

Levels of Formality in Diagram Presentation

Louise Yeung¹, Beryl Plimmer², Brenda Lobb¹, Douglas Elliffe¹

Department of Psychology¹, Department of Computer Science²

University of Auckland, New Zealand

Phone +64 9 373-7599

l.w.s.yeung@gmail.com, {b.plimmer|b.lobb|d.elliffe}@auckland.ac.nz

ABSTRACT

The incremental beautification of hand-drawn diagrams is a process that is poorly understood. Thus implementation of beautification techniques in computer-based sketch tools is ad hoc, with most only supporting the ends of the spectrum: hand-drawn and fully formalized. Hand-drawn diagrams are more effective for early design and review but users are more satisfied with formal designs. This suggests that there may be applications for intermediate levels of formality. By understanding the attributes of visual formality it is possible to beautify a diagram progressively, thereby achieving visually consistent intermediate levels of formality. Here we present a taxonomy of the attributes of visual formality and the implementation of this taxonomy into a sketch tool.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Prototyping, User Centered Design

General Terms

Design, Human Factors

Keywords

Beautification, sketch tools, prototyping

1. INTRODUCTION

Computer-aided design tools have been developed to help designers to create and maintain detailed drawings. The goal, when creating a design with these tools, is a formal, detailed, complete and accurate representation of the intended product. The design, whether it is for a consumer product, building, user interface, database or engine, must adhere to the syntactic and semantic rules of the particular diagram type, as it is these rules that ensure the completeness and accuracy of the description. Increasingly, diagrammatic models are being used to generate a prototype, or in the case of software models, the actual system. Consequently, computer-based diagramming is an area of interest across a wide range of disciplines.

Before finishing a design a final prototype is constructed. Depending on the discipline, this might be a non-functioning computer user interface, a cardboard model of a building or a fully functioning car. Regardless of the rendering, the prototype is generally a physical representation of the product that can be used

to check functional and interaction requirements or market response.

Earlier again in the design process, less well-formed prototypes may be used for the designers to get a feel for the product and explore the functional requirements. Paper prototypes, or rough hand-drawn interfaces that are used to 'play computer' with, [31] have been found to be particularly effective for user interface design. Similar simple models are used in other disciplines.

Earlier still, before any prototype is created, design ideas exist only as imagery in the designer's mind [15]. However human short-term or working memory is quite limited in capacity, so designers generally offload their design ideas as hand-drawn sketches. Sketching is used universally as the preferred method of early design expression and recording.

Sketching has been identified as one of the most important design activities, facilitating such important aspects of design as reasoning [1, 15, 35], problem-solving [32], memory and thinking [33], creativity (e.g. [13, 14, 21]) and communication [4, 34, 36], all of which affect the design outcome [30].

Paper-and-pencil, the traditional sketching toolset, is inexpensive and provides a medium for designers to explore different design ideas [22]. In some situations, however, paper prototypes are insufficient to evaluate a particular design idea fully. For example, a computer interface requiring rapid feedback to users (e.g. Web site design), or complex, dynamic visualization (e.g. engineering and architectural design) usually requires software prototypes, created using computer-aided design tools [4].

Further, paper imposes physical limitations on the artifact. Electronic documents provide better support for editing, version control and sharing. Thus designers can benefit from digital document support for sketching. Unlike paper sketches, digital copies of sketches can be used by recognition and beautification algorithms to transform the hand-drawn sketch into a formal diagram automatically.

Computerization of sketch tools makes it possible to tidy (beautify) the sketch electronically. While a number of sketch tools incorporate beautification, to date there has been no thorough exploration of the concept or dimensionality of formality to guide development and appropriate application of these techniques. This study aims firstly, therefore, to develop a taxonomy of design formality and secondly to realize this taxonomy into a sketch tool.

2. BACKGROUND

Computer-supported informal or sketch-based design tools are being investigated as an alternative to paper or CAD tools across a wide range of disciplines [3, 9, 11, 18, 26]. These tools are seen to bridge the gap between traditional design media (e.g., paper and

OzCHI 2007, 28-30 November 2007, Adelaide, Australia. Copyright the author(s) and CHISIG. Additional copies are available at the ACM Digital Library (<http://portal.acm.org/dl.cfm>) or can be ordered from CHISIG(secretary@chisig.org)

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

pen) and computer-aided design (CAD) software by providing not only a designer-friendly computer-supported sketching environment, but also normal document-management functionality for editing and version control. Many of these sketch tools also include recognition engines to transform the sketch into a formal representation. Some also include beautification: a process by which the pen input is transformed into a more formal, regular representation (e.g. lines are straightened and set at the exact horizontal or vertical).

