A Handwriting Tool to Support Creative Activities

Kazuo Misue and Jiro Tanaka

Graduate School of Systems and Information Engineering,
University of Tsukuba,
1-1-1 Tennoudai, Tsukuba, 305-8573 Japan
{misue, jiro}@cs.tsukuba.ac.jp

Abstract. Handwriting in computer environments satisfies many requirements that are necessary to support creative activities, as it is easy to use, natural, flexible, and informal. This article proposes a tool to support creative activities using handwriting. The tool has the following features: (1) It is targeted for use in processing diagrams with logical structures. (2) It provides a handwriting input interface. (3) It maintains the handwritten feel of diagrams. (4) It supports creative activities intellectually and actively. The architecture and the behavior of the tool are explained using some screen shots of a prototype.

1 Introduction

Computers augment human creativity with their information processing capability. Many tools have been developed to support human creative activities [1]. Most of them are focused on providing a variety of advanced functions.

However, human creativity is very delicate. It contains some factors that we cannot control by ourselves. Most tools developed so far seem to have given careless consideration to the aspects of feelings and impressions, which may influence human creative activities. As a result, designers often use paper (or a whiteboard) and a pen in the early stages of their creative activities even though a lot of tools with advanced functions are available [2].

Recently, user interfaces of computers have been changing into forms that are appropriate for activities in the real world. For example, pens (or styluses) and large screen displays are becoming popular. Such interfaces have enabled several kinds of handwriting, such as a pen as an input device and sketching as an input method.

Handwriting in computer environments satisfies many creative needs by being easy to use, natural, and flexible. Additionally, the informality provided by handwriting seems to benefit human creative activities. Therefore, we propose a tool to support creative activities that exploits the features of handwriting. The tool is especially targeted to support creative activities involving logical structures.

2 Use of Handwriting to Support Creative Activities

2.1 Externalize Information Using Drawings

In this article "creation" is defined as making something that is conceptually new. Examples of creative activities include developing a computer system, making web sites, constructing a building, and so on. Such creative activities contain operations of concept in their upstream. It is important to organize and/or to integrate information that is disorganized or fragmentary. We may do such operations in our head when there are only a few concepts to be handled. However, we often externalize information in our head using drawings; we express concepts with words, symbols, or diagrams, and connect them with lines, enclose them with circles, and so on. Drawing diagrams with logical structures and looking at them objectively accelerates the creative activities, especially when there are a lot of concepts, as well as complex relationships between them.

2.2 Requirements for Tools to Support Creative Activities

The externalizing processes in creative activities do not occur only once but require a lot of trial and error repetition. Pen and paper are not very convenient for repetition, so computers are the logical choice to support such processes. Therefore, it is important to develop tools that support externalization efficiently and appropriately.

When we get an idea, we often forget it soon after. The idea is often vague and ambiguous. It is rarely a complete idea; usually it is partial and imperfect. Therefore, it is important to write down such fragile, vague, partial, or incomplete information efficiently. Moreover, it is also important to write it down appropriately. Excluding ambiguity from the information and converting it into a more formal style for computer input might cause us to lose the intrinsic point of view, resulting in a misunderstanding. Consequently, it is necessary to achieve the following objectives when designing tools to support creative activities.

- To be able to write down ideas quickly and easily.
- To be able to write down ideas without having to deal with useless or indirect operations.
- To be able to write down ideas freely in the order we get them.
- To be able to write down vague, partial, or incomplete ideas appropriately.

2.3 Handwriting in Computer Environments

We think that handwriting in the computer environment satisfies some of the following requirements for input devices, input methods and expression forms.

i) **Input device**: Pen-type devices are used for input. Styluses for PDAs or tablet PCs are pen-type devices that are widely used. Typical monitors used with these devices are liquid crystal or plasma screens that have touch panels.

ii) **Input method**: Sketching or gestures are used as input methods. Sketching adopts the tracks of a pointing device as ink data. Gestures are similar to sketching but interpret tracks of pointing devices as commands.

iii) **Expression form**: Curves drawn freehand or that seem to be drawn freehand are used to express forms. This requirement depends on the character of the expression. Thus, the expression of the so-called "handwritten style" also satisfies this requirement.

2.4 Merits of Handwriting in Supporting Creative Activities

Handwriting offers several merits in supporting creative activities. Here we explain the merits according to the requirements described above.

i) **Pen-type device (input device)**

Easy: Pen-type devices are simple to use; they are similar to normal ink pens that people are used to using, so they are easy to operate. This ease of use enables quick input of ideas.

