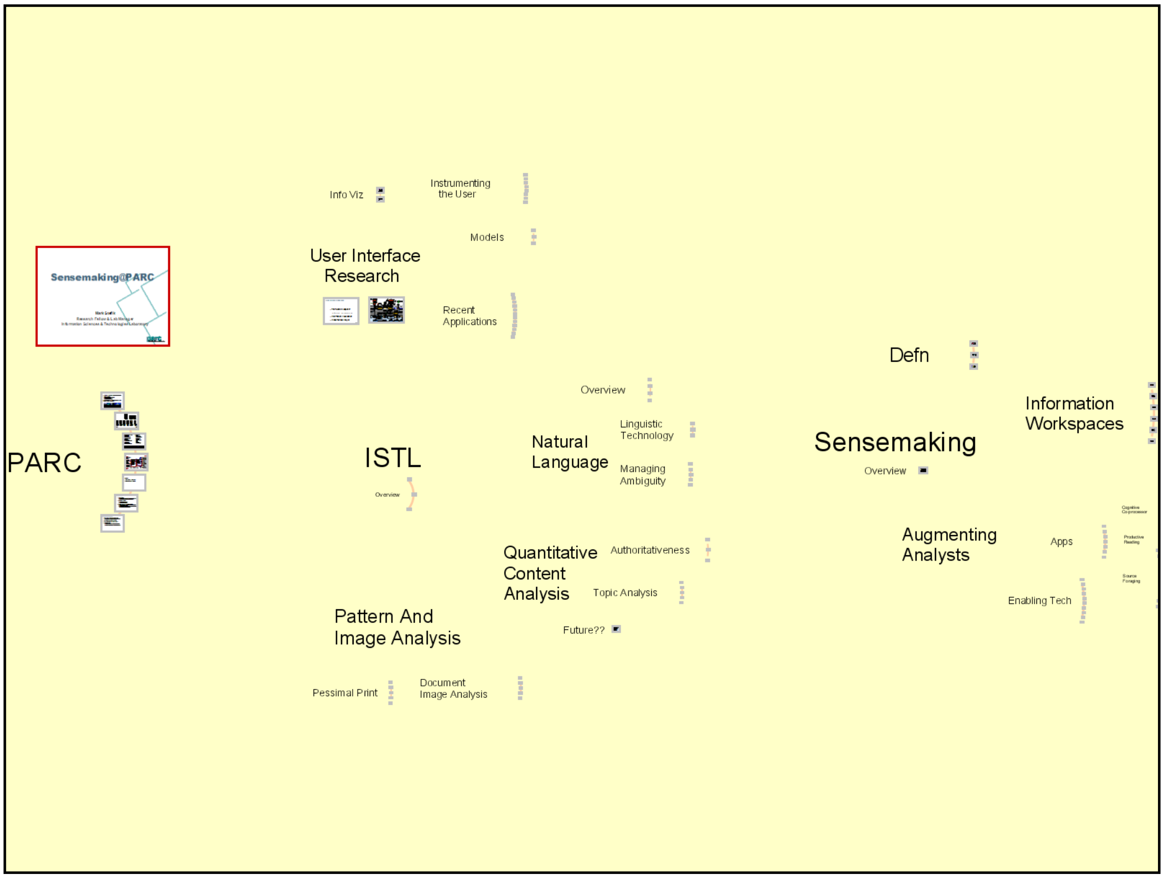


Figure 53: A version of Mark Stefik's lab overview presentation in CounterPoint.

Chapter 5:
CounterPoint Features and Implementation

5.1 Implementation

5.1.1 Architecture

CounterPoint is built on top of Jazz [10], a Java toolkit for building ZUIs, and is currently available for download on the web (http://www.cs.umd.edu/hcil/counterpoint/). In building CounterPoint, we wanted to take advantage of existing presentation tools. Although there are currently a handful of commercial slide show presentation tools available, the tool that clearly dominates the market is Microsoft PowerPoint [89]. Therefore, to have the greatest potential impact on presentation authors, CounterPoint was created as a plug-in to PowerPoint. This allowed CounterPoint to leverage presenters' existing slide authoring experience since many presenters are already using PowerPoint for presentations. In addition, it also allows presenters to reuse existing slides that have been authored and saved in PowerPoint's specific format. Most importantly, extending PowerPoint greatly reduced the functionality needed in CounterPoint.

CounterPoint uses Visual Basic's COM hooks into PowerPoint to add a toolbar button and manipulate slide content. Because the majority of CounterPoint is built on top of Jazz in Java, one of the Visual Basic application's primary responsibilities is to start a Java application when its toolbar button has been pressed. Its other major responsibility is to start a TCP/IP client by which it will communicate with this Java application.

Similarly, the first responsibility of the Java application is to create a TCP/IP server to communicate with the Visual Basic component. TCP/IP was chosen for interprocess communication because of its simplicity. Once a connection has been established between the two processes, the PowerPoint slide contents are transmitted to CounterPoint. For convenience reasons, the slide contents are not transmitted via the TCP/IP connection but are passed instead via the Windows clipboard.

This transfer of PowerPoint slide contents is possible because PowerPoint uses the Windows metafile format (i.e. files with a list of drawing commands) for posting to the clipboard, in addition to their proprietary file format. In addition, this metafile format also has benefits for our application since Windows provides native support for metafile rendering.

Consequently, a third component of our application is implemented in Windows native code for managing and rendering Windows metafiles. Our Java code uses the Java Native Interface (JNI) to communicate with the native code and to switch between native and Java rendering as appropriate.

The CounterPoint portion of the presentation data, such as slide border colorings, spatial layout parameters, and path orderings, are currently stored in a custom XML file residing in the same directory as the PowerPoint file. Because the format is XML, the file can be manually edited in a text editor in cases where the CounterPoint data has become out of sync with the PowerPoint presentation or for finer grain control over certain parameters.

In total, CounterPoint is comprised of 83 source files representing 292 java classes. The total number of lines of code in all source files is around 15K not including comments and around 19K including comments.

5.1.2 Rendering

One of the implementation issues given significant attention in CounterPoint was rendering PowerPoint slides on the zoomable Jazz surface. As mentioned, the slide data is obtained from PowerPoint in Windows metafile format. Windows metafile format is a vector graphics format used to reduce aliasing effects when rendering at different levels of magnification. One consequence of this vector format is that the system is often required to render many small visual objects for each individual PowerPoint slide. Consequently, when many PowerPoint slides are shown in a single CounterPoint overview, the rendering performance can become too slow for interactive animation frame rates.

CounterPoint addresses this problem by using cached images of the metafiles during animations and user interactions. This results in a fixed rendering time for each slide regardless of content. While this improves performance, it also introduces aliasing effects when the image is rendered at resolutions other than 1-to-1. To reduce these aliasing effects, CounterPoint uses a simple image pyramid with cached images of the metafile at multiple resolutions. The current CounterPoint version uses a fixed number of images in its image pyramid but future versions could take into consideration the number of slides in the presentation, the resolution of the display, the amount of memory on the machine, or the number of other applications running. After the initial image pyramid rendering during animations or user interactions, CounterPoint re-renders each of the affected slides in a background thread using the actual Windows metafile to further eliminate aliasing effects.

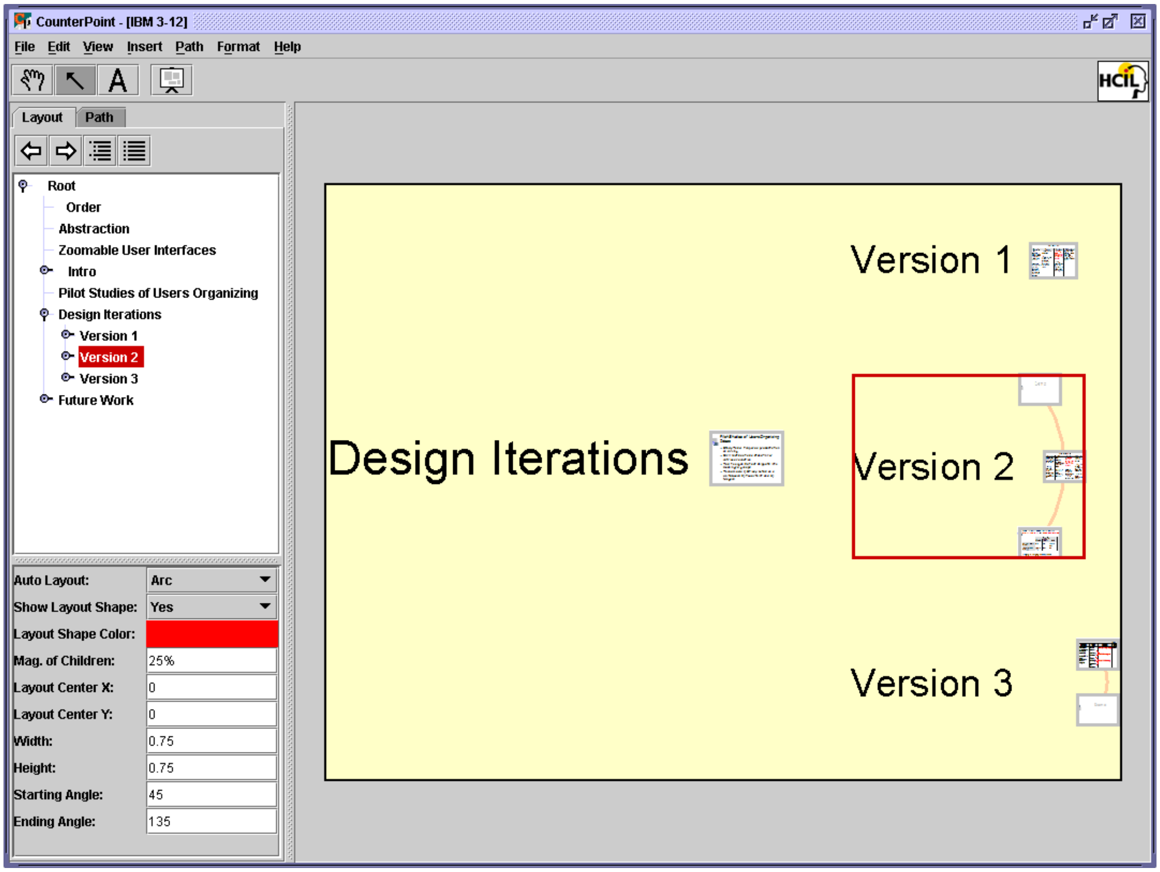
The previous metafile-based rendering approach works well for static content but does not work for animated PowerPoint slides. One experimental solution used in CounterPoint to support animated PowerPoint slides is displaying the actual PowerPoint window on top of the CounterPoint window. The PowerPoint window is shown when a slide is brought into focus and then during the slide's animations. The CounterPoint window is brought back into focus during transitions between slides or to show overviews. In order to eliminate flashing when switching between the two windows, both windows are set to be "layered" which ultimately means that the full window contents are cached as images in memory. This feature is only supported in Windows 2000 or later.

