Sketching Informal Presentations

Yang Li[1]   James A. Landay[1]   Zhiwei Guan[2]*   Xiangshi Ren[3]   Guozhong Dai[2]

Group for User Interface Research[1]
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776 USA
Tel: +1 510 642 9585

yangli@cs.berkeley.edu
landay@cs.berkeley.edu

Intelligence Engineering Lab[2]
Institute of Software
Chinese Academy of Sciences
Beijing 100080, China
guan@acm.org
guozhong@admin.iscas.ac.cn

Department of Information
Systems[3] Engineering
Kochi University of Technology
Tosayamada-cho, Kochi 782-8502, Japan
ren.xiangshi@kochi-tech.ac.jp

ABSTRACT

Informal presentations are a lightweight means for fast and convenient communication of ideas. People communicate their ideas to others on paper and whiteboards, which afford fluid sketching of graphs, words and other expressive symbols. Unlike existing authoring tools that are designed for formal presentations, we created SketchPoint to help presenters design informal presentations via freeform sketching. In SketchPoint, presenters can quickly author presentations by sketching slide content, overall hierarchical structures and hyperlinks. To facilitate the transition from idea capture to communication, a note-taking workspace was built for accumulating ideas and sketching presentation outlines. Informal feedback showed that SketchPoint is a promising tool for idea communication.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *prototyping, user-centered design, interaction styles.*

General Terms: Design, Human Factors.

Keywords

Informal presentation, Zooming User Interface (ZUI), Sketching, Pen-based Computers, Rapid Prototyping, Gestures, Storyboards

1. INTRODUCTION

Informal presentations are lightweight ways for people to communicate ideas to others for feedback and modification. One major form of informal presentations is to sketch ideas on paper and whiteboards during informal discussions, meetings and seminars. Freeform sketches are easily produced and are sufficiently expressive to convey a wide variety of information.

Current software tools provide abundant functionality to customize detailed slides for formal presentations. These heavyweight tools capture the presenter's focus with complicated menus and buttons. From the start of designing a presentation,

presenters often become obsessed by design details, such as colors and fonts, which are unnecessary for quickly communicating ideas in an informal setting.

To better understand informal presentation practices, we distributed a questionnaire to 30 researchers, and interviewed seven of these researchers from our lab. The study showed that informal presentations take an important role in the life of a researcher. The informal representations encourage more suggestions and modifications, affording communication-intensive meetings. Informal presentations are a natural transition from periods of idea accumulation and are thus an important step in the cycle of knowledge production. They enable fast and multiple iterations to further refine ideas. However, existing tools are too heavyweight and do not fit this role very well.

Based on our observations, we built SketchPoint (see Figure 1), a system supporting informal presentation design by freeform
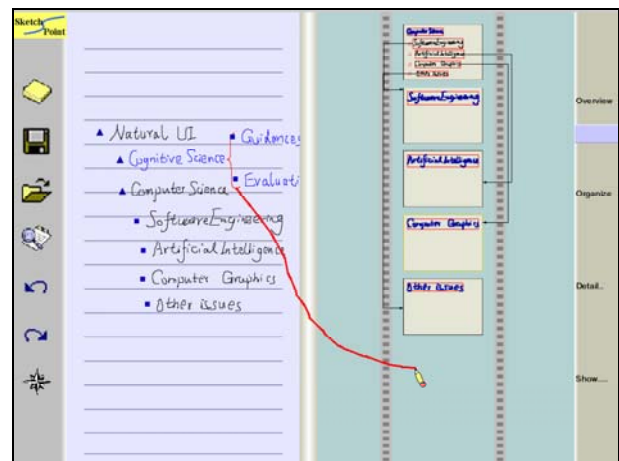

Figure 1. SketchPoint Interface. The left workspace is an electronic notebook and the storyboard on the right is for presentation authoring. The bar separating the two can be dragged to adjust the size of two workspaces. The stroke sketched from left to right will produce slides based on the selected notes and insert them into the storyboard.

* Zhiwei Guan is now a research associate of the National Research Council in Software Engineering Automation Center, Naval Postgraduate School, Monterey, CA 93940, USA.

sketching. SketchPoint enables users to quickly author presentations by sketching slide content, overall hierarchical structures, and hyperlinks. Slides are organized in a storyboard from which users can directly present their slides. To enable a smooth transition from idea capture to communication, a note-taking workspace allows presenters to pick notes for use in slides. SketchPoint supports structure-based manipulations on ink by implicitly determining the structure of ink strokes by their spatial relationships. It also enables users to indicate the structures by gestures. Informal feedback from early users of SketchPoint showed that it is a promising tool for idea communication.

In the next section of this paper, we discuss related work. In section 3, we describe the investigation that inspired us to build SketchPoint. In section 4, we provide a detailed description of SketchPoint. We also discuss the interaction techniques used in SketchPoint in section 5. In section 6, we present the current status of SketchPoint and our plans for future work. We conclude in section 7.

2. RELATED WORK

Influential commercial products, such as Microsoft PowerPoint [13], provide presenters with lots of functionality, but these tools focus on the creation of detailed formal slides, which are not suitable for informal presentations that aim at fast and lightweight idea communication.

