Appeared in the Proceedings of Hypertext2002, ACM Press, College Park, MD, pp.35-44, June, 2002.

Spatial Hypertext for Linear-Information Authoring: Interaction Design and System Development Based on the ART Design Principle

Yasuhiro Yamamoto Kumiyo Nakakoji
PRESTO, JST, and RCAST, University of Tokyo
Knowledge Interaction Design Laboratory, RCAST, University of Tokyo
4-6-1 Komaba, Meguro, Tokyo, 153-8904, Tokyo, Japan
+81-3-5452-5286
yxy@acm.org kumiyo@ai.rcast.u-tokyo.ac.jp

Atsushi Aoki
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 spatial hypertext systems that support early stages of linear-information authoring, such as paper writing and movie editing. They are designed based on the ART (Amplifying Representational Talkback) principle, which emphasizes the importance of visual interaction and the power of external representations. The systems use spatial hypertext not as a medium for representing final artifacts but as a means of interacting with linear information during an authoring process. This paper first describes the role and the effect of the spatial hypertext representation plays in support of early stages of authoring linear information, and explains the ART interaction model for the approach. The ART#001 system, which supports early stages of writing, is described in detail and the other three ART systems are used to illustrate the essential aspects of our approach. The paper concludes with a discussion on the semiotic interpretation of spatial hypertext as a representation, and on the innovative use of spatial hypertext as an instrument to compose information, rather than as an information medium.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical User Interfaces, Interaction styles.*

General Terms

Design, Human Factors

Keywords

Spatial hypertext, interaction design, cognitive models, external representations, the ART (Amplifying Representational Talkback) principle, instrumental interaction

1. INTRODUCTION

We have been constructing a conceptual framework for the support of early stages of linear-information authoring since 1996. We have instantiated the framework by the development of four systems that all use spatial positioning of objects as a means of externalization and interaction. Human-computer interaction design has been playing the central role in the development of the systems.

Positioning of objects in a 2D or 3D space as a way of representation has been studied in the area called spatial hypertext [14]. Based on observations of people naturally using spaces in organizing and structuring objects, spatial hypertext systems allow users to spatially position objects instead of explicitly specifying nodes and links for the purpose of representing information. Spatial positioning is viewed as implying relationships among the objects. Visual cues, such as objects' adjacency, alignment, inclusion, occlusion, size or color are used to interpret emerging relationships.

In most spatial hypertext systems, spatial positioning of objects is used as media with which people incrementally generate, organize and structure information [21]. The space is used to let people put objects in a flexible manner, leaving structure "implicit and informal" [14]. They help users gradually define and fix relationships among objects using emerging structures. Some systems, such as VKB, support this process by providing a mechanism that automatically parses the positioned objects and suggests implied relationships [20].

In contrast to such existing approaches, our systems use spatial positioning of objects as a means to interact with linear information being composed. We view linear-information authoring tasks, such as paper writing, movie editing or music composition, as design tasks, where a user coevolves the parts and the whole going through a hermeneutic circle [22]. We argue that the space that uses spatial hypertext techniques provides an effective external representation to support such tasks not by directly representing information to be authored but by serving as a means to interact with linear information.

With each of our systems, a user first generates a part, and then specifies where to put it in the currently-composed linear information by spatially positioning the part in a 2D space, where positioned parts are serialized in the order from top to bottom (or from left to right, whichever the natural order for the target domain). We have developed the ART#001 system that supports writing based on this approach. We then applied the same framework for other linear-information authoring domains, including notes-summarization tasks (ART#002), multimedia-data

HT'02, June 11-15, 2002, College Park, Maryland, USA.
ACM 1-58113-477-0/02/0006.

analysis tasks (ART#003), and movie-editing tasks (ART#004). Throughout the interaction design of these four systems, the design principle called ART (Amplifying Representational Talkback) has been applied. ART emphasizes that computational media should (1) allow a user to easily represent what he/she wants to externalize, (2) allow a user to easily understand what he/she has externalized, and (3) "be quiet," not offering disturbing services [25].