5.2 Features

5.2.1 Layout

CounterPoint distinguishes the conceptual structure from the spatial structure by providing tools for organizing slides into a conceptual hierarchy. Spatial layouts can then be automatically generated using the containment relationships from the conceptual structure. From this initial automated layout the presenter is able to manually modify parameters of the automatic layouts as well as edit the spatial layouts for more direct control. Using the conceptual structure also allows automatic layout parameters to be propagated down the hierarchy to give all members of a conceptual group a consistent visual appearance. Currently CounterPoint primarily uses the connectedness and proximity principles from the gestalt theory of perception to indicate conceptual relationships (see Figure 54). However, additional perceptual principles such as color similarity or closed forms can also be added to further emphasize group membership.

When CounterPoint loads a presentation for the first time, the presentation's slides are arranged in a grid within CounterPoint's zoomable space. Hence, the typical first step in creating a presentation in CounterPoint is to modify the arrangements of the PowerPoint slides in the zooming space. To arrange slides manually, CounterPoint provides simple tools for manipulating objects in this space similar to those found in PowerPoint, drawing programs, and previous zoomable demo programs (e.g., PadDraw [9] and HiNote [10]).



- Root
  - Order
  - Abstraction
  - Zoomable User Interfaces
  - Intro
  - Pilot Studies of Users Organizing
  - Design Iterations
    - Version 1
    - Version 2
    - Version 3
  - Future Work

Auto Layout:	Arc
Show Layout Shape:	Yes
Layout Shape Color:	🟥
Mag. of Children:	25%
Layout Center X:	0
Layout Center Y:	0
Width:	0.75
Height:	0.75
Starting Angle:	45
Ending Angle:	135

Figure 54: CounterPoint in layout organizer mode. This allows the author to create text labels and define a hierarchy with the slides.

CounterPoint also provides a hierarchy editor, shown in Figure 54, for automating these arrangements. Using this editor, the author can organize the presentation contents into a semantically meaningful hierarchy. Then, for each parent in the hierarchy, the author can apply a modifiable layout template to spatially arrange the parent's children according to the template format. Currently, CounterPoint provides layout templates corresponding to geometric shapes, such as lines, ellipses, arcs, and rectangles. For example, the slides that are children of the label "Version 2" can be automatically arranged using an arc layout to achieve this effect.

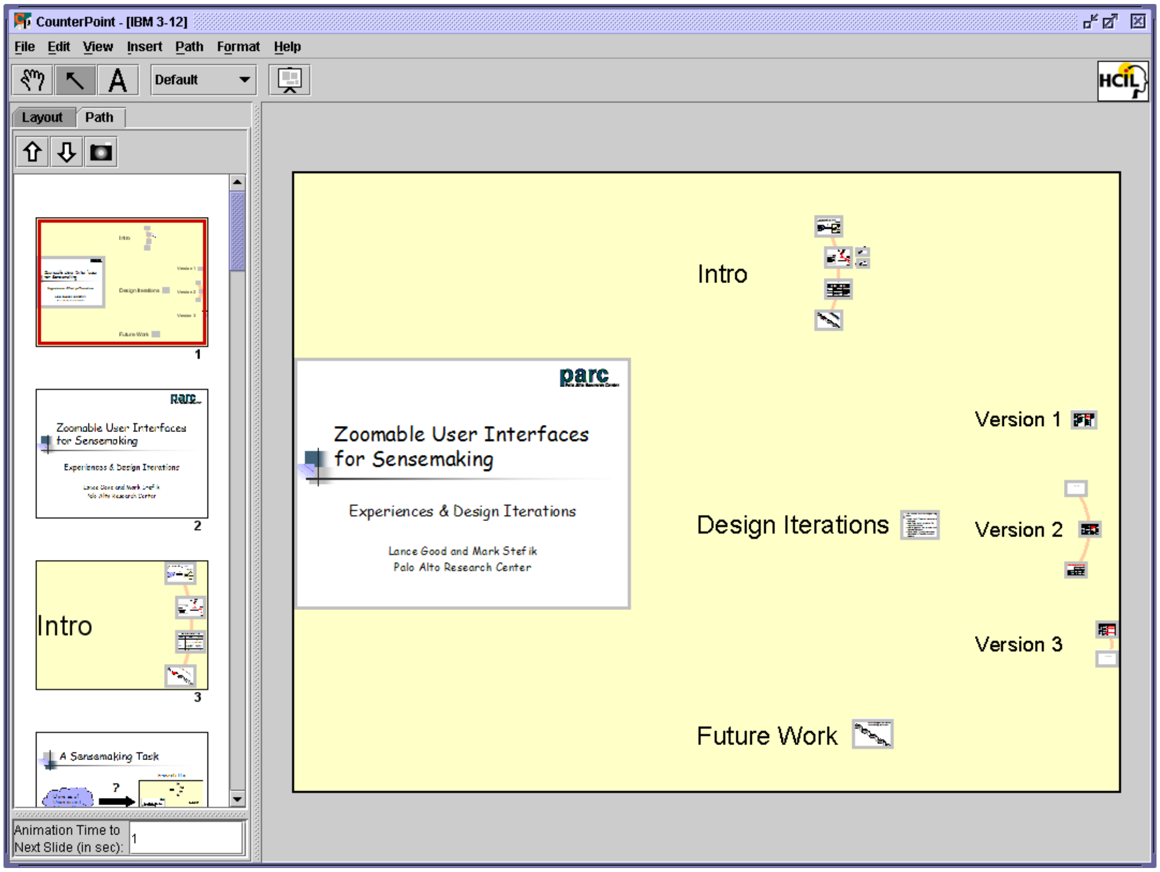
5.2.2 Path

The next step in the authoring process is to create paths through the presentation space. CounterPoint's path editing mode is shown in Figure 55. When CounterPoint loads a presentation for the first time, a single default path is automatically generated that visits each of the PowerPoint slides. In general, these paths are composed of two types of components. The first, more obvious type is the actual PowerPoint slide, which is inserted on a path to animate the slide to full screen size. These slides are inserted into the path using a simple scrolling list of thumbnails. Each slide can also be inserted multiple times in a single path.

The second type of path component is a view of a particular region of the zoomable space. These views are the more interesting path component as they allow the author to include views containing multiple slides and the structure of the presentation. Views are useful for showing an overview of the entire presentation or focused overviews of particular subsections of the presentation.

The current mechanism used to create these types of views is similar to taking a picture or creating a screen snapshot. First, the author navigates to the particular region of space to be added to the path. The author then presses the camera toolbar button (see

Figure 55) and a new component, represented by a thumbnail image of the view, is added to the path. These thumbnails are actually implemented as live views onto the presentation space so that modifications to the zoomable space are reflected in the thumbnail.

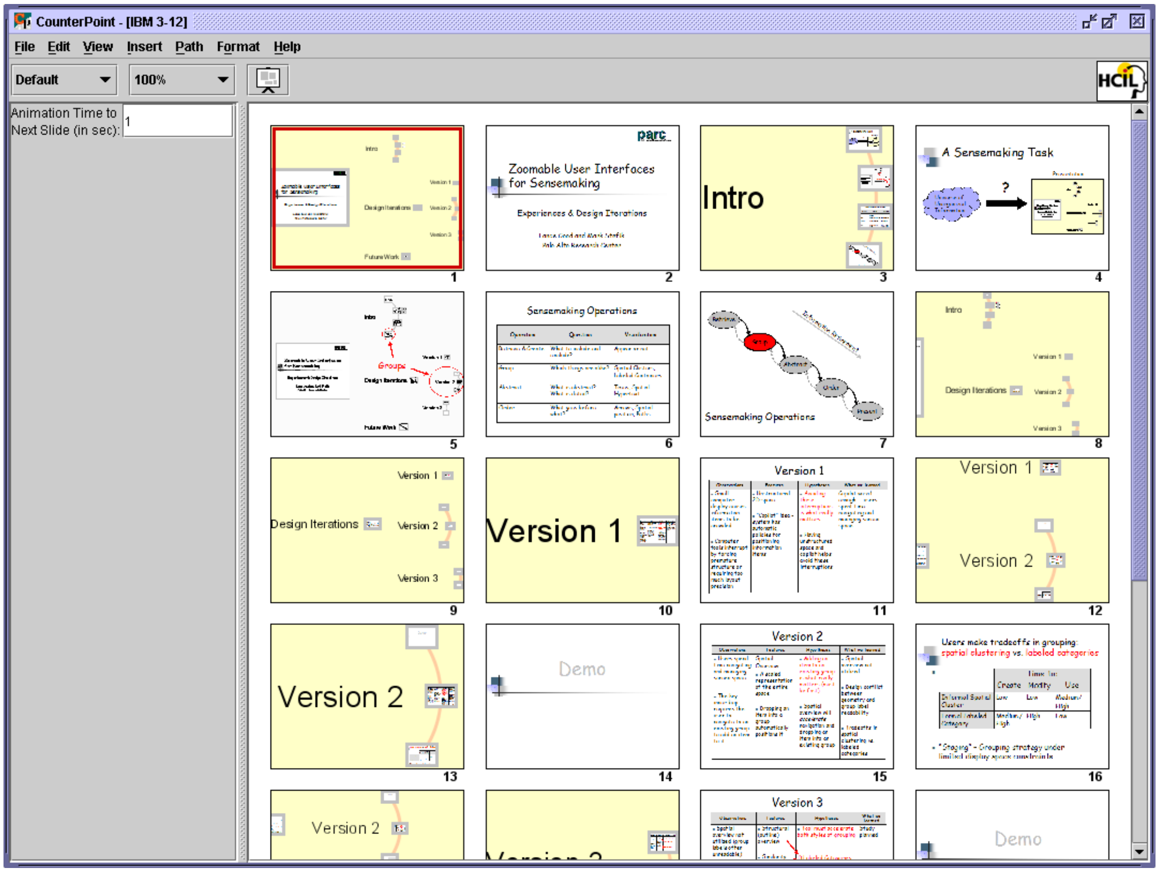

CounterPoint - [IBM 3-12]
File Edit View Insert Path Format Help
Default
HCIL
Layout Path
1
2
3
Zoomable User Interfaces for Sensemaking
Experiences & Design Iterations
Lance Good and Mark Stefik
Palo Alto Research Center
Intro
A Sensemaking Task
parc
Intro
Version 1
Design Iterations
Version 2
Version 3
Future Work