Sinha et al's MultiPoint [18] enhances PowerPoint with multimodal interaction, which enables pen-based and speech-based input for slide authoring. Essentially, it provides recognition-based interaction that complies with the existing formal design metaphor of PowerPoint.

Sketching is an efficient and natural way to demonstrate a wide variety of ideas and information. Several systems show the capabilities and efficiencies of sketch-based user interfaces in design, idea capture and communication [1, 3, 4, 5, 7, 11]. We are the first to employ sketch-based user interfaces for informal presentation design.

Implicit, or "lazy," sketch processing greatly lowers the cognitive load of users and allows natural interaction. To enable structure-based manipulations on freeform sketches, implicit structuralizing has been explored in systems employing electronic notebook and
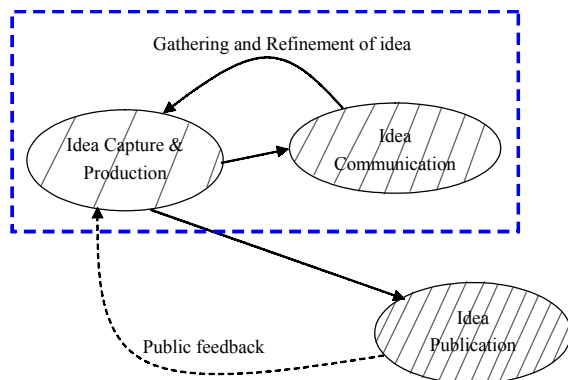

Figure 2. Idea processing requires iterative cycles of Idea Capture & Production followed by Idea Communication before eventual Idea Publication. The denser the hash marks indicates more people are involved in that step.

whiteboard metaphors [3, 10, 12, 14, 21]. Based on this prior work, we enable high-level interaction with freeform sketches by structuralizing strokes. This allows presenters to sketch line-based clusters (e.g., words), graphs (e.g., flow charts), and outlines.

Our earlier study [9] presented the conceptual design of SketchPoint as a tool for integrating note-taking and presentation. In the current work, we further the previous work with emphasis on informal presentations in the early stages of idea communication. We refine and present SketchPoint as a tool for sketching informal presentations. This paper presents the details of interaction techniques for creating informal presentations, the algorithms for converting sketched outlines into presentation slides, the structure-based manipulation based on both raw and interpreted views of ink strokes, and the implementation of SketchPoint.

One of our main goals is to provide a natural way for presenters to organize their information while designing presentations. Shipman et al [19] conducted a user study to find the traditional ways people organize information. Based on their results and our own investigations, we chose implicit and explicit outlines as the major structure for both presentation organizations and slide contents.

The storyboard metaphor has been widely used in design [1, 7, 11] because it provides an intuitive and natural way to specify sequences and maintain the structure of information. We applied the interactive storyboard as a means for maintaining the structure of a presentation and for carrying out slide-level manipulations, such as slide duplication, removal and re-layout.

Zoomable User Interfaces (ZUIs) [2] present an efficient way to organize lots of information in a limited space. DENIM [11] successfully employs this metaphor to organize the different representations of a web site design into different zoom levels. SketchPoint also utilizes a ZUI to organize the activity of presentation design into four continuous steps, i.e., four zoom levels, and allows the appropriate interaction at each level.

There has been much research in facilitating the activity of giving presentations [15] and in enhancing presentations in conference and classroom settings [4, 20]. Although our major concern is informal presentation design, we also created some advanced features for giving presentations, such as capture and retrieval of annotations when demonstrating slides.

3. CURRENT PRACTICE

To learn more about the practice of informal presentation for idea communication, we studied researchers both in our laboratory and at our university. First, we distributed a questionnaire to 30 people asking about the necessity of informal presentations, the organization and content styles they most often employed and the difficulties they have with these presentations. The results showed that all those surveyed thought informal presentations are important in idea communication. We chose to interview seven of the participants who worked in our lab so that we could easily collect and analyze samples from their notebooks and whiteboards. The detailed results of the survey, interviews, and our analysis of the collected samples are described below.

3.1 Idea Communication

Idea communication is an important step in furthering draft ideas by clarifying them and gathering suggestions from others (see