Natural: Input is very natural using a pen. We can input directly at the operating position of the device and can move the operating position quickly and directly to a target position on the screen. Such features also contribute to quick input of ideas.

ii) **Sketching input (input method)**

Flexible: We can write/draw anything by sketching it; characters, expressions, diagrams, pictures, and so on. Sketching input doesn't require any input modes, so we can write/draw anything in the order we prefer.

Direct: We can write/draw what we want anywhere we want to in a direct way. We do not need to worry about such things as learning how to use drawing tools, templates of shapes, Chinese character conversion, etc.

iii) **Freehand drawings (expression form)**

Informal: Body type fonts, geometric figures, and regular arrangements tend to give a formal and "hard" impression, while drawings done by hand tend to give an informal and "soft" impression. Furthermore, the former style often gives the impression of being regular, static, stable, complete, elaborate and consistent, while hand-drawn images often give the impression of being disorderly, unsteady, dynamic, unstable, incomplete and ambiguous.

The KJ-method developed by Jiro Kawakita is one of the most famous methods developed to enhance creative activities [3]. handwritten diagrams prepared using the KJ-method give very different impressions when they are compared with diagrams automatically generated by computer. The KJ-method seems to exploit the informality of hand-drawn diagrams to benefit thinking processes [4].

3 A Handwriting Tool to Support Creative Activities

In this section, we give an outline of a handwriting tool designed to support creative activities. The tool is presented using screen shots of a prototype. We also describe the technical challenges of developing such a tool.

3.1 Overview of the handwriting tool for logical drawing

The features of the proposed tool are as follows.

- The main processing objects of the tool are diagrams with logical structures.
- The tool provides a freehand sketching interface.
- The tool maintains the feel and ambiguity of drawings done by hand.
- The tool supports drawing intellectually and actively.

Here, the diagrams with logical structures are not diagrams in which arrangement in coordinates and geometrical shapes are essentially important, but are diagrams where logical relationships such as connective relationships and inclusion relationships are important. Examples include system configuration diagrams, flow charts, concept maps, and network topology diagrams.

The tool provides integration of the handwriting environment and drawing support for diagrams with logical structures. It aids in drawing efforts using handwriting by analyzing the logical structures of diagrams and using these structures for layout constraints or automatic layouts of the diagrams. It not only exploits handwriting as an input method but also preserves the handwritten "feel" of the image and supports redrawing of the images.

3.2 Architecture and Technical challenges

Figure 1 shows the architecture of the tool; it consists of five functional parts. The "gesture recognizer" deals with the user's sketching input as gestures or interprets it as ink data. The "spatial parser" analyzes diagrams consisting of ink data to grasp logical structures of the diagrams. The "constraint solver" solves some constraints according to the logical structure of the diagram. The "layout modifier" computes new layouts of the diagrams, and then the "handwritten style generator" modifies the shapes of the diagrams to preserve the handwritten feel of the original diagrams.

To develop the prototype tool, we used the SATIN toolkit [5] to recognize standard gestures and to make a spatial parser. The spatial parser constructs graph structures of diagrams [6]. Isolated shapes are interpreted as nodes, and shapes connecting to other shapes are interpreted as edges. The other functional parts are written in Java without any toolkit. The constraint solver works to preserve connections between nodes and edges. Moving the nodes influences the positions and shapes of connected edges. The handwritten style generator modifies shapes of the influenced edges.

To further evolve and improve the tool, the rules handled by the five functional parts should be reconsidered. The parsing rules should cover more complex diagrams, and the rules of the constraint solver should be extended to handle various transformations of diagrams. Some of the techniques used in the image-editing program ScanScribe [7] might be used for this purpose.

