Appeared in the Proceedings of the Working Conference on Advanced Visual Interfaces (AVI2002), ACM Press, Torento, Italy, pp.367-372, May, 2002.

Visual Interaction Design for Tools to Think with: Interactive Systems for Designing Linear Information

Yasuhiro Yamamoto[1] Kumiyo Nakakoji[1,2] Atsushi Aoki[2]
[1]PRESTO, JST and RCAST, University of Tokyo
4-6-1 Komaba, Meguro
Tokyo, 153-8904, Japan
+81-3-5452-5286
{yxy, kumiyo}@ai.rcast.u-tokyo.ac.jp

[2]SRA-KTL Inc.
3-12 Yotsuya, Shinjyuku
Tokyo, 160-0004, Japan
+81-3-3357-9361
aoki@sra.co.jp

ABSTRACT
We have developed a series of tools, which use spatial positioning of objects as a means of externalizations in designing linear information. They include a tool for collage-style writing, a tool for notes summarization, a tool for multimedia data analysis, and a tool for movie editing. With these tools, linear information design is viewed as a concurrent process of framing parts and determining the order of the parts. In designing these tools, visual interaction design has been a center of our project. Our design priority has been put in minimizing the user's cognitive load in creating and modifying parts in a space, and manipulating them in the space. This paper first presents a brief philosophy underneath the system development, and describes interaction techniques used in the tools, such as how a user distinguishes objects positioned in the space, how a user resizes the space by dragging objects toward one of its edges, and how a user sees a trajectory of the objects the user is moving in the space.

Keywords
Visual interaction design, cognitive models, external representations, the ART (Amplifying Representational Talkback) principle, spatial positioning

1. INTRODUCTION
While a number of commercially available tools, shareware, and freeware support users in authoring and designing information, many of such tools focus on the final artifact by offering direct manipulation through GUIs. In early stages of information design, however, designers often need to "play with" externalizations (such as sketches and doodles), which are not necessarily parts of the final form [10].

While CAD tools have been widely used in architectural design to produce final artifacts, in early stages of design architectural designers still use paper and pencil, heavily depending on hand-written sketches and notes to understand what is important and what the real problem is [1]. Such sketches and hand-written notes may not be used as a part of final artifact (solution) but help them understand what the problem is as well as what the final form should be; by allowing them to externalize seemingly "rough" ideas with minimum cognitive overload and to view the externalized representations in a very flexible manner [4].

Early stages of information design are very much thought-intensive. What we need are tools that do not disturb such intensive thinking processes [3]. In order to support such design processes, interactive tools needs to serve as media that a designer can interact with and think with. Visual interaction design needs to play an essential role in designing such systems [9].

This paper presents our interaction-design-based project, which has produced a series of systems that support early stages of designing linear information. These systems use spatial positioning of objects as a means of externalization [8], which is helpful for designers in early stages of linear information design. They include a tool for collage-style writing (ART#001), a tool for notes summarization (ART#002), a tool for multimedia data analysis (ART#003), and a tool for movie editing (ART#004). Each of them consists of three components: ElementEditor (EE), ElementSpace (ES), and DocumentViewer (DV). A user can create and modify parts in EE and place them in ES where the user can resize, move, and merge them. DV simultaneously serializes parts in ES from top to bottom or left to right, whatever the order "natural" to the task.

Throughout this interaction-design based software development project, the design principle called ART (Amplifying Representational Talkback) has been used [10]. Based on Donald Schoen's design theory [6], the ART concept emphasizes on the role and effects of representations play during the designer's thinking processes. The concept embraces the power of paper and pencil in sketches, focusing on perceptual feedback as a complement to cognitive feedback. While producing the above four ART systems, we have identified design requirements that are common across multiple domains, as well as those that are unique in each application domain.

2. 2D POSITIONING FOR DESIGNING LINEAR INFORMATION
Designing linear information, such as paper writing or movie editing, can be viewed as a concurrent process of framing parts and determining the order of the parts. In linear information design, a user needs to construct parts and to frame the whole by determining the order of the parts through a trial-and-error process. The whole and the parts depend on each other and co-evolve forming a hermeneutic circle [7].

For a system that supports designing linear information in this manner, what is most essential is to support the process of co-evolving the whole and the parts. The system should not force the user to directly edit a part in terms of the whole linear information, but rather, should allow the user to have clear identification of each "part," to have an overview of the whole without any scrolling, and to easily understand how those parts depend on each other and are related to the whole.

Based on this consideration, instead of allowing a user to directly manipulating the linear information to work on, we provide a space for objects to be positioned as an *instrument* for interaction [2]; objects as parts framing the whole. A user can freely place objects in a 2D space, and the system automatically serializes the objects from top-to-bottom or left-to-right, whatever the "natural" order in the application domain (Figure 1).