List or
title-list

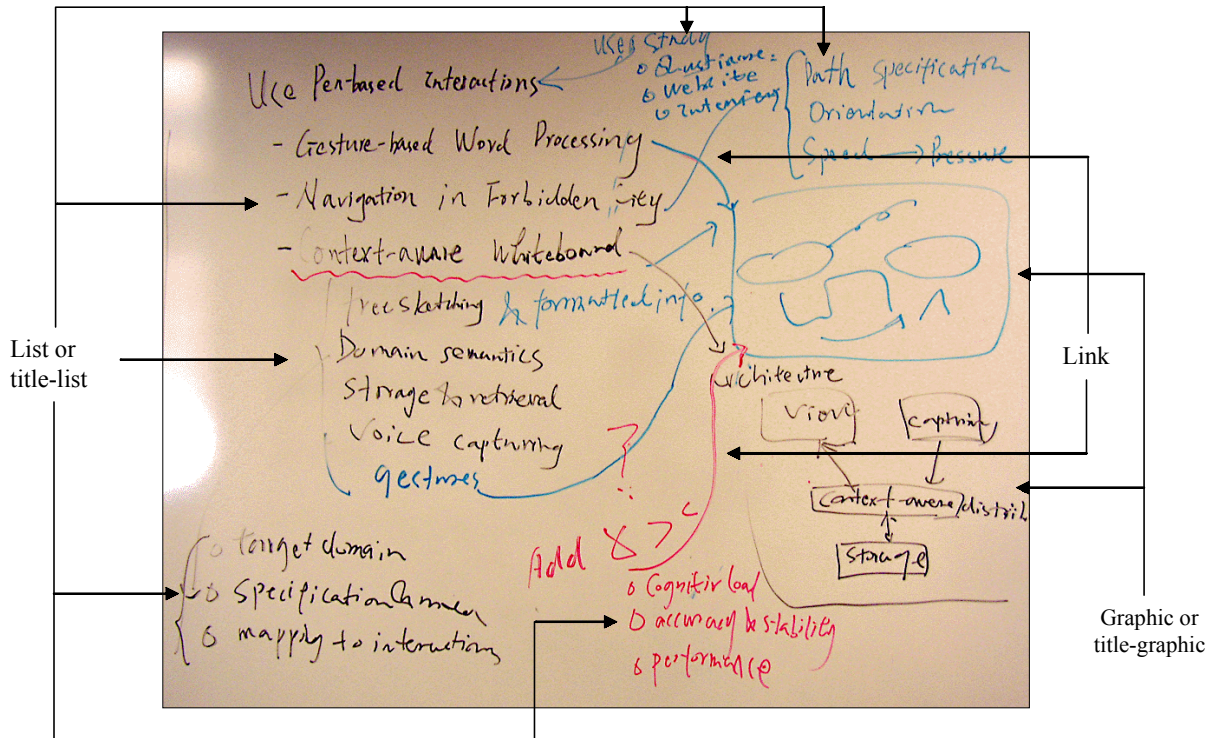Link

Graphic or
title-graphic

Figure 3. Whiteboards are often used to create and capture informal presentations.

Figure 2). In informal situations, like weekly internal seminars, our participants go through multiple iterations of idea capture & production and idea communication. Once the idea is mature, it is communicated to more people through idea publication.

Informal presentations are often used for communicating ideas in small groups. The pen and paper metaphor has great advantages in informal presentations. It allows diverse information to be easily conveyed, such as words and images, by sketching on paper, whiteboards or walls. Our participants said sketching is also helpful to clarify draft ideas and make abstract ideas concrete.

Existing electronic presentation tools bring much functionality to authoring formal, well-formatted presentation slides. However, our investigation showed that frequent internal idea communication still relies on pen-and-paper-based informal presentations. As shown in Figure 2, before an idea matures and is published, it needs multiple iterative refinements. Most of the researchers thought precision-oriented slides are well-accepted by large-scale public audiences; however, they require lots of work to author. It is not easy to achieve fast idea communication for

refining draft ideas with precision-oriented slides since they force, or at least encourage, presenters to expend too much effort on representation details, such as fonts, colors, and layout.

In addition, informal presentations seem to welcome more feedback and resulting modification, which is important for getting an idea right. This is because the informal representations imply that it is unfinished work and is welcome to further modification, as was demonstrated for web site design [6].

Our investigation also showed that researchers believed more formal presentations are suitable for idea publication (see Figure 2), which is infrequent enough to allow for relatively long and detailed preparation.

Note taking is a convenient way to capture ideas, and it is closely related to informal presentations. People organize captured ideas found in their notes and communicate them to others. In other words, note taking is an important preparation step and a source of material for informal presentations.

3.2 Creating Informal Presentations

We asked our participants how they currently create and conduct informal presentations and what kinds of schemes they use to organize their information. The investigation showed that researchers usually started with their paper notebooks, which contain captured ideas from their daily work life and some rough organization of information for presentation. Based on these materials, they transcribed and sketched the information on whiteboards, which work well for small group discussions.

Our investigation found that a tree-like hierarchical structure is the most frequently used organization scheme. This organization is efficient for presenters to organize content in a logical way, and it is also makes it easy for the audience to understand the big
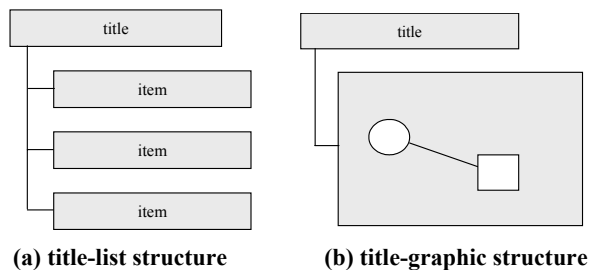
title	title
item	
item	
item	

(a) title-list structure   **(b) title-graphic structure**

Figure 4. Common organizations used in informal presentations.

picture. For example, the structures of title-list or title-graphic are common organizations of information (see Figure 3). That is, a topic can be backed up by several sub-topics and each of them can be recursively decomposed into further details.

We found that often different pieces of content are semantically linked. Presenters specify the links by sketching arrowed lines between the related information (see Figure 3). These links might indicate further explanation, references or affiliation.