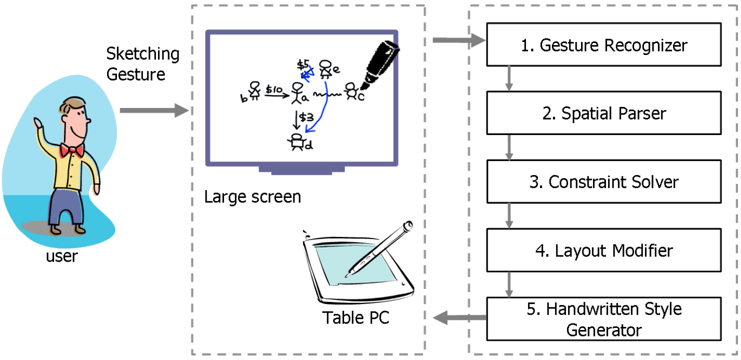
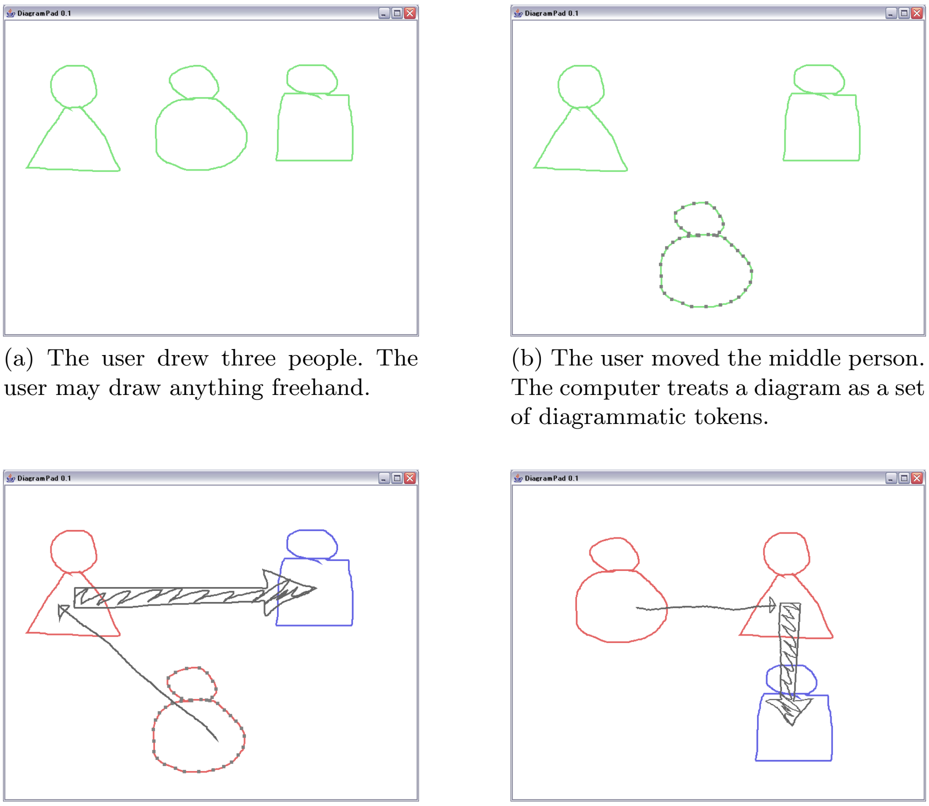


Fig. 1. Architecture of the proposed tool



(a) The user drew three people. The user may draw anything freehand.

(b) The user moved the middle person. The computer treats a diagram as a set of diagrammatic tokens.

(c) Two arrows have been added between the people. The computer understood that the arrows expressed logical relationships between the people.

(d) The user moved the people. The computer changed the positions and shapes of the arrows according to the movement of the people.

Fig. 2. Sample screens of handwriting tool intended for a logical domain

3.3 Behavior of the Prototype Tool

We introduce the behavior of the tool by referring to some screen shots. In Figure 2(a) the user had drawn three people freehand, and in Figure 2(b) the user moved the middle person. The tool treats a diagram as a set of diagrammatic tokens, which are semantic units. In Figure 2(c) the user added two arrows representing relationships between the people. The arrows were also drawn freehand without a template. At that time, the tool understood that the arrows expressed logical relationships between the people. In Figure 2(d) the positions of the people have been moved by the user. The tool automatically changed the positions and the shapes of the arrows according to the movement of the people.

By using some other existing tools, logical relationships can also be expressed between unit shapes. However, an important point here is that the tool extracts logical structures from handwritten diagrams drawn freely and transforms the diagrams while preserving the feel of handwritten diagrams.

4 Related Works

We make the position of the proposed tool clear by introducing some related tools that use handwriting.

Tivoli [8] is an electronic whiteboard application targeted to run on the Xerox LiveBoard. It is designed to support informal workgroup meetings and uses handwriting input with a pen-type device. Electronic Cocktail Napkin [9, 10] and Flatland [11] are also tools aimed to develop informal features of the whiteboard.