Animation Time to Next Slide (in sec): 1

Figure 55: CounterPoint in path editing mode. The panel on the left represents the sequence of slides and views in the current presentation path.

While a one-dimensional representation of the current path is available in path editing mode, CounterPoint also provides a two-dimensional path editor, shown in Figure 56, that mimics the functionality of PowerPoint's slide sorter. This will likely allow for the transfer of pre-existing PowerPoint skills since the concepts of path editing and slide sorting are so similar.

In addition, paths can be created automatically in CounterPoint using both conceptual and spatial structure. CounterPoint uses a default spatial ordering algorithm that is intended to follow the reading conventions of the culture. For western culture, this convention suggests an order from top to bottom and left to right. The algorithm also uses conceptual structure to compute a depth-first ordering of the hierarchy with overviews added for transitions between hierarchy levels or groups. Naturally, this depth-first traversal could also be manually modified to use a breadth-first or other tree traversal at any point in the hierarchy. CounterPoint represents the path as a set of viewpoints that resembles a traditional slide show. As a result, the path can be modified using techniques similar to those used in traditional software presentation authoring tools.


CounterPoint - [IBM 3-12]
File Edit View Insert Path Format Help
Default
100%
HCIL
Intro
Zoomable User Interfaces for Sensemaking
Experiences & Design Iterations
A Sensemaking Task
Sensemaking Operations
Groups
Design Iterations
Version 1
Version 2
Version 3
Future Work
Demo
Users make tradeoffs in grouping: spatial clustering vs. labeled categories
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Animation Time to Next Slide (in sec): 1

Figure 56: CounterPoint’s 2D path editor that mimics PowerPoint’s slide sorter.

5.2.3 Presentation Delivery

Perhaps the most interesting and novel interactions occur in CounterPoint's presentation mode (Figure 57). The default behavior of sequentially stepping through one of the author's predefined paths is still available. This default behavior is achieved with the standard PowerPoint controls of left mouse button, the space bar, the page down key, or right arrow key on the keyboard.

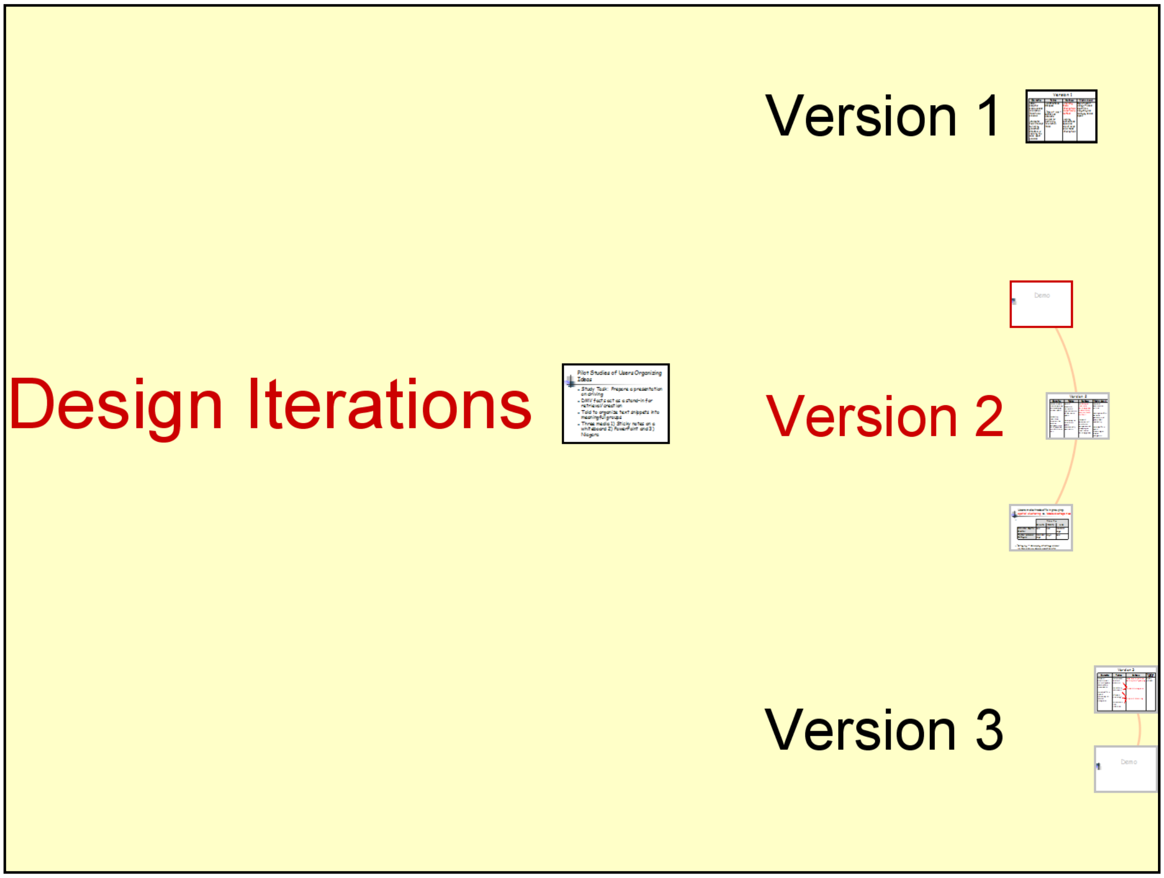


Figure 57: A screen shot of CounterPoint in presentation mode. Here, the presenter can alter pre-scripted traversals using various presentation-time interactions. Black borders indicate slides already visited during the presentation. The current focus is highlighted in red.

However, CounterPoint offers two modifications to this standard interaction for improvisation. First, the presenter can press the up arrow key to navigate up the previously defined hierarchy. This zooms out enough to get an overview of a semantically meaningful group of slides. If the layout hierarchy has not been defined, pressing the up arrow key zooms out to give an overview of the entire space.

A second interaction allows a presenter to dynamically navigate to various interesting locations in the presentation. However, before navigating to a target location, the presenter must navigate to an overview where the location is visible. This is typically achieved by zooming out using the up arrow key. For immediate access to a PowerPoint slide, left clicking on the slide animates the view to that location. Similarly, when the presenter moves the mouse within the bounds of a sub tree, the bounds of the target view highlight. Left clicking within these highlighted bounds navigates to that location.

In cases where a presenter alters the presentation path using one of these dynamic navigations, the system attempts to pick an appropriate point in the path from which to resume. In cases where the target appears in multiple places on the path, CounterPoint picks the path entry closest to the point at which the presenter deviated from the path. If the slide does not appear at all in the current path, the system does not try to infer a new path entry but rather resumes from the point at which the presenter deviated from the path.

One other traditional hypertext element that was added to CounterPoint to improve usability is visited colorings. CounterPoint provides modifiable slide border colorings to indicate which slides have been visited during a presentation. We have found these colorings to be useful both for the presenter and the audience for providing feedback as to which slides the presenter has visited and to give a sense of the overall progress of the presentation.

Chapter 6: Future Work

6.1 Authoring Presentation Content

Our understanding of a presentation authoring task was refined through 3 iterations of building and evaluating a software authoring tool. Through this iterative process, a number of areas for further work were identified. Several features and interaction techniques in Niagara need further development. The scope of authoring tasks supported in the tool also needs to be expanded. In addition, many aspects of the authoring process need further empirical study and the new interaction techniques need more real world use.

6.1.1 Niagara Implementation

Niagara introduced a number of unique features that suggest areas for further exploration. One technique that needs further work is automatic text reduction. There is a range of different algorithms for reducing text from the simplest word based techniques, implemented in Niagara, to complex techniques based on natural language semantics. These different techniques need to be tested with different document types to see how they interact with variations in content and document length. In addition, automatic text reduction could also be integrated with other text visualizations, such as enhanced thumbnails, to preserve different kinds of distinct formatting cues and document layouts. Similarly, automatic text reduction could also be improved to take into consideration the grouping structure to improve text views. For example, Figure 39 shows a number of overlapping text boxes that could be aggregated to display a more useful view for the group as a whole.

A second technique in need of further development is automatic grouping. One immediate feature suggested by Niagara's current implementation is the ability to create groups of groups. This would allow hierarchical organizations that will be essential to larger and more complex authoring tasks. Nevertheless, extending automatic grouping to multiple levels may introduce conceptual challenges for the user since groups are created and destroyed solely through object positions. Niagara's automatic groups also suggest a vast space of possible visual group designs such as different title and boundary decorations. However, these different designs must be chosen carefully since they can impact the user's expectations in interacting with the groups.

