Intelligent Mind-mapping

Vincent Chik
Department of Electrical and Electronic Engineering
University of Auckland
vchi010@ec.auckland.ac.nz

Beryl Plimmer
Department of Computer Science
University of Auckland
beryl@cs.auckland.ac.nz

John Hosking
Department of Computer Science
University of Auckland
john@cs.auckland.ac.nz

ABSTRACT

Current computer based mind-mapping tools are much slower to use than pen and paper because users are distracted by tool operations such as finding and arranging widgets. The shift in focus from brainstorming to tool management interrupts the rapid brainstorming process that mind maps are intended to support. Our pen based mind-mapping software that includes intelligent ink recognition, editing and export alleviates these intrusions as the user only has to worry about writing on the canvas, yet usual digital document support is provided. The digital ink recognition and manipulation techniques described here will be of interest to others working with informal documents.

Categories and Subject Descriptors

H5.2. [User Interfaces] *Input devices and strategies*

General Terms

Design, Human Factors

Keywords

Mind-map, digital ink, ink reflow, ink recognition, ink grouping

1. INTRODUCTION

A mind-map is a sketchily structured visual representation of one's thoughts which may lead to a train of related ideas. It is based on radiant thinking, a concept which describes how the human brain processes ideas and information, whereby different ideas are associated to each other through relationship hooks [5]. The four main features of a mind-map (figure 1) are as follows:

- Each mind-map has a starting location, the center node that contains the central theme or idea.
- The ideas of the mind-map "radiate" from the central node as branches with sub-nodes connected to each other in parent-child relationships.
- The final structure of the mind-map becomes a hierarchy of linked nodes.
- Each connector/branch has keywords or an image associated with it.

Mind-maps are traditionally hand drawn and used for critical thinking tasks such as strategic planning. They are an effective way of rapidly jotting down and arranging information, affording reinforced association of ideas and recall.

OzCHI 2007 Proceedings, ISBN 978-1-59593-872-5

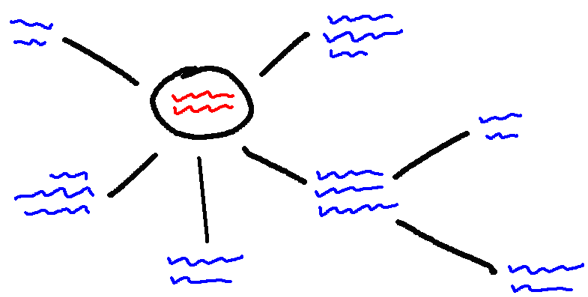

Figure 1: Basic mind-map structure of: A centre node and branches that connect nodes together in a hierarchy.

Compared with normal note taking or brain-storming, mind-maps have several advantages. For instance, time is saved by just noting down relevant key words. Associations between key points are highlighted while passively creating a hierarchy of ideas. Reviewing a mind-map takes considerably less time than to overview a set of written notes as the mind-map is effective in displaying the relevant keywords associated with a particular topic. By providing a visually stimulating environment, the retention of information by the brain is made easier [5].

There are a number of computer applications for mind-mapping. However these are widget-based tools that require the user to select an appropriate node widget or connector before they can enter data. We hypothesize that these tools will adversely affect the idea generation process in the same way as widget-based design tools have been shown to adversely affect the design process [8]. In contrast, this adverse affect of the computer is minimal with sketch-based computer tools [3, 13].

In order to explore the validity of our hypothesis we must first build a computer-based mind-mapping tool that more closely matches traditional pen and paper environments. Sketch-based computer design tools must also provide usual computer editing and archiving support. Here we report on the design and development of our pen-based mind-mapping software.

2. RELATED WORK

As a background to this project we have reviewed the functionality available in current widget-based mind-mapping tools and compared that with the functionality provided in sketch-based tools designed for other tasks. From these we have formulated a list of the technical challenges that this project must address and related research on those topics.

2.1 Mind-mapping Tools

Mind Manager 6 [1] (figure 2) is typical of current mind-mapping tools. The user clicks on the canvas to create a new node. Text can be added to a pre-selected node via the keyboard. Connections are made by dragging nodes on top of other nodes. Once the connection is made the layout is automatically imposed by the software. For example the distance between nodes is restricted and prevents the user from utilizing all the canvas space. A useful function of mind-mapping tools is support for moving a branch and its associated sub-branches to a different location.