As opposed to “formal” design tools such as CAD and interface-design tools in programming IDEs (e.g., Visual Studio), sketch-based design tools are regarded as “informal” design tools [19] because they support the ambiguity and informality of sketching [8, 22, 23]. The advantages of sketches over formal computer design tools for constructing new designs were first recorded by Black [5]. Subsequent studies have supported her hypothesis with sketches resulting in better designs [2, 13]. Likewise Wong [39], a computer interface designer, observed in 1992 that she got more appropriate feedback from other team members if she showed them a hand-sketched design. She reported that the usual discussions, when she presented a formal computer-rendered design, were about unimportant details like the color or font. In contrast, a sketch elicited more appropriate discussion on the functional requirements. Subsequent studies comparing reviews of designs presented as either fully formal or informal, hand-drawn diagrams have confirmed her observations [27, 37].

Plimmer and Grundy’s [28] discussion of beautification issues in computer-supported sketch-based design tools has illustrated different ways of supporting beautification, as well as user-interface design, by implementing FreeForm [27] and SUMLOW [7] for testing and evaluation purposes. Plimmer and Grundy further identified issues associated with sketch-time beautification (beautifying sketch content as the user draws) and delayed, user controlled beautification (when sketched content is beautified as required), and identified the system requirements to support each approach to beautification.

SUMLOW [7] maintains two visualizations of the UML model (Figure 1). FreeForm [27] takes a different approach for user interface design: the sketch is retained as is, but a formal visual basic form can be generated from the sketch (Figure 2).

Some other sketch tools have included a variety of other beautification techniques. For example Knight [10], has multiple renderings of UML class components (Figure 3). Denim [22], a web site design tool, immediately recognizes simple symbols such as rectangles and lines. This tool varies the degree of beautification applied according to the zoom level. For example, at storyboard level a line drawn to indicate navigation between pages is smoothed and has a dot added to the source point and arrow to the destination point while at page level it is rendered as raw ink.

A different approach is to beautify input immediately. Igarashi et al. [16] assumed that pen strokes are intended to be either straight lines or regular curves and so their algorithm immediately transforms pen strokes into line segments. They apply constraints so that lines lie at fixed angles and connections and intersections are exact. If the intention of the user is ambiguous, the system presents multiple alternatives from which the user can choose.

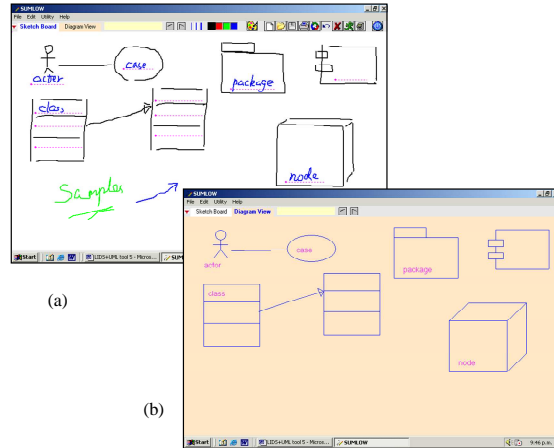


Figure 1. Formal and informal representation from SUMLOW [7]

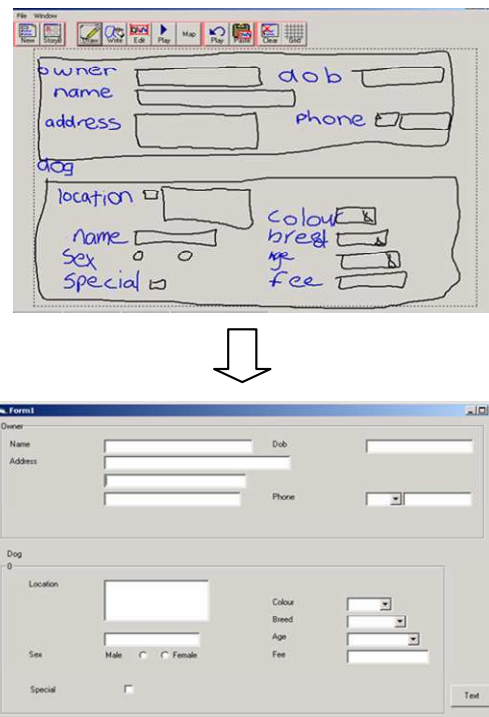


Figure 2. FreeForm sketch and VB form [27]