Niagara is currently limited to organizing simple, segmented text based content. A natural area for further investigation is incorporating different object types such as multi-page documents, images, diagrams, and digital ink input. Niagara also concentrates on the organization component of authoring. However, Niagara introduces a natural platform to support additional authoring tasks such as building more formal argument structures or designing other kinds of visual relationships.

A more practical area in need of further work is integrating the Niagara's functionality with existing presentation and office applications such as PowerPoint and Word. Because Niagara creates a hierarchical structure, it could also connect more directly to our work in authoring hierarchical presentations in CounterPoint.

6.1.2 Evaluation

The user studies in this work have been intended primarily to get at the high level challenges in authoring presentations on a limited size computer display and whether ZUIs address these challenges. As a result, they have not focused on the individual effects of the interface techniques that were used to make ZUIs usable. Instead, this work suggests a number of follow up studies to determine the optimal ZUI navigation interactions, the most useful semantic zooming representations, and the best techniques for transitioning from informal to formal structure. Our work also indicates that different kinds of users are likely to benefit from different interface techniques. More work needs to be done to determine the specific individual differences that are driving these differences.

The authoring techniques described here also need more real world testing. This testing will help determine the class of tasks for which a Niagara-like interface is appropriate and those for which it is not. Real world testing for large amounts of information will also help identify the scale limitations where these techniques break down.

6.2 Presentation Delivery

Much of what was learned about presentation delivery came from having a concrete prototype that integrated with an existing presentation tool. This allowed us to gather a large amount of feedback to guide future development. The majority of what is needed for ZUI presentations to gain wider acceptance is engineering to take our research prototype and turn it into an actual product. However, our work has also raised a number of interesting questions that need further investigation.

6.2.1 CounterPoint Implementation

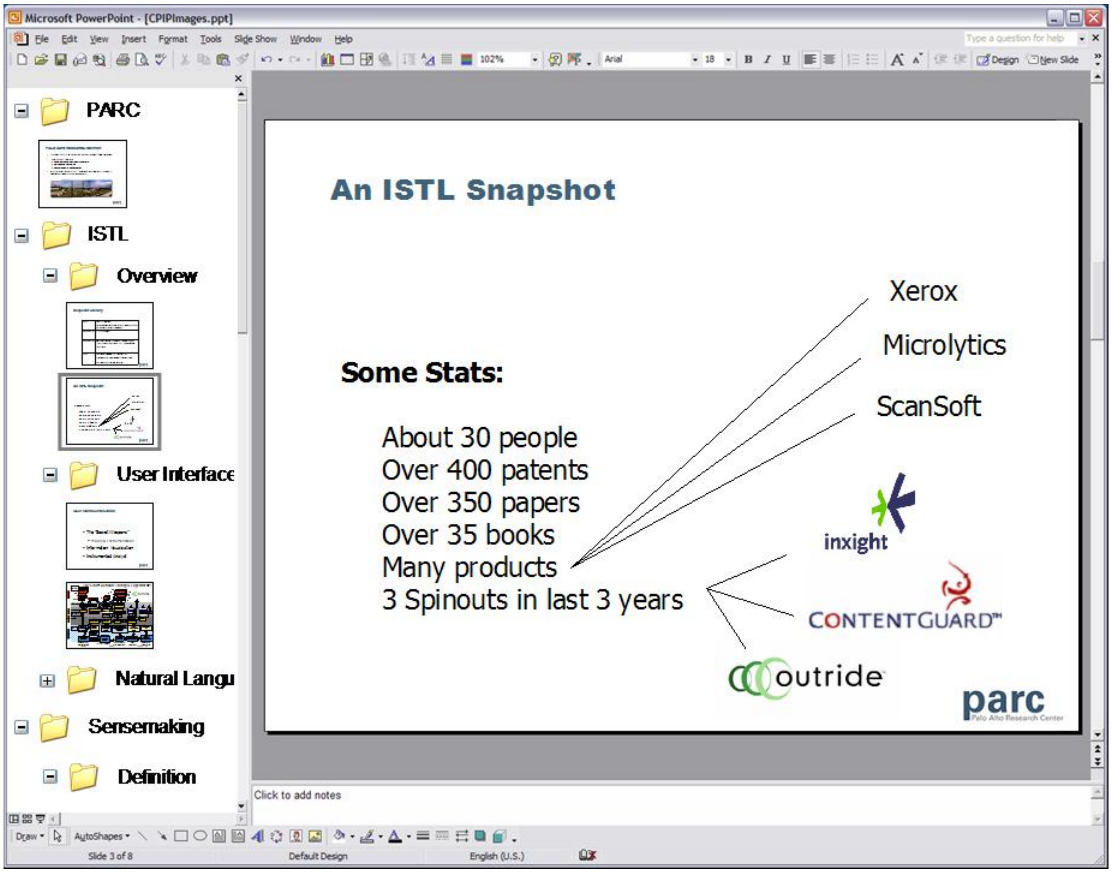
One of the practical concerns in promoting CounterPoint to a wider audience is minimizing the effort needed to author these presentations. Our existing CounterPoint implementation requires the presenter to individually attend to the hierarchical structure, spatial layout, and scripted traversal in order to obtain a usable presentation. A more ideal CounterPoint design would require the presenter only to specify the presentation's hierarchical structure. Default versions of the spatial layout and scripted traversal would then be created automatically based on this hierarchical structure. The spatial layout and scripted traversal could later be manually modified if the defaults are not satisfactory.

Figure 58 below suggests integrating a hierarchical structure with an existing slide authoring tool. Integration would provide a single unified structure that could facilitate a more natural ZUI presentation tool. This single structure could also eliminate many of the synchronization issues that currently plague CounterPoint. An integrated tool would also allow for combining advanced slide features such as within-slide animations with the spatial animations in ZUIs.

Figure 58 shows an example of an interface design for creating a presentation's hierarchical structure. The interface is much the same as a standard slide show editor but provides facilities for hierarchically grouping slides. The example design mimics the behavior of a graphical file browser but other designs, such as a textual outline editor, are also possible. Apple's Keynote presentation software already provides tools for hierarchical slide organization [63].

Once the presentation's hierarchical structure has been specified, our envisioned tool could generate an automatic spatial layout. Two example automatic layouts are shown below including a rectangular layout (Figure 59) and a network layout (Figure 60). These automatic layouts can also serve as a starting point for more manual control.

Specifically, colors, layout shapes, container properties, label sizes, and label positions can all be manually modified from the initial automatic layout.



An ISTL Snapshot

Some Stats:

About 30 people
Over 400 patents
Over 350 papers
Over 35 books
Many products
3 Spinouts in last 3 years

Figure 58: An example interface for hierarchically organizing slides.

From the presentation’s hierarchical structure and spatial layout, a default hierarchical traversal can be automatically generated. This invention proposal uses a default spatial ordering algorithm that is intended to follow the reading conventions of the culture. For western culture, this convention suggests an order from top to bottom and left to right. The algorithm also uses conceptual structure to compute a depth-first ordering of the hierarchy with overviews added for transitions between hierarchy levels or groups. Naturally, this depth-first traversal could also be manually modified to use a breadth-first or other tree traversal at any point in the hierarchy.

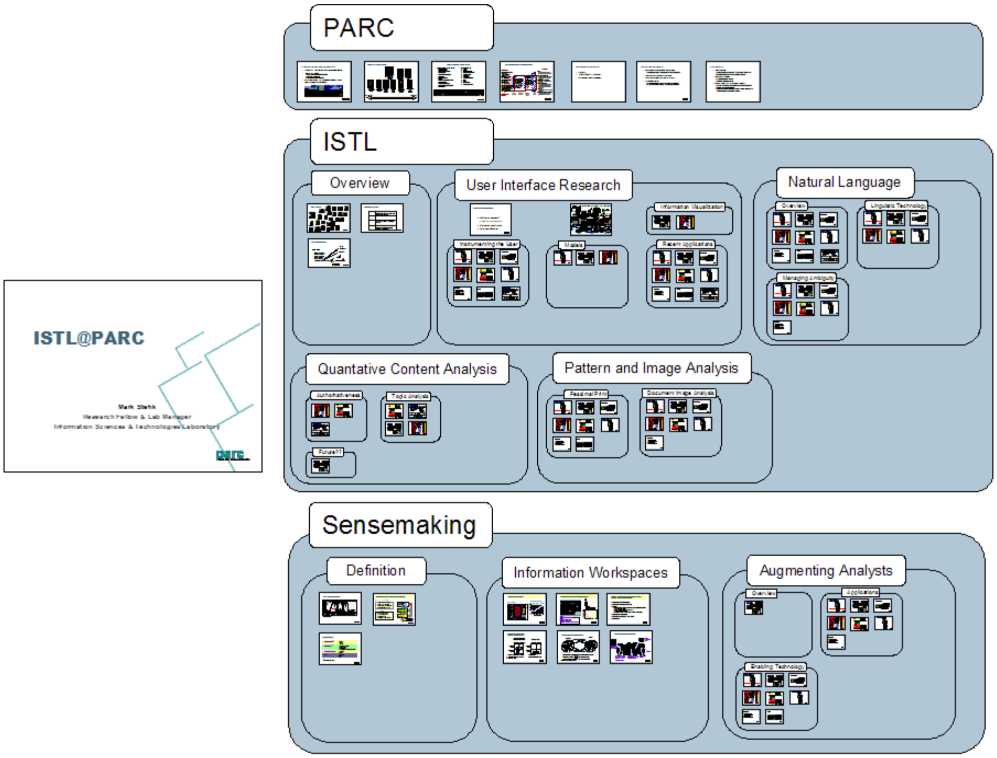


Figure 59: An example automatic layout with nested rectangular groups.