In this paper we first describe what the fundamental issues are in support of linear-information authoring and why the spatial hypertext technique serves as an effective means to interact with linear information. We then explain a design framework and requirements for our approach, followed by a description of a set of design features of our systems. We describe in detail the ART#001 system for collage style writing and introduce the other three ART systems to illustrate the approach further. We conclude the paper by discussing the semiotic interpretation of spatial hypertext as a representation, and the innovative use of spatial hypertext as an instrument to compose information rather than as media to represent information.

2. SPATIAL HYPERTEXT AND LINEAR INFORMATION

2.1 Supporting Early Stages of Linear-Information Authoring

In our everyday life, many of our intellectual activities need to produce linear information. Regular textual information is linear, scientific papers and reports are still produced in a linear form, and even with the advent of multimedia technologies, hypertext information is read in a linear fashion. We view linear information is a basic form of information and knowledge communicated among people. Linear-information authoring, therefore, constitutes a major part of people's intellectual creative tasks.

Our research has focused on the design and the development of computational environments that support authoring of linear information. In this paper, we use *writing* as a typical linear authoring task, but our discussion should not be viewed as limited to writing; it should be read as applicable to other linear authoring tasks, such as movie editing and music composition.

We have especially looked at early stages of linear-information authoring tasks. Existing computational tools have mainly focused on the support of the production of a final form of information (such as word processing software for pretty formatting), and the early stages have not been well studied. Early stages of any design tasks are cognitively intensive, going through a number of trial-and-error processes starting with a nebulous state of mind. External representations play a critical role in such stages [11]. Many people still prefer paper and pencil during the process not because it provides rich functionality but because it allows them to generate what they want to express in a quite flexible manner [1, 13]. Architects heavily depend on sketches in early stages before they start using CAD systems [13]. How can we support users in "sketching" [10] for authoring linear information?

In architectural sketching, rough shapes and retraced lines give the designer visual cues regarding the state of the mind, for instance, how much he/she is sure about the drawn object [12]. Interacting with such visual, perceptual representations help the designer understand what missing parts are and which directions the design task should proceed in a reflection-in-action cycle [19].

In writing, on the other hand, it is difficult to represent such visual cues with regular text editor. Changing font size and color might help, but this would distract the author from focusing on the writing task. Authoring linear information does not proceed in a linear fashion, and does not always require us to interact with linearly ordered chunks of information.

From these considerations, we have identified two requirements, which are essential for computational tools to support early stages of linear-information authoring (i.e., writing).

First, a user must be able to interact with a chunk of text as an element and linear-information authoring must be supported in a way that the user generates, edits, orders, combines, and divides such elements in a direct manipulation style in order to easily understand how the information flows and to easily change the flow.

Second, the user must be able to see an overview of the unified whole of the linear information being authored while interacting with elements. For textual information (and for any linear information), the meaning of each part is determined by the context of the whole; the meaning of the whole in turn is determined by those of the parts. Thus, the parts and the whole depend on each other forming a hermeneutic circle [22]. The "focus+context" technique developed in information visualization allows a user to look at both the focus (detail) and the context (overview) at the same time [6]. In support of writing, we also need to provide a user with the focus and context simultaneously. Switching modes (for instance, between the outline mode and the printing mode) provided by existing word processing software does not support the hermeneutic circle. How can we represent the whole as a context while a user focusing on a part?

2.2 Using the Spatial Hypertext Representation

We have taken an approach to use the spatial hypertext technique to fulfill the above two requirements. Using the power demonstrated by spatial hypertext systems where spatial positioning of objects is used as media with which people incrementally generate, organize and structure information [21], we have designed a system that allows a user to generate, place, move, and resize objects in a 2D space as parts of the linear information to be composed (Figure 1). One dimension of the 2D space is devoted to serve as a way of serializing objects, and the system always shows linear information consisting of objects as the result of serialization. With this system, the user does not directly interact with the linear information being authored, but composes linear information through interacting with the spatial hypertext representation.