By analyzing answers from our survey and the collected samples from our interviews, we distilled two major information organizations of informal presentation as shown in Figure 4[1]. We also found that flowcharts (see Figure 3) with nodes and links are an efficient way to describe relationships and that many other sketched graphics share a similar topology, e.g., UML diagrams and State Transition Diagrams.

3.3 Problems Creating Presentations

Our investigation showed that there was no efficient electronic tool to help our participants with their informal presentations. Traditional presentation tools focus on a formal style of presentation design that is not suitable for informal presentation purposes. People prefer the pen-and-paper metaphor for informal presentations.

Although sketching on physical whiteboards and on paper is natural, there are some problems. The information is not easy to maintain and edit. For example, information on physical materials is difficult to rearrange. Presenters have to maintain semantic relationships by themselves. It can be difficult to be clear about the ideas you are working on, with complicated arrows going to and fro on whiteboards (see Figure 3) or on paper. Both presenters and audiences easily get lost when a large amount of information is scattered across whiteboards. And there are no electronic tools to support these activities, other than to clean up images taken with a digital camera after the fact.

Although commercial electronic whiteboard systems make it easier to capture and edit information than with physical whiteboards, it is still difficult to use these systems for informal communication since they do not support higher level semantics.

Manually transcribing notes to informal presentation media, like a whiteboard, is not efficient for idea communication since it can be abrupt for the presenter to change modes and may disturb the concentration of the audience. No existing tools support a smooth transition from note taking to informal presentations.

4. THE SKETCHPOINT SYSTEM

Based on our interviews and personal observations, we designed SketchPoint to support and enhance informal presentations. The design of SketchPoint is based on the pen-and-paper metaphor. It supports the task of informal presentation design by using two workspaces, organized in four continuous zoom levels as shown in Figure 1. A natural path from note taking to presentation design enables the transition from idea capture to idea communication. SketchPoint abstracts and unifies the processes of informal presentation design into four continuous zoom levels based on the

---

[1] Both types of organization found in Figure 3 are shown in vertical layouts, but they can also have horizontal layouts, e.g., the title is to the left side of the contents (or list).
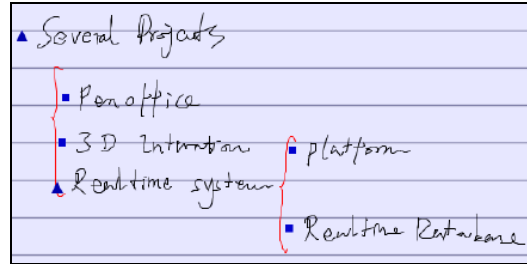

Figure 5. SketchPoint users can specify an outline by explicit gesture, i.e., large brackets. The solid triangles or squares, called manipulators, are produced by SketchPoint based on structure parsing. These are used to manipulate the outline, e.g., collapsing or expanding it.
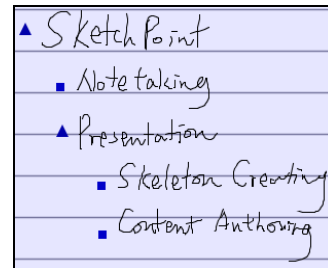

Figure 6. An outline can also be specified implicitly by indenting text as it is written.
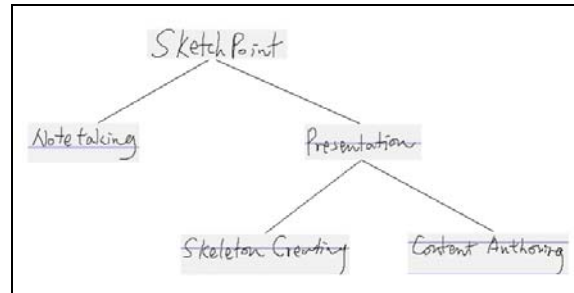

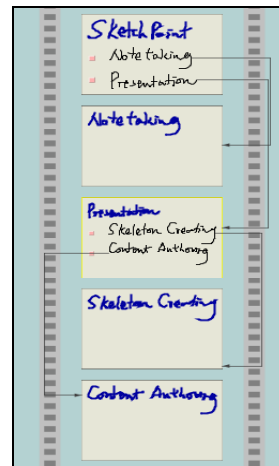**Figure 7. Internal structure of the outline in Figure 6.**

Figure 8. SketchPoint automatically produces the presentation skeleton with hyperlinks to later slides from the outline of Figure 6.

ZUI metaphor [2]; they are *Overview*, *Organize*, *Detail* and *Show*. (The zooming slider is on the rightmost portion of the screen in Figure 1) A storyboard metaphor is used to organize presentation slides.

4.1 Converting Notes to Informal Slides

As shown in Figure 1, the note-taking workspace enables users to sketch freeform notes that can be structuralized through either explicit gestures (see Figure 5 and Figure 14) or implicit layouts (see Figure 6). The basic unit of the notes is a cluster, rather than a stroke. Clusters are parsed based on the algorithm described in [10]. The high-level structures supported by SketchPoint are vertical and horizontal title-list structures or their combinations as illustrated in Figure 5. These structuralized notes allow users to manipulate them based on high-level semantics. For example, users can tap the solid triangle to collapse or expand a list. Users can also delete or select an entire list by deleting or selecting the list title.