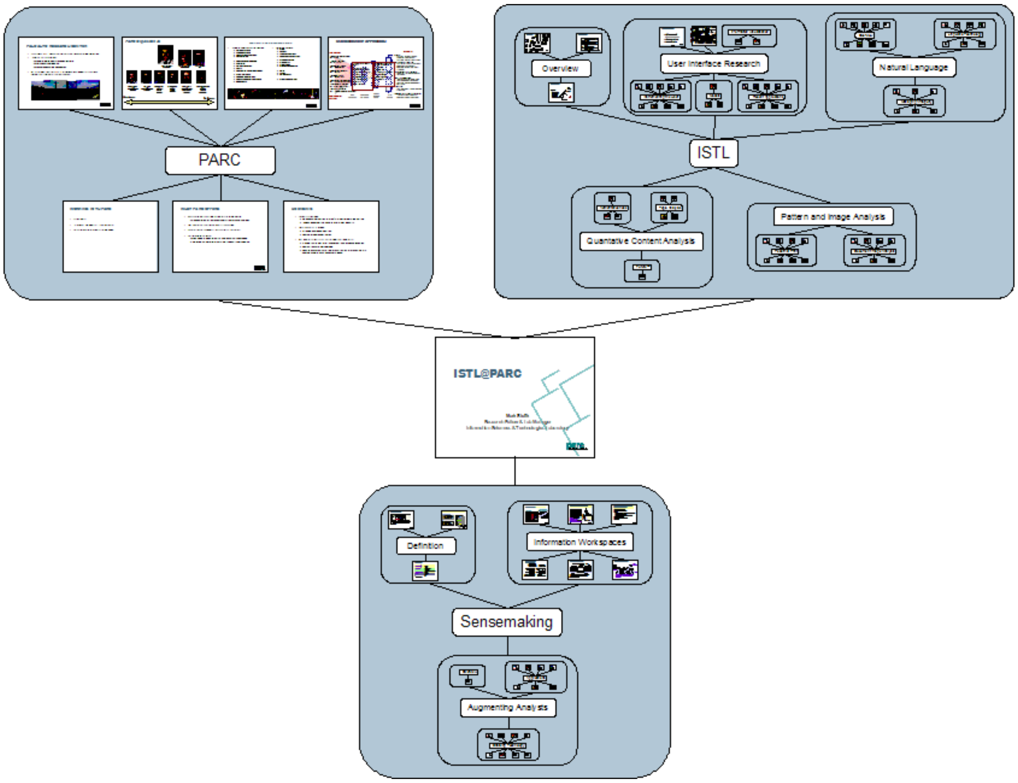


Figure 60: An example of an automatic layout with a network format.

An additional area for further work is navigation controls during the delivery of ZUI presentation controls. Although unscripted navigation is one of the areas where ZUI presentations can excel, the current interface controls still need refinement. A particular problem with CounterPoint's navigation controls is that they do not provide keyboard shortcuts for zooming-in. In addition, presenters are likely to need more visual feedback indicating the navigation options available to them.

6.2.2 Evaluation

This dissertation described an initial exploratory study looking at the risks and benefits of ZUI presentations for the audience. Although this study provides some evidence that spatial layouts are better at conveying information about presentation structure, it did not find significant differences in the explicit measures of recall for structure or content. As a result, additional studies are needed to verify this preliminary evidence. Future studies could look at longer presentations, since ours was only 10 minutes, and could separate users based on their spatial and verbal cognitive abilities. Future studies could also look at the impact of spatial layout and animation on the audience for more specific kinds of presentations or presentations events. For example, ZUI presentations seem to be able to communicate location in the presentation during unexpected changes. A study could be designed to isolate this aspect of the presentation.

In addition, our initial study showed indications that animation may negatively impact audience recall for content. This is likely to be a result of the novelty of these animations. As these kinds of presentations become more common, this kind of study could be repeated to see if this effect is mitigated by experience.

This dissertation reports a substantial amount of real world experience with CounterPoint indicating its benefits for presentation delivery. However, controlled studies are needed to formalize these benefits. Since we expect that CounterPoint may help reduce the load on the presenter in communicating presentation structure and location, future studies may want to investigate the amount of attention needed to deliver these presentations as well as their resulting quality. Finally, formal studies are also need to identify the best techniques for authoring ZUI presentations.

Chapter 7: Conclusion

The goal of this work was to address the most prominent problems in using current computer-based tools to author and deliver presentations. It is hard to overestimate the costs incurred due to these problems with over 30 million PowerPoint presentations being authored or delivered each day [82]. As software slide presentations are becoming more common, presenters are becoming more discontented with flashy visual effects, bulleted lists, chart junk, and auto-content wizards and are searching instead for tools to help them effectively make sense of and present their large collections of information.

For authoring tasks involving large collections of information, a widely recognized problem in using computer software is the lack of sufficient overviews. However, previous research has not clearly pinpointed the reasons why good overviews are necessary for authoring. This dissertation identifies organization as a specific component of authoring where effective overviews are essential. Good overviews are necessary to reduce the time needed to compare objects, allowing the author to consider more inter-object relationships in less time. The benefits of overviews for organization were demonstrated in this dissertation through several software prototypes and user studies.

In delivering presentations with large collections of information, presenters have difficulty conveying the structure of their presentations to the audience. They also have problems keeping the audience oriented in the presentation structure during impromptu navigations through their presentation slides. CounterPoint, the zoomable presentation tool described in this dissertation, introduces support for a new genre of presentations that are hierarchically structured and support interactive navigation. This new genre of presentations makes it possible to communicate larger and more complex presentation structures while tailoring these presentations to the audience based on their feedback. The benefits of these new presentations are demonstrated in this work through user studies and real-world experiences.

7.1 Presentation Authoring

In comparing our early prototypes to existing authoring tools, small qualitative studies found that users were predictably most effective with paper based authoring tools on large physical surfaces. However, paper-based information is difficult to integrate with normal computer-based work practices and the large physical surfaces are typically not archivable or portable. Instead, this research explored techniques for a standard PC that would allow the active computer display to compensate for the limited display space. Nonetheless, the first qualitative studies of our prototype found that users spent too much time navigating and managing space, which distracted them from the actual organization task.

These disappointing results motivated a change in the design of Niagara to reduce navigation costs by encouraging zooming. Three techniques were introduced to improve the value of zoomed out objects including automatic text reduction, automatic grouping, and dynamic text popups. Prior experience suggested that evaluating the revised version of Niagara in a controlled setting would be difficult due to the cognitive complexity of text-based authoring tasks. Instead, a simplified user study as designed to determine whether ZUIs offer any advantages over more traditional techniques for organizing information. Only when an effect was found in this simplified study was an evaluation performed for text based tasks.

The organization task in our simplified study involved putting shapes into groups based on common shape properties. The study compared a ZUI interface to a folder-based interface for organizing the shapes. The results from 13 subjects indicated 30% faster completion times and higher subjective ratings for the ZUI interface than for the folder-based interface. These results seemed to be driven by the faster comparisons between objects facilitated by the ZUI's continuous workspace in contrast to the disconnected folder workspaces.

Following on the promising results in the shape study, a similar controlled study was designed to try to extend these results to text. A preliminary analysis of the data from 14 subjects in this second study indicated a significant subjective preference for ZUIs as well as fewer labeled groups in the ZUI condition. A closer look at the data revealed 3 different labeling strategies that seemed to correlate with 3 different kinds of users. Analyzing the data with labeling strategy as a between subjects factor showed a significant effect of tool type on completion time and subjective preference and a significant interaction of tool type and labeling strategy on completion time. All subjects showed a substantial preference for the ZUI condition over the folder condition. However, subjects were differentiated by labeling strategy in whether they were faster with the ZUI or folders. Subjects who did not label their ZUI groups completed the task faster with the ZUI condition whereas subjects who labeled all their groups completed the task faster with the folder condition.

Overall, there were significantly fewer labeled groups in the ZUI condition versus the folder condition. This difference suggests that subjects are generally able to keep more of the groups, or at least the group criteria, in mind at once in the ZUI condition. As before, one explanation for this phenomenon is that having all the objects on-screen at once in the ZUI facilitates more rapid access to the objects and thus provides a more accessible memory aid. An additional factor is that the groups in the ZUI workspace are reinforced by spatial memory which is limited in the folder condition.

Although this research focused on organization tasks, many different kinds of tasks are likely to benefit from the ZUIs support for holding more groups in mind at once. One specific task domain where this may be particularly important is evidence structuring and hypothesis management. A particular challenge in these tasks is maintaining awareness of different kinds of evidence for several competing hypotheses. ZUIs may facilitate keeping a larger number of hypotheses or different kinds of evidence in mind at once than traditional techniques.

ZUIs had a more substantial impact in the shape organization task than in the text-based organization task. One likely reason for this is that shapes retain more of their value when zoomed out. This suggests, perhaps unsurprisingly, that the value of ZUIs for an organization task is likely to depend directly on the quality of semantic zooming representations available for the objects in the workspace. Nevertheless, these studies demonstrated benefits for ZUIs over folders for organization tasks involving two radically different object types. Consequently, there is reason to believe these results also extend to other object types, such as web page thumbnails, pen input, images, etc.

7.2 Presentation Delivery

Our research on presentation delivery was largely driven by real-world experiences using our ZUI presentation prototype, CounterPoint, to deliver actual presentations. From this use, we developed intuitions leading to an evaluation of the effects of ZUI presentations on audience preference and recall. In addition, this use also revealed a number of presentation settings for which ZUI presentations are most beneficial.

This dissertation describes a controlled study that compared presentations that varied in their use of spatial layout and animation. The audience showed an overall preference for presentations with spatial layout. They also rated spatial presentations higher in terms of organization. Though not statistically significant, these subjective ratings of organization were also supported by patterns in measures of audience recall for outline structure and content. An additional non-significant pattern indicated that the animated spatial layout presentations may result in lower recall for presentation details. This last trend is likely due to novelty effects that will be alleviated as animated, spatial presentations become more common.

This dissertation also compiled our own experiences and the experiences of 7 other presenters in using CounterPoint for well over 100 actual presentations. Presenters cited the ability to communicate overviews, hierarchical structure, progress, and orientation during presentation delivery as fundamental benefits of CounterPoint over traditional software slide presentations. Experienced users also found compelling CounterPoint's facilities to gracefully navigate the presentation and dynamically tailor their talk for the audience. A number of problems were also identified with our current CounterPoint implementation that suggest areas for future work.