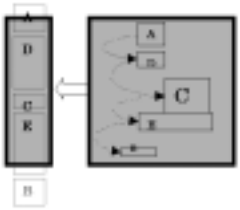

Figure 1. Spatial positioning for linear information design

In our approach, objects or parts that constitute linear information are called elements, and linear information, which is formed by serializing the contents of the elements, is called document. By using the space and spatially positioned elements, the user can take advantage of the perceptual information in reflecting in the current situation [5]. For instance, the user may remember the situation of the current concern by using spatial memory, or identify an element using a spatial arrangement as a visual cue.
This has led to the construction of the architecture (Figure 2) consisting of:

1. ElementEditor (EE) for creating an element to be placed in the space (ES),
2. ElementSpace (ES) for serving as the space, and
3. DocumentViewer (DV) for showing the serialized information as a document.

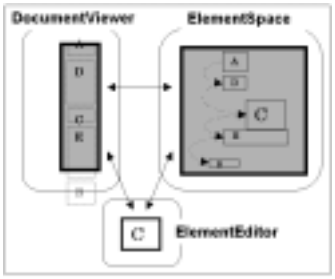

Figure 2. The architecture

We have applied this architecture in different linear information design domains, including writing, notes-summarization, data-analysis, and movie-editing. The basic idea across these multiple linear information design domains is that while linear information design is to produce a 1D representational artifact, a 2D space provided by the system gives a user an additional one dimension which does not affect the final artifact (i.e., serialization) and is not associated with any a priori semantics, and thereby serves for the user as a flexible and powerful representational medium for externalization.

This architecture allows the user in designing linear information in the process of co-evolving the whole and the parts. The user can construct an element and examine a detail of each element in EE, can decide where to put the element in terms of the whole by positioning in ES while looking at other elements, can look at how each element is related to one another and how each element is related to the whole in ES, can change the order of elements by directly manipulating elements in ES, and can read the whole document in DV, checking how transitions between two consecutive elements flow.

3. THE ART SYSTEMS

Four interactive systems have been designed and developed based on this framework. They are all implemented in Visualworks Smalltalk 3 and 5i, and can be downloaded at our Web site[1]. Figure 3 shows the four systems:

They are all for early stages of linear information design consisting of EE, ES, and DV. With each system, what the user does in designing linear information is to create an element in EE, drag and drop it in ES. The system then serializes the contents of all the positioned elements in EE and shows it in DV. Once positioned, the size and location of elements can be changed by direct manipulation. When selecting an element either in EE or in DV by clicking on it, the content appears in EE and the user can modify and update the content. This is reflected in DV when the user "accepts" the edited content. As the user changes the position of the elements, the system automatically updates the serialized document in DV. The content of DV can be saved as a plain text or in an HTML format so that the user may further work on the document with other tools.

For instance, in ART#003, a user can view multimedia data (e.g., a movie recorded a subject's behavior) in EE, and identify an interesting part of the movie by segmenting the movie by specifying a starting point and an ending point. The user can then drag and drop the segmented movie and place it in ES. The user can visually annotate the positioned elements by changing their visual appearance. That is, the user may move and resize the element in the EE; for instance, important factors toward the top and interesting parts but unknown factors toward bigger size. The user can also textually annotate the positioned object. Each object together with its annotation are serialized (from top-to-bottom or left-to-right as the user specified) and shown in DV in a table format. The content of DV can be saved in an HTML format.

The detailed description of ART#001 and ART#002, and detailed user observations using ART#001 using an eye-tracking system can be found in Nakakoji et al. [5]. A brief overview of ART#003 is presented in Yamamoto et al. [11].

4. DESIGN RATIONALE

In designing visual interaction based on this architecture using the spatial positioning of elements for designing linear information, we have identified design requirements and rationale for the requirements that are common across different application domains in linear information design.

[1] http://ccc.aist-nara.ac.jp/systems/ARTware/

Appeared in the Proceedings of the Working Conference on Advanced Visual Interfaces (AVI2002), ACM Press, Torento, Italy, pp.367-372, May, 2002.