Selected notes can be converted to informal slides in the presentation workspace by sketching a converting gesture as illustrated in Figure 1 (the stroke is drawn from the note-taking workspace to the presentation workspace). The conversion is performed based on the parsed internal structure of the sketches, which is maintained by SketchPoint.

The outline is a natural organizational scheme for presenters to specify the structure of a presentation since it can be used to produce the skeleton of a presentation. The outline in Figure 6 can be represented by a tree as shown in Figure 7. The presentation skeleton produced from this outline is shown in Figure 8. The skeleton production algorithm is based on the traversal of the tree structure. The algorithm recursively transfers an outline element from the notes into a set of sorted and linked slides according to the internal structure of the notes. The algorithm is described below:

Given:
- x : the root cluster of an outline structure of selected notes
- x' : the clone[2] of x
- x_i : the ith immediate child cluster of x
- x_i' : the clone of x_i
- n : the number of immediate children of x
- y : the slide in the storyboard after which the produced slides will be inserted

CreateSlidesFromOutline (x, y)
1. Build a slide s that includes cluster x' and x_i' , $0 \leq i < n$. Set x' as the title of slide s
2. Insert s after y
3. for $i := n - 1$ to 0:
4. $m := CreateSlidesFromOutline$ (x_i, s)
5.   Create hyperlink from x_i' of slide s to slide m
6. Return s

4.2 Using Storyboards to Specify Sequencing

By sketching and using outlines for the overall structure of a presentation, presenters can keep a clear vision of how their slides are organized. SketchPoint converts these outlines into a

presentation skeleton made up of sequentially arrayed slides in a storyboard. Presenters can then further refine and add to the skeleton.

The interactive storyboard, shown in Figures 8, 9 and 10, is rendered as a filmstrip that is centered in the presentation design workspace. The linear sequence of slides, arrayed vertically, in the storyboard determines the default demonstration sequence of an informal presentation.

The sequential order of slides in the storyboard is simple enough to be understood and anticipated by presenters. The skeleton embodies the hierarchical structures of a presentation by combining the linear slide sequence with non-linear hyperlinks. In Figure 9 a presenter is drawing a hyperlink to enable a non-linear transition when giving a presentation.
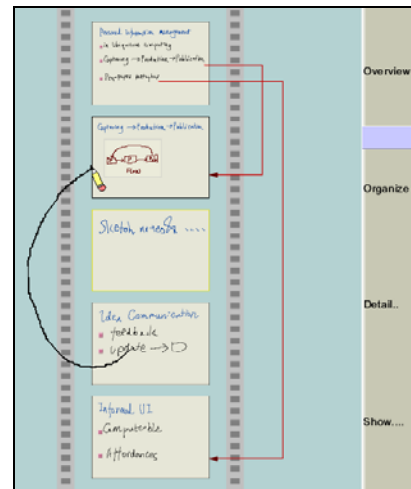

Figure 9. Users can sketch hyperlinks from a cluster in a slide to a target slide to allow jumps during a presentation.
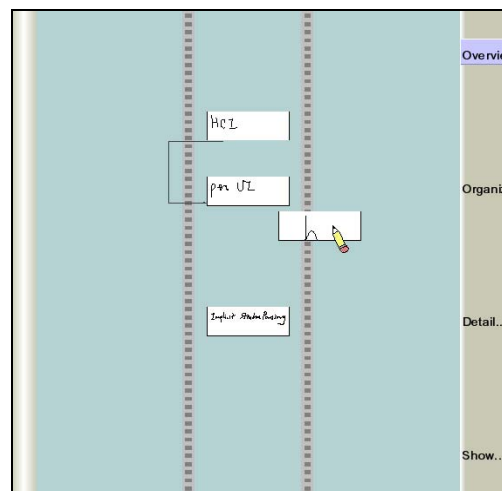

Figure 10. At the Overview level, only rough views of slides are shown (If a slide has a title, only the title is shown). In this figure, a user is dragging a page with a title "h". The user can resequence or delete a slide by dragging and dropping it to a new position in the storyboard or away from the storyboard, respectively.

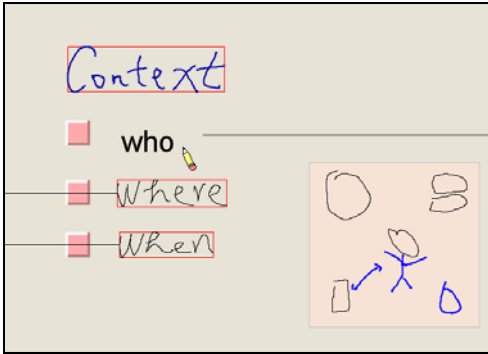[2] A duplication of a stroke cluster

Figure 11. Line-based clusters organized in a title-list structure. Users can add strokes to an ink-based cluster, and they can drag recognized objects, e.g., words, as shown here. Three horizontal lines are hyperlinks to other slides.

Besides maintaining the sequence of the slides, the interactive storyboard also enables fundamental slide level manipulations, such as removing a slide, inserting a slide and adjusting the presentation order of a slide.