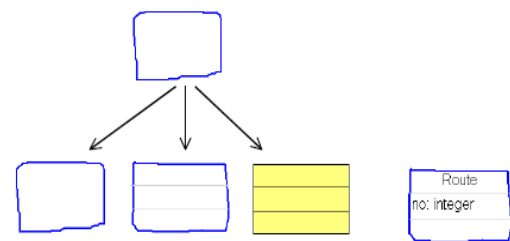


Figure 3. Different renderings of a class component in Knight from [10]

Wang et al. [38] took a similar approach to immediate beautification of pen strokes, using a grid. Each line segment was assumed to start and finish on a grid intersection and the line was transformed into a straight-line, half circle or circle depending on the pen path. Handwriting can be similarly transformed [29] Figure 4.

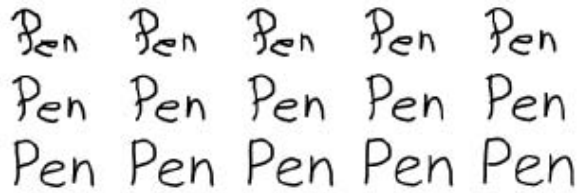


Figure 4 Smooth Morphing of Text [29]

It is clear from the various comparative studies that sketching and hand-sketched representations are better artifacts to work with at the early stages of design. However, people continue to express a preference for [25] and spend time on [6] beautifying

representations. Bolz [6] claimed that fifty percent of the total time spent on creating formal designs on a computer is on beautification operations such as aligning and sizing the components, hence the various efforts to beautify hand-drawn sketches that are found in sketch tools. It may be possible to satisfy both goals by partially beautifying a sketch or re-rendering a formal design to make it look hand-drawn [17]. To do this requires understanding of the relationships between attributes of design components and formality. As a first step, we are developing a taxonomy of the attributes of design components

3. Taxonomy

Consider a sketched design for a user interface, such as is shown in Figure 5a, compared with a formalized representation of the same design (Figure 5d). The components used in the website design - words, textboxes, dropdown lists, radio-button checkboxes and labels - are the same in both representations but the attributes of those components - smoothness of the lines, relative sizes, alignment of the components, spacing between components and lastly, the appearance of the words - differ.

Table 1 Attributes of design components which may be beautified

	Hand-drawn	Low-level Formality	Medium-level Formality	High-Level Formality
Smoothness				
Size				
Alignment horizontal				
Alignment vertical				
Spacing horizontal				
Fonts				

In the sketch the words are hand-written letters, each a little different and misaligned, while the font used in the formal diagram has each letter perfectly formed and aligned as a type-face font. As the attributes increase in smoothness and move from variable to more standardized size, spacing and alignment, the visual image changes its appearance from an informal hand-drawn sketch to a formal diagram. Table 1 is a taxonomy of the attributes of design components which can be beautified and examples of each attribute at four levels of formality.

Consider the images in Table 1: we made each set to demonstrate a particular aspect of beautification. However there are likely to be interactions between the different processes of beautification. For example, in order to apply alignment to the differently-sized components in the “size” example, a decision would need to be made as to whether to align their tops, middles or bottoms. This alignment of itself would not have much effect on the visual tidiness of a larger diagram. To implement multi-set beautification, decisions must be made about when and how to apply changes to each attribute of each component

4. IMPLEMENTATION

The beautification of designs through changes to attributes of components as described above has been realized in InkKit [12, 24]. InkKit is a software toolkit for creating sketched diagrams with a pen on a tablet PC. The InkKit recognition engine is used to identify the sketched components, *Component* is the term used for a logical unit of the sketch. For example, the components on the user interface examples in this paper are: textbox, dropdown list, radio-button checkbox and label. After the recognition, component type, size and position data are available. After recognition the beautification can be applied selectively from a menu. For each attribute, up to four levels of representation ranging from hand-drawn to formal have been implemented. Thus a diagram can be presented at four different levels of formality; hand-drawn, low-level beautification, medium-level beautification and high-level beautification (fully formalized).