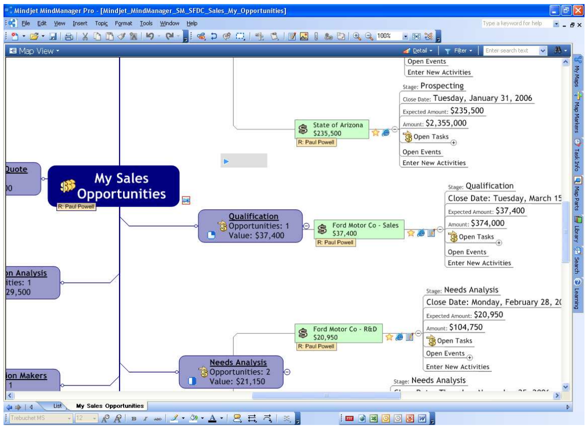

Figure 2: An example of a mind-mapping tool is the Mind Manager 6 Pro.

Although these computer based mind-mapping tools possess many capabilities, much time is consumed with dragging and dropping nodes and connectors from the toolbar before the user can start filling in the nodes with information and form a comprehendible diagram. Research from design domains [3, 8, 13] suggests that users become distracted with arranging the interface as opposed to concentrating on the problem solving process. The advantage of an electronic environment for creating mind-maps over paper is the support for editing, exporting and archiving.

2.2 Sketch Tools

Rough design is often performed with pen and paper. The advantage of using such a medium is the unconstrained drawing space and low cognitive load [8]. There is no restriction as to the layout of a document. Studies comparing pen and paper with widget-based computer tools and sketch-based computer tools have found that using ink is preferable to widgets for design tasks such as user interface design [13], multi-media design and graphic design [3].

There are two attributes of sketches that are important in early design. First, the creation of documents with ink is quicker and using the pen requires less cognitive effort than a widget based design tool. Second the hand-drawn appearance of the ink implies incompleteness, which in-turn suggests that the document can and should be reviewed and changed [12]. While mind-mapping is a problem solving technique rather than a design technique, there is significant similarity between the two tasks that suggest that sketch tools may be more useful than widget based tools.

A pen-based computer tool that recognizes the user's intention intelligently, has the advantages of both computer technology (for editing and so on) and the simplicity of pen & paper to alleviate distractions.

2.3 Challenges

The technical challenges that this project poses consist of three parts: ink recognition and grouping, structural analysis and ink reflow.

- Ink recognition is vital in discerning the elements of a mind-map. It is a precursor to understanding the structure of the mind-map by correctly separating text from drawing. As a part of the recognition, individual ink grouping is required to group related ink strokes into nodes.
- The structure of the mind-map can be established once the nodes and drawing elements have been identified. This structure must be known to support intelligent editing and exporting into other formats.
- Intelligent editing includes the ability to move a branch and its sub-branches to another position on the mind-map. **Ink reflow** is essential when users make adjustments to the mind-map, maintaining the look of the mind-map by rearranging the Ink into a more suitable shape or position.

Existing research suggests techniques we can adopt and modify for ink recognition algorithms [11, 14], grouping [6, 10], structure [7] and ink reflow [2, 4, 9].

3. OUR APPROACH

We have developed a mind-mapping tool for the Tablet PC that is able to recognise text and annotations that are made by the user and treat them as objects as opposed to just a visual representation. This allows the mind-mapping tool to generate an internal logical representation of the user's mind-map with the intent of enabling the user to revise its structure.

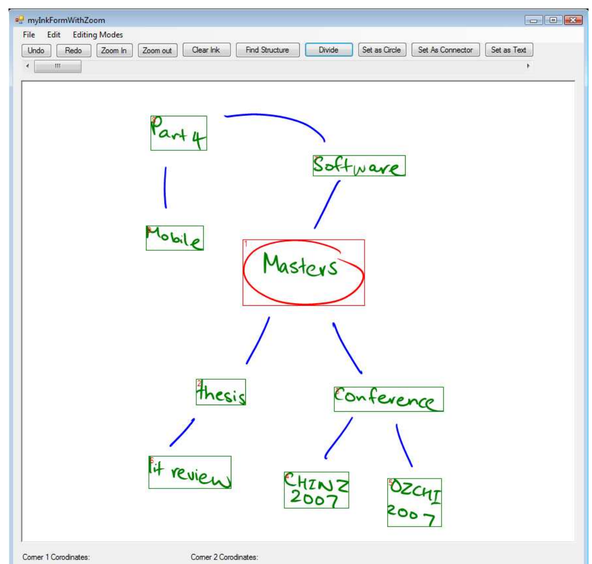

Figure 3: Screenshot of Current Prototype

In addition, in order to allow user flexibility, the system adapts as the user deletes, moves or creates additional data on the existing map. For example, when a user selects a branch of annotations and moves them to another section of the mind-map, the system intelligently reflows the ink to reflect the new orientation and rearranges adjacent branches as necessary to create space for the newly transplanted branch.