SketchPoint employs ZUI [2] based navigation in the presentation design space. Users can select a slide as the zoom focus using a circle gesture and this slide will become the first slide for a demonstration. Unlike DENIM, SketchPoint allows continuous zooming, which gives presenters more flexibility and immediate feedback upon zooming. SketchPoint supports four continuous zoom levels. Any zoom level allows a presenter to insert a new slide using an Insert gesture.

4.3 Adding and Organizing Slide Contents

Presenters can enrich the skeleton by adding new slides in the storyboard or by putting detailed content in the slides. When zooming to the Detail level, presenters can sketch words, graphics and other symbols into a slide just as they can on paper or whiteboards. SketchPoint maintains the naturalness of free-hand sketching but adds implicit structure parsing to enable structure-based manipulations.

As shown in Figure 11, the slide structure supported by SketchPoint is title-list. A title is a line-based stroke cluster. The items of a list can be a line-based cluster or a structure patch. A structure patch, such as the one containing the flowchart in Figure 12, is used to support arbitrary graphics.

When sketching, the sketches are dynamically and implicitly grouped into horizontal line-based clusters based on the temporal and spatial distance and density information. To achieve low cognitive load, the feedback is given in a lazy style so that the presenter can sketch continuously without being disturbed. These line-based clusters work not only for words but also for any symbols that can fit on lines. SketchPoint does not care about the semantics of the sketches themselves. Instead, SketchPoint parses the logical structure of the sketch, i.e., clusters and their relationships.

Although SketchPoint mainly supports informal representations of slides, it also provides an optional feature that allows presenters to replace the original ink by recognized words or linked shapes (see Figures 11, 12). The segmentation and recognition of sketches can be invoked when necessary. The presenter can tap a cluster to see
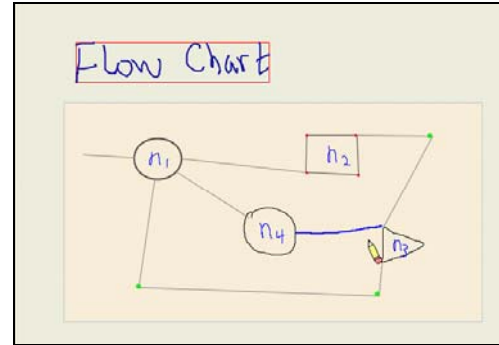

Figure 12. A structure patch containing a flowchart. Both the nodes and links of the flowchart can be adjusted. Users can switch between the recognized and original ink views of the patch by tapping with the pen.

the recognized view and tap it again to change it back to the original ink view.

In the recognized view, a cluster can be dragged and dropped to adjust the layout of the content, as shown in Figure 10. While in the ink view, the presenter can continue to add new strokes to a cluster. A presenter can enter a word correction mode to correct both segmentation and recognition errors by holding the pen on the cluster for a short moment. SketchPoint performs incremental word segmentation and recognition. Parsing can be improved based on corrections made by the presenter. The presenter can terminate the correction mode by holding the pen down again.

SketchPoint also allows presenters to sketch graphics that do not fit in line-based structures. These are placed in structure patches. SketchPoint allows presenters to specify a rectangular area as a structure patch by holding the pen to trigger a rubber-band action. This avoids the need to segment between line-based clusters and arbitrary graphics, which would compromise the stability of the system and introduce usability problems because of the limitations of current algorithms.

Presenters can sketch graphics freely and continuously in the structure patch. A structure patch can be associated with a particular parsing engine to parse different types of graphics. Currently, SketchPoint only supports a flowchart parsing engine, which recognizes graphs with links and closure nodes. A presenter can edit the flowchart structurally by moving nodes along with their associated labels and by adjusting the layout of links. The presenter can sketch words (or any other symbols) as labels of a graph node. The parser does not try to recognize these words but does discern the spatial relationships of these elements.

4.4 Conducting an Informal Presentation

Using the mechanisms described in the previous subsections, a presenter can build an informal presentation. The presenter can zoom the presentation workspace to the Show level. The presenter can then give an informal presentation, as shown in Figure 13. The slides will be projected to fill the entire screen.

The presenter can tap to move forward in the presentation and double-tap to move backward. If the presenter taps on an item with a hyperlink, the presentation will jump to the target slide of the link. We use the duration between two adjacent taps to judge whether the user has performed a double-tap or two separate taps. This can be inexact and should be user definable. The presenter
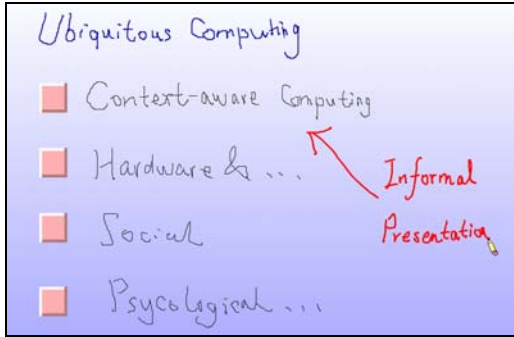
Figure 13. Conducting an informal presentation. Annotations can be sketched on the slide during the presentation and later retrieved by time or slide.

can terminate the Show mode either by holding the pen for a short while or tapping forward until the final slide.