Ink stroke smoothing is thought to be the attribute most critical to changing the visual appearance; it also interacts directly with all the other attribute changes. Smoothing can be applied at three levels. To smooth an ink stroke, first the type of shape is identified as a line, polygon or ellipse. Using the smallest external rectangle as a guide, the bounding box, points are identified for each type of shape, end points of the line, corners for the polygon and the centre of the circle. From these points a perfect, regular shape could be scribed. However, at low- and medium-level beautification the stroke is morphed from its current position to 1/3 or 2/3 of the way to the ‘perfect’ stroke. For high-level beautification it is morphed to the perfect stroke. Table 1 Smoothing, shows the effect of this on a rectangle, triangle and circle.

Size is standardized using the bounding box of the components. All components of the same type are grouped and an average size calculated and normalized across the group. All members of the group are then resized so that their bounding boxes are the same size. An example of the application of this to textboxes, dropdown lists and radio buttons is shown in Table 1 Size.

Horizontal and vertical alignment can be applied separately to a diagram. Both use the same techniques. For horizontal alignment

first each component is grouped with other components that are in approximately the same row (these approximations are calculated using mid-points and top/bottom extremities). Then an average bottom point is calculated and all the members of the group are moved to that horizontal position. For vertical alignment the same process is applied with grouping by column and the bottom point replaced by the left point. As this process can result in components obscuring or overlapping each other, spacing is applied as a part of the same process.

Because of the interaction of the effects of size, alignment and spacing on formality, these three attributes are applied at an atomic level i.e. the attributes are aligned or they are not, the sizes are standardized or they are not. Changes in alignment and spacing are applied together as these attributes are not as critical to the visual appearance of a sketch as smoothness. Further smoothing can result in inconsistent size, alignment and spacing which may require correction.

The appearance of writing (along with smoothing) is thought to be critical to the visual appearance of the diagram. However writing is much more difficult to beautify automatically [29]. We replace hand-writing, with increasingly formal fonts rather than morphing the writing strokes. The low-level formalization uses a font created using the Tablet PC My Font tool [20]. With this tool a handwritten example of each letter is converted into a font with the spacing between letters and words adjusted appropriately. The effect is that letters are handwritten but each example of a letter is identical – this results in a tidier appearance. The font for medium-level formality is Gulim, chosen because it is an uncommon sans-serif font. Sans-serif fonts are considered less formal in appearance than serif fonts. For high-level formality we selected the classic Times Roman font for its serifs, association with printed books and regular appearance.

5. APPLICATION

By applying these changes in attributes of the design components to a hand-drawn sketch in a consistent manner, the sketch is systematically transformed from an untidy hand-drawing to a fully formal diagram.

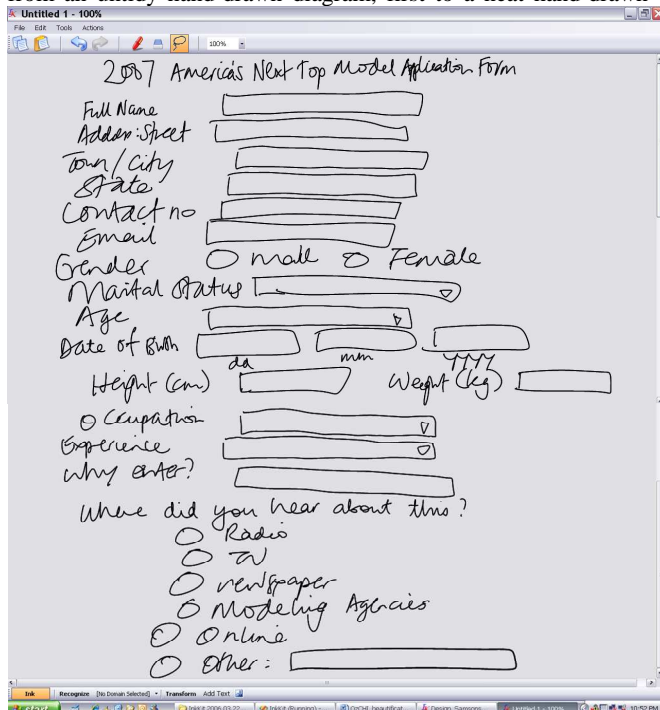
The designs shown in Figure 5 demonstrate the visual effect of the four different levels of representation. a) is hand-drawn into InkKit on a Tablet PC. b) has had low-level beautification applied. Notice that the handwriting has been replaced with the handwritten font – while it still looks handwritten, the font sets the writing on a horizontal baseline and spaces the letters evenly. Low-level smoothing, and horizontal alignment and spacing, have also been applied. The result is particularly noticeable in the first sets of radio buttons. c) has had medium-level beautification applied. The font has been changed to Gulim, which is still an informal font but has the appearance of a computer-produced font rather than of handwriting. Another step of smoothing has been applied to the lines and the component sizes have been approximately standardized. These operations affect the horizontal alignment so that has been recalibrated.