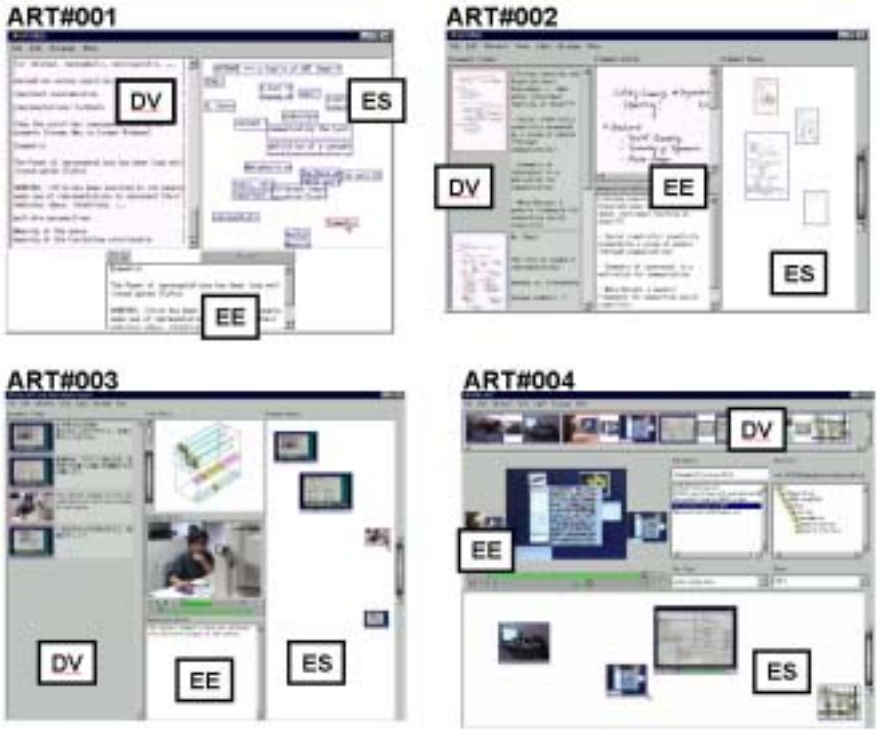

Figure 3. The ART Systems: ART#001 for collage-style writing, ART#002 for notes-summarization, ART#003 for multimedia data analysis, and ART#004 for video editing

Components integration. The integration of the three components is the key in this framework. In looking at details of the whole, DV presents a serialized view of the whole elements positioned in the ES. When moving elements in the space therefore, it must be immediately reflected in the order of the elements in the DV in a real time manner. This integration of the three components is essential across the all of the four systems. The three components must not overlap each other and should be integrated into one tiled-window

Design of the editor. The design of EE depends on the application domain. For ART#001 and #002, EE is for text. For ART#003 and #004, the EE is mostly for images and movies. We have designed a specialized interface for segmenting a movie based on the Quicktime interface (Figure 4).

Figure 4: The scrollbar of EE in ART#004

As shown in the figure, the two buttons on the bottom-left corner are specifically designed for identifying the starting point and ending point of the segment. The bottom green bar indicates which part is segmented in terms of the whole, and the top green bar allows the user to fine-tune the starting/ending points by enlarging the segmented part into a fixed length with 10% of extra time (so that the user may change the starting position to a little earlier position; and vice versa for the ending position). This movie-editing interface is used for ART#004.

Design of the space. The critical task in designing ES in each domain is to allow the user to easily identify each element positioned in the ES. In ART#001, where each element is a chunk of text, we have decided to show 10% of the text content as a default. Asking the user to label the element is creating yet another cognitive overhead; therefore we have intentionally avoided the labeling scheme. In ART#003 and #004, elements are mostly segmented movie parts. We have decided to show a thumb-nailed middle frame of each segmented movie, but it is still difficult to identify, and therefore we provide the Time-Chart window, which is in the top middle of ART#003 (Figure 5). This 3D window shows boxes the faces of which correspond to the thumb-nail images in the ES. The lengths of the boxes correspond to the lengths of each movie. The colored sides indicate which part of the move is segmented from the original movie. The user can change the viewpoint for the 3D space by using the thumbwheel and the hand cursor.

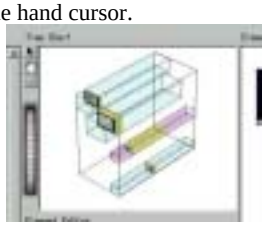

Figure 5. The Time-Chart view in ART#003

Interaction with the space. Because the ES is a place that always needs to give an overview of the whole, it must not have any scrollbar because the use of scrollbar creates invisible space unless the user scrolls up and down. An alternative is the use of

zooming-scale (i.e., enlarging the space), but forcing users to grab a thumb-wheel or to click on a zooming icon to change the zooming scale interrupts the user's thinking process. Considering the situation when the user identifies a need for more space, we have found that the situation emerges when the user moves an element in ES. We have decided to allow users to manipulate the size of ES by dragging the objects toward one of the edges of the ES (Figure 6).

Figure 6. Zooming-by-dragging in ART#002

Interaction with the elements in the space. While observing users with previous versions of ART#001, we have identified that subtle movement of objects in ES plays an important role in a designer's cognitive processes [5]. We have also identified that users often got lost what elements the user was dealing with and where the element was coming from by moving in the space.