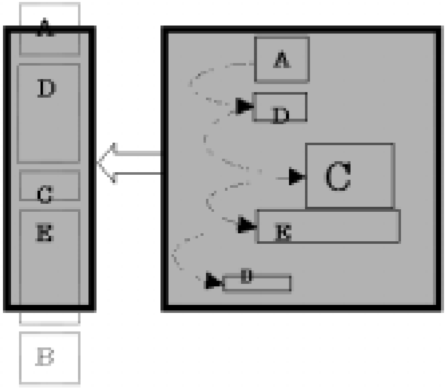


Figure 1: An Approach of Using the Spatial Hypertext Representation for Linear-Information Authoring

In this approach, the 2D space is used not as a totally “free” space to place elements but as a space with a pre-assigned interpretation that associates the vertical relationships among the coordinates of the top-left corners of positioned elements with the order of serialization. All the other geometric and topological features, such as the shape of each element, the size of each element, horizontal relationships among elements, distance between two elements, are left open for the user to represent (or not to represent) any meanings. In other words, the 2D space provides both a means of serialization and a means of flexible externalization like sketching.

At the same time, the 2D space is also used as a way to provide an overview of the whole linear information being authored. As we discussed in 2.1, a challenge in supporting linear information authoring is to provide *focus* and *context* simultaneously. A *focus* can be provided by showing the content of each element when selected. A *context* can be provided by not only showing detailed contents but also by presenting a rough overview of the whole. A visual representation of the positioned elements in the 2D space serves this purpose. Each element positioned in the space may not be readable in detail, but how the elements constitute the whole can be visually apprehensible.

2.3 Design Requirements

Our approach shares the appreciation for the power of spatial positioning as a representation with existing spatial hypertext systems, such as Storyspace [4] and VKB [20]. However, in our approach, spatial positioning of objects is a means to interact with linear information; the user’s primary goal using the spatial hypertext representation is to serialize information rather than to produce hypertext information. This results in different design requirements and interaction designs from most existing spatial hypertext systems.

Design requirements we have identified to support early stages of linear-information authoring by using the spatial hypertext technique are:

(1) a user must be able to first generate an object then to determine where to put the object in the space;

(2) a user must be able to see both linear information being authored and spatially positioned objects constituting the information simultaneously; and

(3) a user must be able to always see all of the spatially positioned objects in the space.

Figure 2 illustrates an interaction model based on the requirements. The model consists of the three components:

- ElementEditor (EE) for generating an element,
- DocumentViewer (DV) for showing linear information, and
- ElementSpace (ES) as a space to position and organize the objects that are vertically (or horizontally) serialized to compose the linear information.

In the remainder of this paper, we call linear information being authored as a *document*, and objects or parts that constitute the document as *elements*.

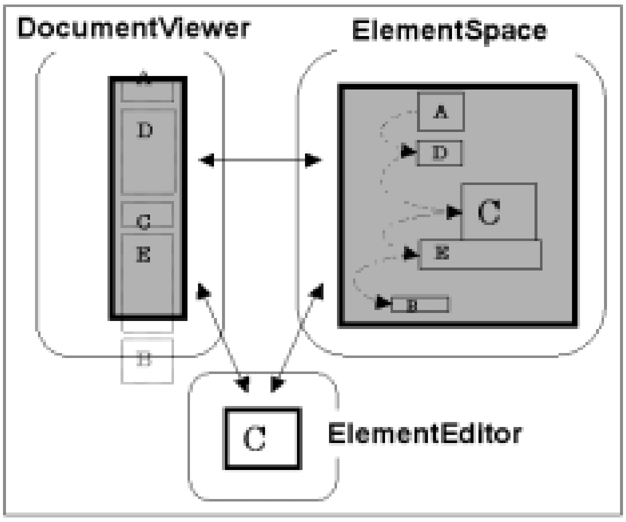


Figure 2: The Interaction Model for the Approach

2.4 The ART Design Principle

In order for a system to serve as an effective, desirable and enjoyable cognitive tool [7], visual interaction design needs to be carefully considered [9, 24]. In conducting interaction design, one needs to decide in a consistent manner what a user would want to do with the system (*tasks*), how the user would do to perform the task (*actions*), and how the system would give visual feedback in response to the actions performed (*reactions*). In order to do so, interaction design should be guided by a design principle throughout the system design and development process [3].