In addition, words have been vertically aligned but drawing components remain in their original vertical position. In d) the diagram has been fully beautified. The font has been changed to Times Roman, all the components are regular shapes and sizes,

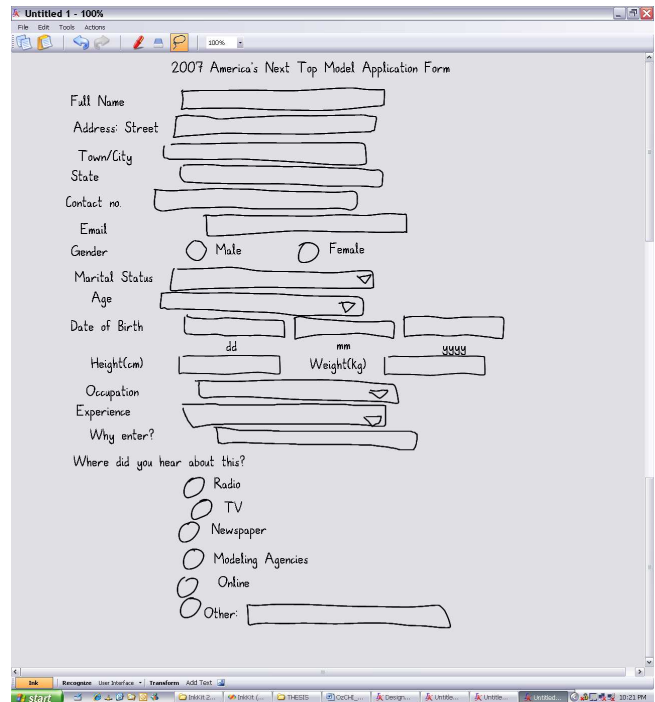
vertical and horizontal alignment and spacing has been fully applied across the entire diagram.

Quickly scanning the diagrams in Figure 5 one can observe the visual effect of this beautification process. The diagram changes from an untidy hand-drawn diagram, first to a neat hand-drawn

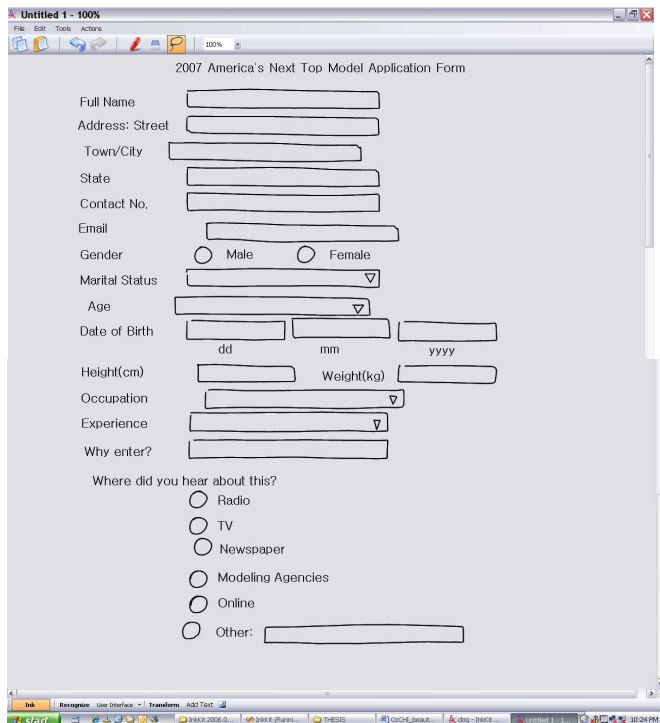
diagram, then to a diagram with the appearance of an informal computer-constructed diagram and finally to a fully formal computer-constructed diagram.