Through these formal and informal evaluations, we have found that ZUI presentations are likely to be more important when the high level organization of the presentation is correspondingly more important in the presenter's message. One situation where this occurs is in highly dynamic "briefings" where audience feedback strongly influences what slides and topics the presenter will cover. Communicating the structure is also likely to be important in larger presentations with many topics and subtopics. More generally, ZUI presentations are likely to be most effective in situations where a substantial percentage of the presentation is spent in transitioning between different levels of detail. Here, the ZUI’s spatial structure and animations are visible for a larger percentage of the presentation allowing them to have a larger impact on the quality of the presentation.

7.3 Contributions

This research on the authoring and delivery presentations contributes 6 major innovations:

Authoring Presentations

- **User Interface Techniques:** Several interface techniques, including automatic grouping and automatic text reduction, that facilitate interactions with abstract information in 2D and ZUI environments.
- **Implementation:** An implemented system that combines automatic grouping and automatic text reduction in a zoomable workspace.

- **Empirical evaluations:** Evaluations demonstrating the benefits of ZUIs over a more traditional folder-based interface for tasks that involve organizing information.
- **Simplified Organization Task:** A simplified organization task involving shapes that can be reused in future research (e.g. in comparing large displays to small displays)

Delivering Presentations

- **System Implementation:** A concrete ZUI presentation system that integrates with an existing presentation tool and serves to guide future exploration in more interactive, structured presentations.
- **Empirical Evaluation:** Evaluation demonstrating audience preference for spatial presentations over traditional slide presentation.
- **Real-World Validation:** Positive feedback from presenters using CounterPoint for a substantial number of actual presentations.

Appendix A: Questionnaires

A.1 Niagara Qualitative Study Questionnaire

Past Experience

1) What is your occupation?
2) How would you rate yourself in terms of computer experience?
   (Circle one number)
   Novice 1 2 3 4 5 Expert
3) What types of tasks do you use a computer for?
   (Circle all that apply)
   a. Word Processing
   b. Games
   c. Data analysis
   d. Programming
   e. Internet (Web/Email/News/Chat)
   f. Design
   g. Other____________________________
4) How often do you use PowerPoint?
   a. Never
   b. Rarely
   c. Once a month
   d. Once a week
   e. At least once a day
5) If you use PowerPoint, what types of tasks do you use it for?
   (Circle all that apply)
   a. Organizing information
   b. Giving presentations
   c. Creating drawings/figures
   d. Making posters
   e. Other____________________________
6) How do you usually do the type of task we've just given you?

Post-Its

7) What is your overall reaction to using Post-It notes for this task?
   Frustrating 1 2 3 4 5 Satisfying
8) How satisfied are you with your final organization of the driving facts?
   Not satisfied 1 2 3 4 5 Satisfied
9) Do you think you could have done a better job with more time?
   Definitely Not 1 2 3 4 5 Definitely
10) How appropriate do you think the Post-It notes were for this task?
    Not appropriate 1 2 3 4 5 Appropriate
11) What was your biggest frustration using Post-It Notes for this task?
12) What did you see as the biggest advantage of using Post-It Notes for this task?

PowerPoint

13) What is your overall reaction to using PowerPoint for this task?

Frustrating 1 2 3 4 5 Satisfying

14) How satisfied are you with your final organization of the driving facts?
Not satisfied 1 2 3 4 5 Satisfied

15) Do you think you could have done a better job with more time?
Definitely Not 1 2 3 4 5 Definitely

16) How appropriate do you think PowerPoint was for this task?
Not appropriate 1 2 3 4 5 Appropriate

17) What was your biggest frustration using PowerPoint for this task?

18) What did you see as the biggest advantage of using PowerPoint for this task?

Niagara

19) What is your overall reaction to using Niagara for this task?
Frustrating 1 2 3 4 5 Satisfying

20) How satisfied are you with your final organization of the driving facts?
Not satisfied 1 2 3 4 5 Satisfied

21) Do you think you could have done a better job with more time?
Definitely Not 1 2 3 4 5 Definitely

22) How appropriate do you think Niagara was for this task?
Not appropriate 1 2 3 4 5 Appropriate

23) What was your biggest frustration using Niagara for this task?

24) What did you see as the biggest advantage of using Niagara for this task?

A.2 Niagara Quantitative Shape Study Questionnaire

Background

1) What is your gender?

2) What is your age?

3) How would you rate yourself in terms of computer experience?
Novice 1 2 3 4 5 Expert

4) How experienced are you using graphical authoring tools such as PowerPoint, PhotoShop, etc…?
Novice 1 2 3 4 5 Expert

Zooming

1) What is your overall reaction to using this tool for the task?
Frustrating 1 2 3 4 5 Satisfying

2) How often did you feel like software interrupted your thinking?
Rarely 1 2 3 4 5 Often

3) What was your biggest frustration using this tool for the task?

4) What was the biggest advantage of using this tool for the task?

Folders

1) What is your overall reaction to using this tool for the task?
Frustrating 1 2 3 4 5 Satisfying

2) How often did you feel like software interrupted your thinking?
Rarely 1 2 3 4 5 Often

3) What was your biggest frustration using this tool for the task?

4) What was the biggest advantage of using this tool for the task?

A.3 Niagara Quantitative Text Study Questionnaire

Background

1) What is your age?
2) How would you rate yourself in terms of computer experience?
   Novice 1 2 3 4 5 Expert

Zooming

1) What is your overall reaction to using this tool for the task?
   Frustrating 1 2 3 4 5 Satisfying
2) How often did you zoom in or out? When did you need to do this?
3) How often did you read the reduce text (not the popup text) while zoomed out?
4) What was your biggest frustration using this tool for the task?
5) What was the biggest advantage of using this tool for the task?

Folders

1) What is your overall reaction to using this tool for the task?
   Frustrating 1 2 3 4 5 Satisfying
2) What was your biggest frustration using this tool for the task?
3) What was the biggest advantage of using this tool for the task?

A.4 CounterPoint Audience Questionnaire

Initial

1) How old are you?
2) What is your gender?
3) What is your major?
4) How many PowerPoint style presentations have you seen in the last year:
   0 less than 10 20 more than 20
5) How familiar were you with genetic engineering prior to this presentation:
   1 2 3 4 5 6 7 8 9
   not familiar very familiar
6) Rate how much you liked the visual part of the presentation:
   1 2 3 4 5 6 7 8 9
   not at all very much
7) How would you rate the presentation's organization:
   1 2 3 4 5 6 7 8 9
   poor excellent
8) Identifying the main areas covered in the presentation was:
   1 2 3 4 5 6 7 8 9
   difficult easy
9) Determining the current progress of the presentation was:
   1 2 3 4 5 6 7 8 9
   difficult easy
10) Please give an outline of the presentation including the presentation title, the main areas covered, and any specific slide titles you can remember. Feel free to use words and/or drawings to represent your outline.
11) List the two types of caterpillars mentioned in the presentation?

12) List the products of agricultural genetic engineering that the presentation mentioned as available in US supermarkets now:
13) According to the presentation, what was the name of the corn that was removed from the market after it was found in taco shells?
14) List one example of a situation that the presentation describes when a genetically engineered product must be labeled:
15) List the differences mentioned in the presentation between traditional breeding and genetic engineering:
16) List the general groups named in the presentation involved in direct or indirect regulation of genetic engineering:
17) What were the main conclusions of the presentation?
18) Please write any other comments you have about the format (user interface) of the presentation:

Follow-up

1) Please give an outline of the presentation including the presentation title, the main areas covered, and any specific slide titles you can remember. Feel free to use words and/or drawings to represent your outline.
2) What crop did the presentation say was the first genetically engineered whole food on the market?
3) According to the presentation, what type of corn that was found to be harmful to Swallowtail caterpillars?
4) List the government regulatory organizations mentioned in the presentation:
5) What characteristic did the presentation say was added to “Golden Rice”?
6) What were the main conclusions of the presentation?

References

1. Allen, B. “Information Space Representation in Interactive Systems: Relationship to Spatial Abilities.” In *Proceedings of International Conference on Digital Libraries,* 1-10. New York: ACM Press, 1998.

2. Baddeley, A. D. and G. J. Hitch. “Working Memory.” In *The Psychology of Learning and Motivation*, edited by G. Bower, 8:47-90. New York: Academic Press, 1974.

3. Baudisch, P., N. Good, V. Bellotti, and P. Schraedley. “Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming,” *CHI 2002, ACM Conference on Human Factors in Computing Systems, CHI Letters* 4 (1): 259-266 (2002).

4. Baudisch, P., and R. Rosenholtz. “Halo: A Technique for Visualizing Off-Screen Locations,” *CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters* 5 (1): 481-488 (2003).

5. Beard, D. V., and J. Q. Walker II. “Navigational Techniques to Improve the Display of Large Two Dimensional Spaces,” *Behavior & Information Technology* 9 (6): 451-466 (1990).

6. Bederson, B. B. “PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps,” *UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters* 3 (2): 71-80 (2001).

7. Bederson, B. B., and A. Boltman. “Does Animation Help Users Build Mental Maps of Spatial Information?” In *Proceedings of Information Visualization Symposium*, 28-35. New York: IEEE Press, 1999.

8. Bederson, B. B., J. Grosjean, and J. Meyer. “Toolkit Design for Interactive Structured Graphics,” Technical Report CS-TR-4432, University of Maryland, 2003.

9. Bederson, B. B., J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. W. Furnas. “Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics,” *Journal of Visual Languages and Computing* 7: 3-31 (1996).

10. Bederson, B. B., J. Meyer, and L. Good. “Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java,” *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2 (2): 171-180 (2000).

11. Bell, B., and S. Feiner. “Dynamic space management for user interfaces,” *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2 (2): 239-248 (2000).