Pegasus [12] is a drawing system for rapid geometric design. It “receives” the user’s free hand strokes and beautifies them by considering geometric constraints among segments. Teddy [13] is a 3D modeling system that provides a sketching interface for designing 3D freeform objects.

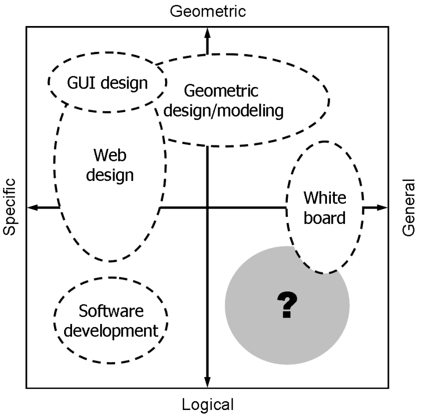


Fig. 3. Target domains of creative activity support

Silk [10, 14] is a tool focusing on the design support of GUI. It combines the advantages of sketching on paper with the advantages of using computer-based tools. DENIM [2] is a system that helps web site designers in the early stages of design.

Knight [15] is a whiteboard system for software development. It is designed to use the best features of the whiteboard and the CASE tool .

Domains of the tools can be characterized along the dimensions shown in Figure 3. The horizontal axis represents the target area of the tool. The vertical axis represents the form of the target information in creative activities. The lower right area marked with "?" is a domain for which handwriting has not been used up to now. The tool we proposed in this article targets this domain.

5 Concluding Remarks

In this article we proposed a handwriting tool to support creative activities. The tool is intended for logical structures, which are useful in the upstream of creative processes. Handwriting, which the tool exploits, benefits human creative activities. We illustrated the position of the tool by introducing some other tools that use handwriting. We also described some technical problems were described in this article. Another issue that needs to be evaluated is how handwriting influences and affects thinking.

References

1. Kunifuji, S. et al.: Trend of research and development of knowledge creation support system and current state of development in JAIST, Report of the 1st symposium on knowledge creation support (2004) 1–10 (in Japanese)
2. Lin, J, Newman, M. W., Hong, J. I., Landay, J. A.: DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design, CHI 2000 (2000) 510–517
3. Kawakita, J.: The KJ-method, Chuokoron-sha (1967) (in Japanese)
4. Misue, K, Sugiyama, K: Freehand-like drawing of compound graphs: A fundamental technique for computer aided abduction, Proc. 37th Conf. Information Processing Society of Japan, 4H-7 (1988) 1304–1305 (in Japanese)
5. Hong, J. I., Landay, J. A.: SATIN: A Toolkit for Informal Ink-based Applications, Proc. UIST 2000 (2000) 63–72
6. Baba, A., Tanaka, J.: Eviss: A Visual System Having a Spatial Parser Generator, APCHI'98 (1998) 158–164
7. Saound, E., Fleet, D., Larner, D., Mahoney, J.: Perceptually-Supported Image Editing of Text and Graphics, UIST '03 (2003) 183–192
8. Pedersen, E. R., McCall, K., Moran, T. P., Halasz, F. G.: Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings, INTERCHI '93 (1993) 391–398
9. Gross, M. D., Do, E. Y.-L.: Ambiguous Intentions: a Paper-like Interface for Creative Design, UIST '96 (1996) 183–192
10. Hearst, M. A., Gross, M. D., Landay, J. A., Stahovich, T. E.: Sketching Intelligent Systems, IEEE Intelligent Systems, Vol. 13, No. 3 (1998) 10–19
11. Mynatt, E. D., Igarashi, T., Edwards, W. K., LaMarca, A.: Flatland: New Dimensions in Office Whiteboards, CHI '99 (1999) 346–353
12. Igarashi, T., Matsuoka, S., Kawachiya, S., Tanaka, H.: Interactive Beautification: A Technique for Rapid Geometric Design, UIST 97 (1997) 105–114
13. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A sketching interface for 3d freeform design, SIGGRAPH 1999 (1999) 409–416
14. Landay, J. A., Myers, B. A.: Sketching Interfaces: Toward More Human Interface Design, IEEE Computer, March (2001) 56–64
15. Damm, C. H., Hansen, K. M., Thomsen, M.: Tool Support for Cooperative Object-Oriented Design: Gesture Based Modeling on an Electronic Whiteboard, CHI 2000 (2000) 518–525