In instantiating the above interaction model (Figure 2) into actual systems through interaction design, we have taken the ART (Amplifying Representational Talkback) principle as the guiding design principle [25]. Based on D. Schoen’s design theory of “reflection in action” and “a talkback of the situation” [19], the RT (Representational Talkback) concept emphasizes the importance of externalization [5, 17, 27] and perceptual feedback from the system [23]. Applying ART, our primary goal of designing the systems has been that computational media should [25]:

(1) allow a user to easily represent what he/she wants to externalize,

(2) allow a user to easily understand what he/she has externalized, and

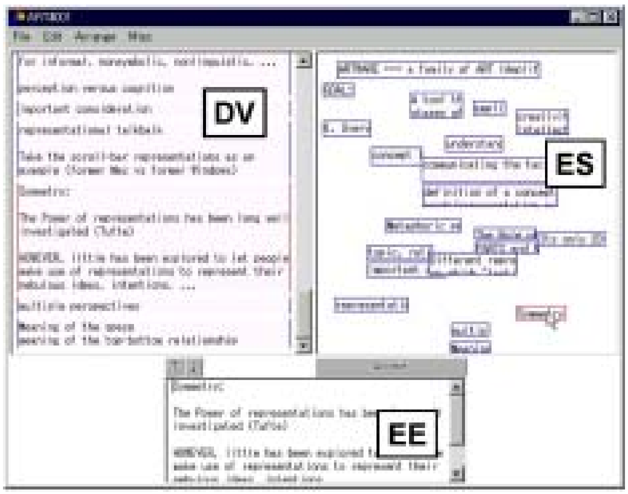
(3) “be quiet” not offering disturbing services.

We have so far applied the model and conducted interaction design based on the ART principle for four different linear-information authoring domains: for writing (ART#001), for notes-summarization (ART#002), for multimedia-data analysis (ART#003), and for movie editing (ART#004). Using ART as a design principle has resulted in a set of design features that characterize the ART systems, such as how to represent objects in the space, how to allow users to manipulate objects in the space, and how to allow users to change the size of the space. The next section takes ART#001 as exemplary and describes in detail how such design decisions features support linear-information authoring using the spatial hypertext representation.

3. ART#001 FOR WRITING

3.1 A System Overview

Before describing detailed interaction design decisions with our systems, we first give an overview of how a user interacts with the ART#001 system. Figure 3 shows a screen image of the ART#001 system.



For informal, nonsymbolic, nonlinguistic, ...

perception versus cognition

important consideration

representational talkback

Take the scroll-bar representations as an example (former Mac vs former Windows)

Geometric:

The Power of representations has been long well investigated (Tufte)

HOWEVER, little has been explored to let people make use of representations to represent their nebulous ideas, intentions, ...

multiple perspectives

Meaning of the space
meaning of the top-bottom relationship

Geometric:

The Power of representations has been investigated (Tufte)

HOWEVER, little has been explored t make use of representations to represent their nebulous ideas, intentions

Figure 3: The ART#001 System for Early Stages of Writing

With ART#001 for text editing, a user generates a chunk of text (words, sentences, or paragraphs) in EE, and then clicks the *Accept* button. Then, the edited text appears in DV and becomes an element in ES. Elements are represented with segmented sidebars in DV and are also represented as thumbnails in ES (a certain amount of a part of the text is shown in a thumbnail by default). The user specifies where to insert the text chunk in the currently composed linear information in DV by dragging the element in ES changing the vertical relationship with the other elements in ES.

ART#001 uses a notion of “selection” of elements. Elements are either selected or not selected, specified by mouse clicking. If a single element is selected, the content of the element is shown in EE and becomes editable. Selected elements are visually distinguished with red sidebars in DV (they are otherwise blue) and red borders in ES (they are also otherwise blue). A selected element shown in EE represents a focused part of the currently composed linear information. DV shows the entire linear document in a scrollable window, which provides a detailed view of the whole. ES shows an overview of the whole document by showing all the elements. Thus, DV and ES together shows a unified whole of the linear information being authored.