3.1 Features

The following are features that were deemed necessary for this mind-mapping application.

Table 1: Mind Mapping Features

Ink recognition	The system must be able to correctly recognize the ink strokes the user writes
Eager Recognition	The system should be able to recognize the text or drawing as the user writes so that editing is continuously supported
Structure Analysis	The system must be able to determine the hierarchical structure of the mind-map
Editing (Undo / Redo)	The system should support the user to add, or delete ink. Also, if recognition errors occur, the system should allow the user to manually change the type of component an ink/group of ink represents.
Move + Reflow	The user must be able to select a connector and be able to move it to another node
Branch Collision Detection	The system should intelligently detect if ink objects overlap other objects when moved
Auto Colour Wheel Segmented	Each connector that branches off the central node can be colour coded relative to its position to the central node
Load & Save	The system should be able to save and reload an existing mind map
Export Capability	The system should allow the user to export the mind map into another format

3.2 System Architecture

The following is a flow diagram of the back end of the system. Each ink stroke drawn by the user is first parsed by the ink divider.

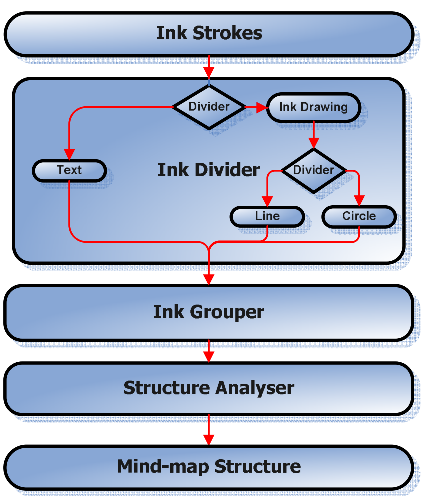

Figure 4: Mind-mapping Tool system architecture

The ink is divided into 3 categories, text, circle or line. Upon stroke division, the ink is then passed into the ink grouper which takes strokes that are geographically close to each other and groups them together as a node. After grouping, a hierarchical order of the nodes is established. As new ink strokes are added the software may alter the categorization of previously entered strokes.

3.3 Ink Divider

The Ink divider is based on the divider from [11] to separate text and drawing ink. The divider looks at specific features of each ink stroke and categorizes them using a binary tree structure. We replaced the bounding box width feature specified in [11] with the ink bounding box diagonal as mind-maps differ from the diagrams used in their training set, mind-maps frequently use slopping lines for connectors. Once the ink stroke is classified as text stroke or drawing, text is passed to the OS recognizer in preparation for export to other formats. The drawing strokes are then separated again into lines or circles by a filter [14]. For a circle, the start and end points of the stroke are generally in close proximity as opposed to a line. The bounding box height and width are again somewhat similar for a circle while a line stroke may not be. The filter consists of two checks; the distance between the start and end points of the stroke is checked to see if it is below 1200 hi metric units and that the ratio of the longest bounding box side and the distance between the start and end points of the stroke is less than 0.5. This ensures ellipses are also categorized as a circle. The recognized writing and drawing strokes are then piped to the ink grouper.

3.4 Ink Stroke Grouping

The ink grouper takes the recognized strokes and groups them into coherent entities of node or connector. Ink classified as lines and circles are automatically categorized as connectors and nodes respectively. All strokes that reside in a circle are automatically grouped together and are put into the same node as the circle. Remaining text strokes must be grouped together into words and groups of words to form uncontained nodes. Text strokes are grouped by examining the top, bottom and sides of each stroke's bounding box. If the bounding boxes top values are within 200 hi metric units of each other and the sides are within 500 hi metric units, the strokes are grouped together. We have arrived at these values with informal testing and will check their validity as a part of our usability testing. At this point the system has logical presentation of nodes and connections between the nodes but lacks the hierarchical structure.