12. Blankenship, J., and D. Dansereau. “The Effect of Animated Node-Link Displays on Information Recall,” *Journal of Experimental Education* 68 (4): 293-308 (2000).

13. Boltman, A., and A. Druin. “Children’s Storytelling Technologies,” Technical Report CS-TR-4310, University of Maryland, 2001.

14. Bridwell-Bowles, L. S., P. Johnson, and S. Brehe. “Computers and composing: Case studies of experienced writers.” In *Writing in Real Time: Modeling Production Processes,* edited by A. Matsuhashi, 81-107. Norwood: Ablex, 1987.

15. Bush, V. “As We May Think,” *The Atlantic Monthly* 176 (1): 101-108 (1945).

16. Card, S. K., T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction*. Hillsdale: Lawrence Erlbaum Associates, 1983.

17. Card, S. K., and D. Nation. “Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface.” In *Proceedings of AVI 2002*. New York: ACM Press, 2002.

18. Chang, B.-W., and D. Ungar. “Animation: From Cartoons to the User Interface.” In *Proceedings of UIST '93*, 45-55. New York: ACM Press, 1993.

19. Combs, T., and B. B. Bederson. “Does Zooming Improve Image Browsing?” In *Proceedings of Digital Libraries '99*, 130-137. New York: ACM Press, 1999.

20. Conklin, J., and M. Begeman. “gIBIS: A hypertext tool for exploratory policy discussion,” *ACM Transaction on Office Information System* 6 (4): 303-331 (1988).

21. Cormen, T., C. Leiserson, and R. Rivest. *Introduction to Algorithms*, Cambridge: MIT Press, 1990.

22. Czerwinski, M., S. Dumais, G. Robertson, S. Dziadosz, S. Tiernan, and M. van Dantzich. “Visualizing implicit queries for information management and retrieval,” In *Proceedings of CHI ’99*, 560-567. New York: ACM Press, 1999.

23. Czerwinski, M., E. Horvitz, and E. Cutrell. “Subjective Duration Assessment: An Implicit Probe for Software Usability.” In *Proceedings of IHM-HCI 2001 Conference*, edited by A. Blandford and J. Vanderdonckt, 2:167-170. Berlin: Springer, 2001.

24. Dee-Lucas, D. “Effects of Overview Structure on Study Strategies and Text Representations for Instructional Hypertext.” In *Hypertext and Cognition*, edited by J. Rouet, J. Levonen, A. Dillon, and R. J. Spiro, 73-106. Hillsdale: Lawrence Erlbaum Associates, 1996.

25. Dertouzos, M. L. *What Will Be: How the New World of Information Will Change Our Lives.* San Francisco: HarperEdge, San Francisco, 254.

26. Dieberger, A., and U. Frank. "A City Metaphor to Support Navigation in Complex Information Spaces," *Journal of Visual Languages and Computing* 9: 597-622 (1998).

27. Dieberger A., C. Miner, and D. Ponceleon. "Supporting narrative flow in presentation software." In *Proceedings of CHI 2001 Extended Abstracts,* 137-138. New York: ACM Press, 2001.

28. Dillon, A., C. McKnight, and J. Richardson. "Space - The Final Chapter or Why Physical Representations Are Not Semantic Intentions." In *Hypertext – A Psychological Perspective*, edited by C. McKnight, A. Dillon, and J. Richardson, 169-191. New York: Ellis Horwood, 1993.

29. Donskoy, M., and V. Kaptelinin. "Window Navigation With and Without Animation: A Comparison of Scroll Bars, Zoom, and Fisheye View." In *Proceedings of CHI '97 Extended Abstracts*, 279-280. New York: ACM Press, 1997.

30. Druin, A., J. Stewart, D. Proft, B. B. Bederson, and J. D. Hollan. "KidPad: A Design Collaboration Between Children, Technologists, and Educators." In *Proceedings of CHI '97*, 463-470. New York: ACM Press, 1997.

31. Engelbart, D. C., and W. K. English. "A research center for augmenting human intellect." In *AFIPS Conference Proceedings*, 33:395-410. Montvale: AFIPS Press, 1968.

32. Flower, L. S. and J. R. Hayes. "The Dynamics of Composing: Making Plans and Juggling Constraints." In *Cognitive Processes in Writing*, edited by L. W. Gregg and E.R. Steinberg, 31-50. Hillsdale: Lawrence Erlbaum Associates, 1980.

33. Foster, G. and M. Stefik. "Cognoter: Theory and practice of a Colab-orative tool." In *Proceedings of CSCW '86*, 7-15. New York: ACM Press, 1986.

34. Freelance Graphics. IBM Corp., http://www.lotus.com/freelance, 2003.

35. Frenckner, K. "The Problem of Getting a Global Perspective When Reading Continuous Texts From a Computer Screen." Technical Report TRITA-NA-P9332, KTH Royal Institute of Technology, 1993.

36. Furnas, G. W., and B. B. Bederson. "Space-Scale Diagrams: Understanding Multiscale Interfaces." In *Proceedings of CHI '95*, 234-241. New York: ACM Press, 1995.

37. Furnas, G. W., and X. Zhang. "MuSE: A Multiscale Editor." In *Proceedings of UIST '98*, 107-116. New York: ACM Press, 1998.

38. Furnas, G., and X. Zhange. “Illusions of Infinity: Feedback for Infinite Worlds.” *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2 (2): 237-238 (2000).

39. Ganzel, R. “Power pointless.” http://www.presentations.com/techno/soft/2000/02/29_f2_ppl.html, 2000 (Retrieved Sept. 14, 2003 via www.archive.org).

40. Gates, P. “Where’s the Power? What’s the Point?” http://www.conference-board.org/publications/atb/articles/gatesMay02_01.cfm, 2001 (Retrieved September 14, 2003).

41. Ghosh, P., and B. Shneiderman. “Zoom-only vs. overview-detail pair: A study in browsing techniques as applied to patient histories.” Technical Report CS-TR-4028, University of Maryland, 1999.

42. Good, L., and B. B. Bederson. “Zoomable User Interfaces as a Medium for Slide Show Presentations,” *Information Visualization* 1 (1): 35-49 (2002).

43. Good, L., B. B. Bederson, M. Stefik, and P. Baudisch. “Automatic Text Reduction for Changing Size Constraints.” In *Proceedings of CHI 2002 Extended Abstracts*, 798-799. New York: ACM Press, 2002.

44. Guimbretire, F., M. Stone, and T. Winograd. “Fluid Interaction with High-resolution Wall-size Displays,” *UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters* 3 (2): 21-30 (2001).

45. Haake, J., C. Neuwirth, and N. Streitz. “Coexistence and transformation of informal and formal structures: Requirements for more flexible hypermedia systems.” In *Proceedings of ACM ECHT ‘94*, 1-12. New York: ACM Press, 1994.

46. Halasz, F. G. “Reflections on NoteCards: Seven issues for the next generation of hypermedia systems,” *Comm. ACM* 31 (7): 836-852 (1988).

47. Hancom Presenter. HancomLinux Inc., http://en.hancom.com/products/hancompresenter.html, 2003.

48. Harvard Graphics. Serif Inc., http://www.harvardgraphics.com, 2003.

49. Hayes, J. R. and L. S. Flower. “Identifying the organization of writing processes.” In *Cognitive processes in writing*, edited by L. W. Gregg and E. R. Steinberg, 3-30. Hillsdale: Lawrence Erlbaum Associates, 1980.

50. Hays, T. A. “Spatial Abilities and the Effects of Computer Animation on Short-Term and Long-Term Comprehension,” *Journal of Educational Computing Research* 14 (2): 139-155 (1996).

51. Henderson, D. A., and S. Card. “Rooms: the use of multiple virtual workspaces to reduce space conten-tion in a window-based graphical user interface,” *ACM Transactions on Graphics* 5 (3): 211-243 (1986).

52. Hightower, R. R., L. Ring, J. Helfman, B. B. Bederson, and J. D. Hollan. “Graphical Multiscale Web Histories: A Study of PadPrints.” In *Proceedings of Hypertext ‘98*, 58-65. New York: ACM Press, 1998.

53. Hinton, P. *Statistics Explained: A Guide For Social Science Students*. London: Routledge, 1996.

54. Hornbæk, K., B. B. Bederson, and C. Plaisant. “Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview,” *ACM Transactions on Computer-Human Interaction* 9 (4): 362-389 (2002).

55. Hunter, W. J. and J. Begoray. “A framework for the activities involved in the writing process,” *The Writing Notebook* 8 (1), (1990).

56. Igarashi, T., and K. Hinckley. “Speed-dependent automatic zooming for browsing large documents,” *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2 (2): 139-148 (2000).

57. Impress. http://www.openoffice.org/product/impress.html, 2003.

58. Jaffe, G. “Pentagon cracks down on … PowerPoint.” *The Wall Street Journal Online*, April 26, 2000.

59. Johnson, J. and B. Nardi. “Creating presentation slides: A study of user preferences for task-specific vs, generic application software,” *ACM Transactions on Computer-Human Interaction* 3 (1): 38-65 (1996).

60. Jones, W. P. and S. T. Dumais. “The Spatial Metaphor for User Interfaces: Experimental Tests of Reference by Location versus Name.” *ACM Transactions on Office Information Systems* 4 (1): 42-63 (1986).

61. Jul, S., and G. W. Furnas. “Critical Zones in Desert Fog: Aids to Multiscale Navigation.” In *Proceedings of UIST ‘98*, 97-106. New York, ACM Press, 1998.

62. Kaplan, N., and S. Moulthrop. “Where No Mind Has Gone Before: Ontological Design for Virtual Spaces.” In *Proceedings of ECHT ‘94*, 206-216. New York: ACM Press, 1994.

63. Keynote. Apple Computer, Inc., http://www.apple.com/keynote/, 2003.

64. KPresenter. http://www.koffice.org/kpresenter/, 2003.

65. Lai, S. “Increasing Associative Learning of Abstract Concepts Through Audiovisual Redundancy,” *Journal of Educational Computing Research* 23 (3): 275-289 (2000).