The ART principle emphasizes the importance of visual feedback. A user’s particular action in one part of the system needs to be appropriately reflected in other parts of the system as visual feedback, which would then give the user a feeling of an integrated behavior among the three components. Table 1 in Appendix A summarizes what the user does in order to achieve a goal, and how ART#001 reacts to the user’s action in each of the three components.

The most critical aspect of integrating the three components is that the positioning of elements in DV is dynamically manipulated through dragging elements in ES. This needs to hold even during the user dragging an element in ES wondering where to put it. In ART#001, the vertical relationships among elements in ES is reflected in the content of DV (i.e., serialized contents of the elements in ES) in a real-time manner. Selecting an element in ES by clicking makes DV scroll so that the content of the selected element is visible in the DV window. Selecting text in DV also results in the selection of the element in ES. These interactions allow the user to focus on an element both from the linear representation (in DV) and from the spatial representation (in ES).

3.2 Design Decisions and Rationale for Implemented Design Features

This subsection gives a detailed explanation of a major part of the design decisions that have been made for ART#001. We describe a list of design features and their underlying rationale.

3.2.1 EE as an Individual Component

Existing spatial hypertext systems mostly allow a user to edit a chunk of text within a space. This requires the user first to specify where to put an object in a space, then to start editing the object.

In contrast, our decision is that we provide the EE component where the user generates and edits each element and that we do not allow the user to edit an element within ES. This decision has been made because of the two following reasons.

First, we argue that in supporting linear-information authoring, it is important that the user can *first* generate a part, and *then* decide where to put it. Using EE, the user is allowed to start typing text in EE without deciding where this text would be placed in the existing linear information. As soon as the user starts editing the content in EE, the color of the *Accept* button changes darker to remind the user that the change has not been accepted and thereby not reflected in the linear information being authored. When the user feels that he/she has externalized a certain chunk of information, he/she clicks the *Accept* button, then the chunk appears as a text in DV and as a floating thumbnail in ES.

By dragging the element in ES, the user can decide where to put it in the linear document by comparing with existing elements both in DV and in ES, seeing how the text contents flow with the newly edited element. In one of our earlier user studies, we have observed that the user often slowly moved an element around in the space determining where to put it [16]. This real-time update of DV (i.e., linear information) corresponding to the state of ES (i.e., spatial hypertext) enables the user to compose linear information by interacting with the spatial hypertext representation.

Second, our ART principle emphasizes the importance of visual appearance of text elements. How ES looks is very critical to serve as an effective representational talkback to the user. If the user directly edits an element in the space, the visual appearance of ES keeps changing and this affects the perceptual feedback from the space to the user. By having EE as an individual component, the visual appearances of DV and ES do not change till the user clicks the *Accept* button. This allows the user to work on an element in EE while looking at the current visual states of DV and ES. The importance of this aspect of interaction has also been confirmed with eye-tracking user studies reported earlier using a previous version of ART#001 [16].

3.2.2 The Design of DV for Accessing Linear Information

The role of DV is to show serialized contents of the elements placed in ES as the currently-composed linear information. The following two critical design decisions have been made with DV.

First, while DV is to show a serialized document, a user still needs to identify each element; that is, to understand what elements constitute the document. As mentioned above, we have decided to use colored sidebars to represent different elements; different elements are represented with segmented sidebars. Selected elements are represented with red (otherwise blue) sidebars.

Second, we have decided that the same font type and size needs to be used in both DV and EE, and that the width of the DV needs to be the same with the width of EE. This is so because if the text wraps at different points, the same text represented in EE would look differently from that of DV, which would distract the user.