To address these issues, we have implemented a trajectory for element movement. In designing trajectory, there are two things to consider: (1) how to represent a trajectory line, and (2) in what timing to erase the displayed trajectory line.

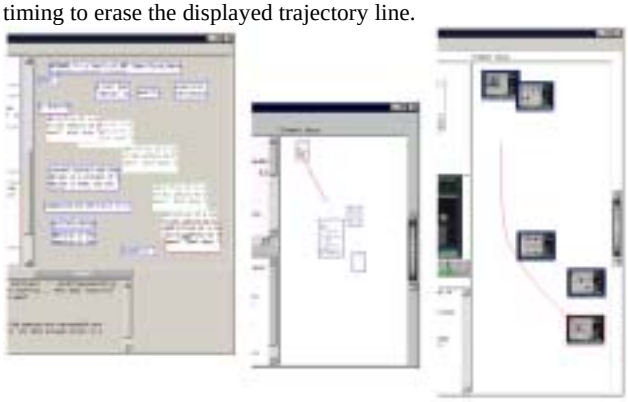

Figure 7. Trajectory in ART#001, #002, and #003 (from left)

So far, we have implemented two combinations. For ART#001, a trajectory is accompanied with multiple images of the moved element (Figure 7). This allows the user to remember which element the user is moving at, and also leaves strong impact of which elements the user has been dealing with. In this implementation, the trajectory line disappears at the time the user releases the mouse button deciding where to put the element.

For the other ART systems, we have implemented a trajectory line as a simple line (Figure 7). The line follows wherever the user is dragging an element. When the user decides where to put the moved element by releasing the mouse, the trajectory line disappears from the starting point toward the ending point incrementally shortening it. This allows the user to reflect on what places the user has been moving the element.

5. CONCLUSION

Through our experience of this interaction-design-based project described in this paper, we argue that interaction-design should be the center of application software development; it should be taken into account from the very first stage of the system development. We particularly emphasize the importance of visual interaction. The first look at the screen shot should tell you everything you can do. If the system requires detailed description of what you can do with it, it may be an indication that the system does not have a good design. We especially worked hard on not adding unnecessary functionality; *what matters is not the number of functions but the "harmony" of functions*. In this sense, user-centered design, which might indicate leaving too much responsibility on users, may not necessarily lead to a good design. In Software Engineering and Human-Computer Interaction research, there has long been an emphasis on the user-developer collaboration. It is time for us to change this perspective and focus more on the role of interaction designer carrying out interaction-design based software development; collaboration among interaction designers, programmers, and end-users.

6. REFERENCES

[1] R. Arnheim, Visual Thinking, University of California Press, CA, 1969.

[2] M. Beaudouin-Lafon, Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces, Proceedings of Human Factos in Computing Systems (CHI2000), ACM Press, pp.446-453, 2000.

[3] A. Cooper, The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity, SAMS Publishing, 1999.

[4] B. Lawson, Design in Mind, Architectural Press, MA, 1994.

[5] K. Nakakoji, Y. Yamamoto, B.N. Reeves, S. Takada, Two-Dimensional Positioning as a Means for Reflection in Design, Design of Interactive Systems (DIS'2000), ACM, New York, NY, pp. 145-154, August, 2000.

[6] D.A. Schoen, The Reflective Practitioner: How Professionals Think in Action, Basic Books, NY, 1983.

[7] A.B. Snograss, R.D. Coyne, Is Designing Hermeneutical?, Architectural Theory Review, Journal of the Department of Architecture, The University of Sydney, Vol.1 No. 1, pp.65-97, 1997.

[8] S. Takada, Y. Yamamoto, K. Nakakoji, Two-Dimensional Positioning as Visual Thinking, Theory and Application of Diagrams (Diagrams2000, Edinburgh, UK), M. Anderson, P. Cheng, V. Haarslev (Eds.), Springer-Verlag, Berlin, pp. 437-452, September, 2000.

[9] B. Weed, Visual Interaction Design: The Industrial Design of the Software Industry, ACM SIGCHI Bulletin Vol.28 No.3, July 1996.

[10] Y. Yamamoto, K. Nakakoji, S. Takada, Hands-on Representations in a Two-Dimensional Space for Early Stages of Design, Knowledge-Based Systems Journal, Elsevier Science, Vol.13, No.6, pp.375-384, November, 2000.

[11] Y. Yamamoto, A. Aoki, K. Nakakoji, Time-ART: A Tool for Segmenting and Annotating Multimedia Data in Early Stages of Exploratory Analysis, CHI'2001, Extended Abstract, pp.113-114, April 2001.

Appeared in the Proceedings of the Working Conference on Advanced Visual Interfaces (AVI2002), ACM Press, Torento, Italy, pp.367-372, May, 2002.