66. Lambiotte, J., D. Dansereau, D. Cross, and S. Reynolds. “Multirelational Semantic Maps,” *Educational Psychology Review* 1 (4): 331-367 (1989).

67. Marcu, D. *The Theory and Practice of Discourse parsing and Summarization*. Cambridge: MIT Press, 2000.

68. Marshall, C. C., and R. A. Rogers. “Two Years before the Mist: Experiences with Aquanet.” In *Proceedings of Hypertext ‘92*, 53-62. New York: ACM Press, 1992.

69. Marshall, C. C., and F. M. Shipman. “Spatial Hypertext and the Practice of Information Triage.” In *Proceedings of Hypertext '97*, 124-133. New York: ACM Hypertext, 1997.

70. Marshall, C. C., F. M. Shipman, and J. H. Coombs. “VIKI: Spatial Hypertext Supporting Emergent Structure.” In *Proceedings of ACM ECHT '94*, 13-23. New York: ACM Press, 1994.

71. Miller, G. “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information,” *Psychological Review* 63: 81-97 (1956).

72. Monty, M. L. “Issues for supporting notetaking and note using in the computer environment.” Ph.D. Dissertation, Department of Psychology, University of California, San Diego, 1990.

73. Moore, T. “Active Use of Hypertext to Aid Learning and Classroom Instruction.” In *Proceedings of Technical Symposium on Computer Science Education*, 297-301. New York: ACM Press, 1995.

74. Mynatt, E., T. Igarashi, W. Edwards, and A. LaMarca. “Flatland: New Dimensions in Office Whiteboards.” In *Proceedings of CHI '99*, 346-353. New York: ACM Press, 1999.

75. Nelson, L., S. Ichimura, E. Pedersen, and L. Adams. “Palette: A Paper Interface for Giving Presentations.” In *Proceedings of CHI ’99*, 354-361. New York, ACM Press, 1999.

76. Neuwirth, C., D. Kaufer, R. Chimera, and T. Gillespie. “The Notes program: A hypertext application for writing from source texts.” In *Proceedings of Hypertext ’87*, 121-142. New York: ACM Press, 1987.

77. Norman, D. A. *The Psychology of Everyday Things*. New York: BasicBooks, 1988.

78. Norman, K. *The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface*. Norwood: Ablex, 1991.

79. OneNote. Microsoft Corp., http://www.microsoft.com/onenote/, 2003.

80. Paez, L.B., J. B. daSilva, and G. Marchionini. "Disorientation in electronic environments: A study of hypertext and continuous zooming interfaces." In *Proceedings of the Asis Annual Meeting 33*, 58-66. Silver Spring: ASIS, 1996.

81. Paivio, A. *Mental Representation: A Dual Coding Approach*. New York: Oxford University Press, 1986.

82. Parker, I. "Absolute PowerPoint: Can a software package edit our thoughts." *New Yorker*, May 28, 2001, 76-87.

83. Patterson, M., D. Dansereau, and D. Wiegmann. "Receiving Information During a Cooperative Episode: Effects of Communication Aids and Verbal Ability." *Learning and Individual Differences* 5 (1): 1-11 (1993).

84. Pederson, E., K. McCall, T. Moran, and F. Halasz. "Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings." In *Proceedings of InterCHI '93*, 391-398. New York: ACM Press, 1993.

85. Perlin, K., and D. Fox. "Pad: An alternative approach to the computer interface." In *Proceedings of SIGGRAPH '93*, 57-64. New York: ACM Press, 1993.

86. Perlin, K., and J. Meyer. "Nested User Interface Components," *UIST 99, ACM Symposium on User Interface Software and Technology, CHI Letters* 1(1) 11-18 (1999).

87. Persuasion. Adobe Systems, Inc., http://www.adobe.com/prodindex/persuasion/, 2002.

88. Plumlee, M., and C. Ware. "Zooming, Multiple Windows, and Visual Working Memory." In *Proceedings of AVI 2002*, 59-68. New York: ACM Press, 2002.

89. PowerPoint. Microsoft Corp., http://www.microsoft.com/powerpoint/, 2003.

90. Prante, T., C. Magerkurth, and N. A. Streitz. "Developing CSCW tools for Idea Finding - Empirical Results and Implications for Design," *CSCW 2002, ACM Conference on Computer Supported Collaborative Work, CHI Letters* 4 (3): 106-115 (2002).

91. Presentations. Corel Corp., http://www.corel.com, 2003.

92. Rivlin, E., R. Botafogo, and B. Shneiderman. "Navigating in Hyperspace: Designing a Structure-Based Toolbox." *Comm. ACM* 37 (2): 87-96 (1994).

93. Robertson, G., M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. “Data Mountain: using spatial memory for document management,” In *Proceedings of UIST '98*, 153-62. New York: ACM Press, 1998.

94. Robinson, D., A. Katayama, and A. Fan. “Evidence for Conjoint Retention of Information Encoded From Spatial Adjunct Displays,” *Contemporary Educational Psychology* 21: 221-239 (1996).

95. Robinson, D., S. Robinson, and A. Katayama. “When Words Are Represented in Memory Like Pictures: Evidence for Spatial Encoding of Study Materials.” *Contemporary Educational Psychology* 24: 38-54 (1999).

96. Samet, H. *The Design and Analysis of Spatial Data Structures*. Reading: Addison-Wesley, 1990.

97. Severinson Eklundh, K. “Problems in achieving a global perspective of the text in computer-based writing,” *Instructional Science* 21: 73-84 (1992).

98. Sharples, M. “An account of writing as creative design.” Chap. 8 in *The Science of Writing: Theories, Methods, Individual Differences and Applications*, edited by C. M. Levy and S. Ransdell. Hillsdale: Lawrence Erlbaum Associates, 1996.

99. Shipman, F., R. Furuta, and C. Marshall. “Generating Web-Based Presentations in Spatial Hypertext.” In *Proceedings of IUI ’97,* 71-78. New York: ACM Press, 1997.

100. Shipman, F., H. Hsieh, R. Airhart, P. Maloor, and J. M. Moore. “The Visual Knowledge Builder: A Second Generation Spatial Hypertext.” In *Proceedings of Hypertext 2001*, 113-122. New York: ACM Press, 2001.

101. Shipman, F., and C. C. Marshall. “Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems,” *Computer Supported Cooperative Work* 8 (4): 333-352 (1999).

102. Shipman, F.M., C. C. Marshall, and M. LeMere. “Beyond Location: Hypertext Workspaces and Non-Linear Views.” In *Proceedings of Hypertext ’99*, 121-130. New York: ACM Press, 1999.

103. Shipman, F.M., C. C. Marshall, and T. P. Moran. “Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information.” In *Proceedings of CHI '95*, 346-353. New York: ACM Press, 1995.

104. Shneiderman, B. “Creating Creativity: User Interfaces for Supporting Innovation,” *ACM Transactions on Computer-Human Interaction* 7 (1): 114-138 (2000).

105. Shneiderman B. *Designing the User Interface*. 3rd ed. Reading: Addison-Wesley, 1998.

106. Shneiderman, B. “The Limits of Speech Recognition.” *Comm. ACM* 43 (9): 63-65 (2000).

107. StorySpace. Eastgate Systems, Inc., http://www.eastgate.com/storyspace/, 2003.

108. Sutherland, I. “SketchPad: A man-machine graphical communication system.” In *AFIPS Conference Proceedings 23*, 323-328. Montvale: AFIPS Press, 1963.

109. Thüring, M., J. Hannemann, and J. Haake. “Hypermedia and Cognition: Designing for Comprehension.” *Comm. ACM* 38 (8): 57-66 (1995).

110. Tinderbox. Eastgate Systems, Inc., http://www.eastgate.com/Tinderbox/, (2003).

111. Toyoda, M., and E. Shibayama. “Hyper Mochi Sheet: A Predictive Focusing Interface for Navigating and Editing Nested Networks Through a Multi-Focus Distortion-Oriented View.” In *Proceedings of CHI ‘99*, 504-511. New York: ACM Press, 1999.

112. Trigg, R. “Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment,” *ACM Transactions on Office Information Systems* 6 (4): 398-414 (1988).

113. Tufte, E. R. *The Cognitive Style of PowerPoint*. Chesire: Graphics Press LLC, 2003, 1-24.

114. Wardrip-Fruin, N., J. Meyer, K. Perlin, B. B. Bederson, and J. D. Hollan. “A Zooming Sketchpad, A Multiscale Narrative: Gray Matters.” In *Visual Proceedings of SIGGRAPH '97*, 141. New York: ACM Press, 1997.

115. Web Squirrel. Eastgate Systems, Inc., http://www.eastgate.com/squirrel/, 2003.

116. Wiegman, D., D. Dansereau, E. McCagg, K. Rewey, and U. Pitre. “Effects of Knowledge Map Characteristics on Information Processing,” *Contemporary Educational Psychology* 17: 136-155 (1992).

117. Woodruff, A., A. Faulring, R. Rosenholtz, J. Morrison, and P. Pirolli. “Using Thumbnails to Search the Web,” CHI 2001, ACM Conference on Human Factors in Computing Systems, CHI Letters 3 (1), 198-205 (2001).

118. Zaphiris, P., and L. Mtei,. “Depth vs. Breadth in the Arrangement of Web Links.” http://otal.umd.edu/SHORE/bs04, 1997.

119. Zellweger, P. Active Paths Through Multimedia Documents. In *Proceedings of International Conference on Electronic Publishing, Document Manipulation and Typography*, 19-34. New York: ACM Press, 1988.

120. Zellweger, P. “A Hypermedia Path Mechanism.” In *Proceedings of Hypertext ‘89*, 1-14. New York: ACM Press, 1989.

121. Zellweger, P., J. Mackinlay, L. Good, M. Stefik, and P. Baudisch. “City Lights: Contextual Views in Minimal Space.” In *Proceedings of CHI 2003 Extended Abstracts*, 838-839. New York: ACM Press, 2003.