When giving a presentation, the presenter can freely sketch annotations as shown in Figure 13 to conduct a more vivid and interactive presentation. The annotations will be recorded and tagged with the time of the presentation and the current slide for later contextual retrieval.

5. INTERACTION TECHNIQUES

SketchPoint takes advantage of the affordances [16] of the pen-and-paper metaphor and provides a natural WYPIWYG (What You Perceive Is What You Get) [17] interaction style. Interaction techniques used in SketchPoint include tap, hold, drag, and a small set of gestures (see Figure 14). These are lightweight techniques, and informal feedback showed that they do not require much effort to learn to use. There is no explicit mode switch for gesture versus sketching; SketchPoint discriminates between them using contextual information. For example, if an "erase" gesture is drawn on blank space instead of over other strokes, it will be considered sketched input.

Users can interact with high-level structures, such as title-list, instead of with single strokes. SketchPoint presents a consistent interface for both sketching and for manipulating content in different contexts. Most sketched content in a slide can be tapped to view the recognized information or tapped again to view the original ink. When viewing ink, SketchPoint keeps the naturalness of sketching and allows presenters to sketch their ideas freely,
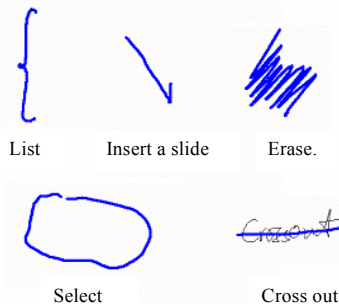

List   Insert a slide   Erase.

Select   Cross out

Figure 14. Gestures used in SketchPoint. The *Erase* gesture is a dense squiggle drawn on top of other strokes. The *Convert* gesture (illustrated in figure 1) is determined by the start and end positions of the stroke.

while in the recognized view this content can be manipulated structurally, e.g., by dragging and dropping items.

We differentiate between a tap and drawing a dot by using the time interval between the tap action and the stroke cluster that is tapped on. If the interval is long enough, SketchPoint interprets it as a tap instead of a dot.

6. STATUS AND FUTURE WORK

SketchPoint was implemented in Java 2 using the Penbuilder toolkit [8]. We developed the algorithms for continuous word segmentation and graph parsing in SketchPoint. We intend to expand our graph-parsing algorithm to support more graph semantics.

We tested SketchPoint with our colleagues to acquire feedback on the usability of its interaction techniques and functionality. They were tested with several simulated tasks. SketchPoint has also been deployed in various setups in our laboratory. We used Wacom LCD tablets (Wacom PL550 and PL300) attached to a 1.3 GHz Pentium 4 PC. We also tested the system with a Wacom Intuous Tablet (9×12) attached to an 800 MHz Pentium III PC with a 17-inch monitor.

This early feedback showed that our participants were interested in using SketchPoint for informal idea communication. They found the automatic transformation from notes to informal slides attractive, and thought that SketchPoint would be helpful for communicating ideas informally in internal settings. We observed that people could easily author an informal presentation.

Besides asking us to fix some stability problems, our participants also asked for search functions and requested that SketchPoint import non-ink content (e.g., images, text, and audio). User feedback also showed us that SketchPoint needs a more intuitive method for creating hyperlinks when the target slide is not visible in the same screen as the start of the hyperlink. We will explore visualization techniques other than zooming to improve the efficiency of this type of operation.

We can easily extend our current system to also support speech annotations. In the future we plan to add this feature along with the playback of time-synchronized ink and speech annotations.

Currently, SketchPoint only supports a limited set of ink structures, such as item-list. We intend to add more structures in the future and to use more explicit and implicit interactions to indicate these structures, e.g., using a pen with different ink colors for different structures.

We are now deploying SketchPoint for use by graduate students and teachers in Kochi University using Tablet PCs and electronic whiteboards as the target platforms. We will conduct long-term evaluations of SketchPoint and collect more feedback from daily usage. We would like to know how SketchPoint impacts idea communication in daily work, how SketchPoint effects existing ways of using whiteboards and how SketchPoint relates to formal presentations.

7. CONCLUSION

Our investigation showed that informal presentations are an important way to communicate ideas, and that existing electronic tools do not support this task well. To address this need, we created SketchPoint, the first tool to support informal presentations, which enables fast idea communication in frequent,

informal situations. SketchPoint allows presenters to quickly author informal slides by sketching the slide content, overall hierarchical structure and hyperlinks of a presentation. The presenter can then easily organize slides in a storyboard and present them directly from the tool. By incorporating note taking in SketchPoint and enabling automatic conversion from notes to slides, we offer a smooth transition from idea capture to idea communication. SketchPoint also enables formal and informal views of ink strokes and allows structure-based manipulation that provides more efficient interaction. SketchPoint's lightweight mechanisms facilitate multiple iterations of idea refinement. An informal evaluation of SketchPoint generated positive feedback from users.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Bailey, B.P., Konstan, J.A. and Carlis, J.V. DEMAIS: Designing Multimedia Applications with Interactive Storyboards. In Proceedings of the ACM Conference on Multimedia, Ottawa, Canada, Sept.30 – Oct.5, 2001, pp.241-248.