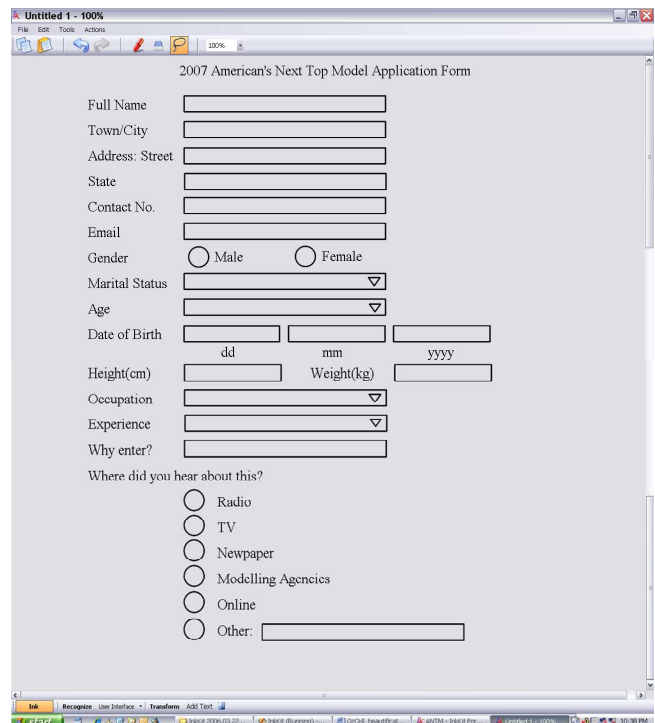
a) Hand-drawn



b) Low-level Formality



c) Medium-level Formality



d) High-level Formality

Figure 5 User interface designs at four levels of formality

6. DISCUSSION

Sketches are regularly used as the first recording method for designs. In contrast, formal computer-based designs are preferred for final product design and are increasingly used directly to generate products. Traditionally paper and pencil were used for initial designs; these tools, however, lack the editing and archiving functionality of digital documents. However, the current generation of computer design tools has been shown to be obstructive to the early design process. Sketch-based design tools have been proposed to bridge this gap.

Computer-based sketch tools provide the computational support for editing and archiving expected of any document management tool. Comparative studies have shown that they do not have the detrimental effect on the design process that has been observed with formal design tools [27, 37]. With an appropriate recognition engine and computational support, these tools can convert the hand-drawn diagram into a formal and operational model.

A number of tools have implemented some beautification of the diagram. Some have immediately morphed the user's pen strokes so that the diagram is never shown as a sketch while others have had two views of the diagram – one sketched and the other a formal representation. Immediate beautification may be useful if the only reason for using a sketch tool is speed – some studies have suggested that a formal diagram can be constructed more quickly with a sketch tool [6, 25]. However, if the purpose is to use the diagram as a design artifact, then given the consistent evidence that a sketch is a better artifact to work with during design, immediate beautification of the sketch is counter-productive.

The research comparing performance with sketches and formal diagrams is compelling, consistently showing that users perform better with sketched representations. Yet users continue to prefer formal representations, particularly if the diagram is being presented to superiors or clients. Partial beautification of the sketches may satisfy both needs, by producing a tidy hand-drawn look and feel. To achieve this it is necessary to untangle the various attributes of diagram components that contribute to formality so that beautification can be systematically applied.