3.5 Mind-map Structure

The mind-map is sorted hierarchically. The first node that is drawn is typically the central node and all other nodes connected to it via connectors are its sub nodes. On this basis, the system implements a recursive algorithm, starting by finding the sub nodes of the central node. This is carried out by finding the branches linked to the main node. The first branch is determined by the first connector that is drawn and its subsequent branches are found until there is none left before coursing down the next branch linked with the main node. With recursion down the tree, the complete structure of the mind-map can be found. With this structure, the mind-map can be easily exported into ordered digital formats. At times people create mind-maps with cross connections. This causes our recursive algorithm to loop endlessly. Hence a checklist was implemented that records the nodes that the system has traversed and skips the nodes already analysed.

3.6 Editing and Reflow

To support the relocation of a branch to another part of the mind-map basic reflow has been implemented. The relocated nodes repel other nodes away should they overlap. The area around the relocated node's bounding box is divided into 8 sectors. A starburst metaphor is used to move existing nodes. The relative positions of the centre point the relocated and existing nodes determines the movement direction. The distance nodes are pushed back is calculated by the overlap amount. The connectors that link the nodes are then elongated or compressed. The amount of compression and elongation is found by the distance and direction the node was shifted and the connector transformed and rotated accordingly.

4. CONCLUSIONS

Sketch-based mind-mapping tools offer an alternative to paper or widget-based tools. Our first prototype presented here concentrates on text nodes to reduce recognition problems. It can formulate a hierarchy of nodes and supports basic restructuring. We have conducted some informal evaluations and participants have commented favourably on the similarity of the tool to mind-mapping on paper. Some recognition errors occurred, which annoyed them, but they were satisfied with the functionality provided to correct misclassified ink.

Upon completing the initial system, a formal usability study will be undertaken and a second prototype developed. The mind-mapping tool can then be compared with other computer/paper oriented mind-mapping techniques to test our hypothesis on the affect of tools on mind-mapping.

5. ACKNOWLEDGMENTS

We would like to thank Microsoft Research Asia for their support of this project.

6. REFERENCES

[1] Mindjet: Software for Visualizing and Using Information, Retrieved on 27th June 2007 from:

http://www.mindjet.com

[2] Arvo, J., Novins, K., Appearance-preserving manipulation of hand-drawn graphs, Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, ACM Press, (2005),

[3] Bailey, B. P., Konstan, J. A., Are Informal Tools Better? Comparing DEMAIS, Pencil and Paper, and Authorware for Early Multimedia Design, in proc CHI 2003, ACM, (2003), 313-320

[4] Bargeron, D., Moscovich, T., Reflowing digital ink annotations, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, (2003),

[5] Buzan, T., Buzan, B., The Mind Map book, BBC Worldwide Limited, (2000), 269

[6] Chung, R., Mirica, P., Plimmer, B., InkKit: a generic design tool for the tablet PC, Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural, ACM Press, (2005),

[7] Freeman, I. J., Plimmer, B., Connector semantics for sketched diagram recognition, Proceedings of the eight Australasian conference on User interface - Volume 64, Australian Computer Society, Inc., (2007),

[8] Goel, V., Sketches of thought, The MIT Press, (1995),

[9] Golovchinsky, G., Denoue, L., Moving markup: repositioning freeform annotations, Proceedings of the 15th annual ACM symposium on User interface software and technology, ACM Press, (2002),

[10] Moran, T. P., Melle, W. v., Chiu, P., Spatial interpretation of domain objects integrated into a freeform electronic whiteboard, Proceedings of the 11th annual ACM symposium on User interface software and technology, ACM Press, (1998),

[11] Patel, R., Plimmer, B., Grundy, J., Ihaka, R., Ink Features for Diagram Recognition, Sketch Based Interfaces and Modeling IEEE, (2007),

[12] Plimmer, B., Apperley, M., Evaluating a sketch environment for novice programmers, CHI '03 extended abstracts on Human factors in computing systems, ACM Press, (2003),

[13] Plimmer, B. E., Apperley, M., Software for Students to Sketch Interface Designs, in proc Interact, (2003), 73-80

[14] Priest, R., Plimmer, B., RCA: experiences with an IDE annotation tool, Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI, ACM Press, (2006),