[2] Bederson, B.B. and Hollan, J.D. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In Proceedings of the ACM Symposium on User Interface Software and Technology: UIST'94. Marina del Rey, CA. pp.17-26. Nov.2-4, 1994.

[3] Chiu, P. and Wilcox, L. 1998. A Dynamic Grouping Technique for Ink and Audio Notes. In Proceedings of the ACM Symposium on User Interface Software and Technology: UIST'98, 195-202. San Francisco, CA. Nov.6-8.

[4] Davis, R.C., Landay, J.A., Chen, V., Huang, J., Lee R.B., Li, F., Lin, J., Morrey, C.B., Schleimer, B, Price, M.N. and Schilit B.N. NotePals: Lightweight Note Sharing by the Group, for the Group. In Proceedings of the ACM Conference on Human Factors in Computer Systems: CHI'99, Pittsburgh, PA, May 15-20, 1999, pp. 338-345.

[5] Hearst, M.A., Gross, M.D., Landay, J.A. and Stahovich, T.F. Sketching Intelligent Systems, IEEE Intelligent Systems, vol.13, no.3, IEEE, May-June 1998, pp.10-19.

[6] Hong, J.I., Li, F.C. and Landay, J.A. End-User Perceptions of Formal and Informal Representations of Web Sites. Short Paper of the ACM Conference on Human Factors in Computer Systems: CHI'01, Seattle, WA, Mar.31-Apr.4, 2001. 385-386.

[7] Landay, J.A. and Myers, B.A. "Sketching Interfaces: Toward More Human Interface Design." IEEE Computer, vol. 34, no.3, March 2001, pp. 56-64.

[8] Li, Y., Guan, Z., Chen, Y. and Dai, G. Penbuilder: Platform for the Development of Pen-based User Interface, In Proceedings of International Conference on Multimodal Interfaces ICMI'2000, Springer Press, 2000, Beijing, China. Oct. 14-16.

[9] Li, Y., Guan, Z., Ren, X. and Dai, G. SketchPoint: A Smooth Bridge from Note-taking to Presentations, In Proceedings of International Conference on Asia-Pacific Computer-Human Interaction APCHI'2002, Beijing, China. Nov.1-4, 2002.

[10] Li, Y., Guan, Z., Wang, H., Dai, G. and Ren, X. Structuralizing Freeform Notes by Implicit Sketch Understanding, In Proceedings of the Sketch Understanding Symposium of AAAI, Stanford University in Palo Alto, CA. March 25-27, 2002.

[11] Lin, J., Newman, M.W., Hong, J.I. and Landay, J.A. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In CHI Letters: Human Factors in Computing Systems, CHI 2000, 2000. 2(1): pp. 510-517.

[12] Moran, T.P., Melle, W.V. and Chiu, P. Spatial Interpretation of Domain Objects Integrated into a Freeform Electronic Whiteboard. In Proceedings of the ACM Symposium on User Interface Software and Technology: UIST'98, San Francisco, CA, Nov. 6-8, 1998, pp.175-184.

[13] MS PowerPoint, http://www.microsoft.com/powerpoint.

[14] Mynatt, E.D., Edwards, W.K., LaMarca, A. and Igarashi, T. 1999. Flatland: New Dimensions in Office Whiteboards. In Proceedings of the ACM Conference on Human Factors in Computer Systems: CHI'99, 346-353. Pittsburgh, PA, May 15-20.

[15] Nelson, L., Ichimura, S., Pedersen, E.R. and Adams, L. Palette: A Paper Interface for Giving Presentations. In Proceedings of the ACM Conference on Human Factors in Computing Systems: CHI'99, Pittsburgh, PA, May 15-20, 1999, pp.354-361

[16] Raskin, J. The Humane Interface: New Directions for Designing Interactive Systems, Addison-Wesley Pub Co; ISBN: 0201379376; 1 edition (March 24, 2000).

[17] Saund, E. and Moran, T.P. A perceptually-supported sketch editor. In Proceedings of ACM Symposium on User Interface Software and Technology, Marina del, Rey, CA. pp. 175-184, November 2–4 1994.

[18] Sinha, A.K., Shilman, M. and Shah, N. MultiPoint: A Case Study of Multimodal Performance for Building Presentations. Student Posters of ACM Conference on Human Factors in Computer Systems: CHI'01, Seattle, WA, Mar.31-Apr.4, 2001. 431-432.

[19] Shipman, F.M., Marshall, C.C., Moran,T.P. 1995. Finding and Using Implicit Structure in Human Organized Spatial Layouts of Information. In Proceedings of the ACM Conference on Human Factors in Computer Systems: CHI'95, 346-353. Denver, Colorado, May 7-11.

[20] Truong, K.N. and Abowd, G.D. StuPad: Integrating Student Notes with Class Lectures, in CHI'99 Extended Abstracts: Human Factors in Computer Systems, Extended Abstracts, 1999. Pittsburgh, PA: pp.208-209.

[21] Wilcox, L.D., Schilit, B.N. and Sawhney, N.N. 1997. Dynomite: A Dynamically Organized Ink and Audio Notebook. In Proceedings of the ACM Conference on Human Factors in Computer Systems: CHI'97, 186-193. Atlanta, GA.