Here we have presented a taxonomy of these attributes. We implemented beautification of each of these attributes in InkKit. The changes to attributes could be applied individually, but this would result in an inconsistent look and feel to the diagram. We explored various combinations to produce the two intermediate levels of beautification shown in Figure 5. Our experimentation suggested to us that the attributes that had the most effect on appearance were the smoothness of the lines and font. We implemented three steps of beautification to produce four levels of formality for these attributes. Our informal evaluations suggested that the most critical attributes for visual formality are smoothness of lines and fonts. Size, horizontal and vertical alignment and spacing seem to be less critical. Also, from a practical perspective there is an interrelationship between these elements that needs to be considered to avoid occlusion of components. Therefore, horizontal alignment and spacing were applied at low-level formality, and vertical alignment and spacing, and size standardization at the medium-level.

We chose to apply the beautification in specific stages. An alternative approach would be to have continuous levels of beautification where the user can set the beautification level to

n%. In order to do this successfully, more investigation is needed to define the order and speed of beautification of each of the relevant attributes, because the interaction between alignment and spacing would need to be resolved continuously. Furthermore, the writing would need to be continuously morphed [29], rather than changed to a different font in an all-or-none manner, so that there was a smooth transition between each point on the continuum.

7. CONCLUSIONS

Beautification of computer-based sketches is a natural extension to sketch tools that has been incorporated in a number of these tools. However, most support only the total beautification of the shape components and do not address text beautification. Here we have described the attributes of sketches that contribute to formality and how each can be adjusted progressively to obtain intermediate levels of formality. We have implemented these techniques within InkKit, a sketch toolkit to apply the taxonomy in a practical setting. By careful order of application the visual appearance of the sketch moves from an untidy hand-drawn diagram to a formal diagram.

It is clear that sketches are better than formal diagrams as artifacts during early design but there has been no evaluation of partially beautified diagrams. The next step of this project is to evaluate the effect of the various levels of formality on the design process. Supporting continuous beautification is also an area worthy of further investigation.

8. REFERENCES

- [1]. Atkin, O., *Psychology of architectural design* 1986, London: Pion.
- [2]. Bailey, B.P. and J.A. Konstan. *Are Informal Tools Better? Comparing DEMAIS, Pencil and Paper, and Authorware for Early Multimedia Design*. in *CHI 2003*. 2003. Ft Lauderdale: ACM.
- [3]. Bailey, B.P., J.A. Konstan, and J.V. Carlis. *DEMAIS: Designing Multimedia Applications with Interactive Storyboards*. in *ACM Multimedia*. 2001.
- [4]. Beaudouin-Lafon, M. and W. Mackay, *Prototyping tools and techniques*, in *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. 2003, Lawrence Erlbaum Associates. p. 1006-1031.
- [5]. Black, A., *Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers*. Behaviour and information technology, 1990. **9**(4): p. 283-296.
- [6]. Bolz, D. *Some aspects of the user interface of knowledge based beautifier for drawings*. in *Intelligent user interfaces '93*. 1993.
- [7]. Chen, Q., J.C. Grundy, and J.G. Hosking, *SUMLOW: Early Design-Stage Sketching of UML Diagrams on an E-whiteboard*. Software Practice and Experience accepted November 2006., in press.
- [8]. Chung, R., P. Mirica, and B. Plimmer. *InkKit: A Generic Design Tool for the Tablet PC*. in *CHINZ 05*. 2005. Auckland: ACM.
- [9]. Damm, C.H., K.M. Hansen, and M. Thomsen. *Tool support for cooperative object-oriented design: Gesture based modelling on and electronic whiteboard*. in *Chi 2000*. 2000: ACM.

- [10]. Damm, C.H., K.M. Hansen, et al. *Supporting Several Levels of Restriction in the UML*. in *UML 2000*. 2000. York, UK.
- [11]. Do, E.Y.L. and M. Gross, *Thinking with Diagrams in Architectural Design*. *Artificial Intelligence Review*, 2001(15): p. 135-149.
- [12]. Freeman, I. and B. Plimmer. *Connector Semantics for Sketched Diagram Recognition*. in *AUIC*. 2007. Ballarat, Australia: ACM.
- [13]. Goel, V., *Sketches of thought*. 1995, Cambridge, Massachusetts: The MIT Press.
- [14]. Goldschmidt, G., *The dialectics of sketching*. *Creative Research Journal*, 1991. **4**(2): p. 123-143.
- [15]. Goldschmidt, G., *Serial sketching: Visual problem solving in design, The backtalk of self-generated sketches*. *Cybernetics and Systems*, 1992. **23**: p. 191-219.
- [16]. Igarashi, T., S. Matsuoka, et al. *Interactive Beautification: a technique for rapid geometric design*. in *UIST 97*. 1997. Banff.
- [17]. Insight Development Corp. *Squiggle*. 2000 [cited 2002; 3.0:[Available from: <http://www.residential.com/squiggle.html>]
- [18]. Landay, J., *Informal user interfaces for natural human-computer interaction*, in *IEEE Intelligent Systems*. 1998. p. 14-16.
- [19]. Landay, J. and B. Myers, *Sketching Interfaces: Toward more human interface design*. *Computer*, 2001. **34**(3): p. 56-64.
- [20]. Microsoft Corporation. *My Font Tool for Tablet PC*. 2004 [cited 2007 20 May]; Available from: <http://www.microsoft.com/windowsxp/downloads/powertoys/tabletpc.mspx>.
- [21]. Nakakoji, K., A. Tanaka, and D. Fallman. "Sketching" nurturing creativity: commonalities in art, design, engineering and research. in *CHI '06 extended abstracts on Human factors in computing systems*. 2006. Montreal, Quebec, Canada: ACM Press.
- [22]. Newman, M.W., J. Lin, et al., *DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice*. *Human-Computer Interaction*, 2003. **18**(3): p. 259-324.
- [23]. Plimmer, B. and M. Apperley. *From Sketch to Blueprint: Supporting the creative design process*. in *Workshop on improving and assessing pen-based input techniques*. 2005. Edinburgh.
- [24]. Plimmer, B., G. Tang, and M. Young. *Sketch Tool Usability: Allowing the user to disengage*. in *HCI 2006*. London: ACM.
- [25]. Plimmer, B.E., *Using Shared Displays to Support Group Design; A Study of the Use of Informal User Interface Designs when Learning to Program*, in *Computer Science 2004*, University of Waikato: Hamilton
- [26]. Plimmer, B.E. and M. Apperley. *Freeform: A Tool for Sketching Form Designs*. in *BHCI*. 2003. Bath.
- [27]. Plimmer, B.E. and M. Apperley. *Software for Students to Sketch Interface Designs*. in *Interact*. 2003. Zurich.
- [28]. Plimmer, B.E. and J. Grundy. *Beautifying sketch-based design tool content: issues and experiences*. in *AUIC 2005*. 2005. Newcastle: Australian Computer Society.
- [29]. Pomm, C. and S. Werlen. *Smooth morphing of handwritten text*. in *AVI '04*. 2004. Gallipoli: ACM.
- [30]. Purcell, A.T. and J.S. Gero, *Drawings and the design process* *Design Studies*, 1998. **19**(4): p. 389-430.
- [31]. Rettig, M., *Prototyping for tiny fingers*. *Communications of the ACM*, 1994. **37**(4): p. 21-27.
- [32]. Romer, A., S. Leinert, and P. Sachse, *External Support of Problem Analysis in Design Problem Solving*. *Research in Engineering Design*, 2000. **12**(3): p. 144-151.
- [33]. Scaife, M. and Y. Rogers, *External cognition: how do graphical representations work?* *International Journal of Human Computer Studies* 1996. **45**(2): p. 185-214
- [34]. Schenk, P., *The role of drawing in the graphic design process*. *Design Studies*, 1991. **12**(3): p. 168-181.
- [35]. Tversky, B. *What does drawing reveal about thinking*. in *Visual and spatial reasoning in design*. 1999. Cambridge, Mass.
- [36]. van der Lugt, R., *How sketching can affect the idea generation process in design group meetings* *Design Studies*, 2005 **26**(2): p. 101-122.
- [37]. Walker, M., L. Takayama, and J.A. Landay. *High-fidelity or low-fidelity, paper or computer medium?* in *Human Factors and Ergonomics Society*. 2002. Baltimore.
- [38]. Wang, B., J. Sun, and B. Plimmer. *Exploring sketch beautification techniques*. in *CHINZ 05*. 2005. Auckland: ACM.
- [39]. Wong, Y.Y. *Rough and ready prototypes: Lessons from graphic design*. in *Human Factors in Computing Systems CHI '92*. 1992. Monterey.