3.2.3 The Visual Appearance of Elements in ES

The design of how we represent elements placed in ES has been the most critical concern in our interaction design. Many existing spatial hypertext systems have been concerned with how to represent "readable" objects in a space. Although a user of our systems also needs to be able to identify each element in ES, readability of each element in ES is *not* a top priority because our systems are equipped with EE and DV, which allow the user to examine the content of each element when needed. Instead, we have given priority to the preservation of the visual layout of elements in ES: the size, shape, and topology of elements. We have thus decided that once created, the size of each thumbnail element in ES would remain the same even the user edits the content of the element (unless the user explicitly changes the size and shape of the element). In addition, it is essential that the system would not require the user to do any additional work such as labeling or numbering to create an element other than writing the content itself because it would then disturb the intensive cognitive process of linear-information authoring.

The above discussion has led to a decision where the system creates an element (as a thumbnail) by showing a certain amount of the area from the top-left corner of EE when the user clicks the *Accept* button; the longer the text is, the larger the corresponding thumbnail becomes. This gives enough visual identity for each element to be distinguished from the others. We have observed in our user study that this decision had some effects on a user's behavior, such as putting textual description in the first line of an element to serve as a label for the element. No subjects had any complaints about this design decision [26].

3.2.4 The Design of ES for Viewing a Unified Whole

As we argued above, one of the two purposes ES serves is to specify and change the order of serialization of elements in DV. With the space, only the vertical relationships among the coordinates of the top-left corners of elements are interpreted by the system, and are used to serialize the elements into a linear document. When the user moves an element in ES causing the change in the vertical relationships resulting in the change of the order of elements in the document.

To visually stress the impact of such movement, we have implemented a mechanism to visualize how the user has been dragging an element by showing its initial position and the trajectory of how the element has been moved in the space. Translucent images of elements are displayed along the trajectory with lighter-red borders. Figure 4 shows how trajectory is represented in ES. When the user releases the mouse button and specifies where to put the element, the trajectory disappears.

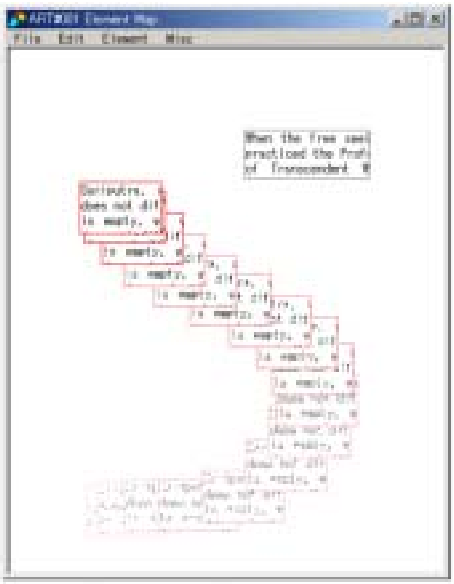


Figure 4: Trajectory of an Element in ART#001

While vertical relationships are used for serialization in ES, it is important that any other interactions with the elements in ES such as resizing or reshaping should not have any impact on the information being authored. In order to achieve this requirement, we have decided that resizing and reshaping of an element is possible only by grabbing its bottom-right corner, which does not cause any change in the vertical order.

Since the system compares only the top-left coordinate of each element, the user may place elements overlapping each other. Our user studies have identified many situations when subjects produced elements overlapping each other [26]. Such spatially overlaid elements may result in some objects completely occluded [18]. To deal with the problem of such occlusion, we have implemented a mechanism that allows the user to toggle the position of an element between front and back; clicking makes an element move to the front-most and double-clicking makes the element move to the back-most.

3.2.5 The Manipulation of the Space

The other primary purpose of ES, which is to provide an overview of the whole document being authored, requires that all the elements are always visible in the space. The use of scrollbars is therefore not appropriate for ES because it would create areas that are invisible. However, there emerges a situation when the user needs to have more room to place elements. To deal with this situation, we have implemented the zooming-out-by-dragging mechanism.

Figure 5 illustrates how the mechanism works. When the user drags an element and releases the mouse button outside of the current view of ES where the user wants to place the element in terms of the positions of the other elements, the system zooms-out in ES so that the dragged element fits within the view of ES. Conversely, when the user wants to maximize the visual appearance of elements in ES, the "fit-zoom" command evoked by the menu would zoom-in in ES so that all the existing elements are shown larger in ES.

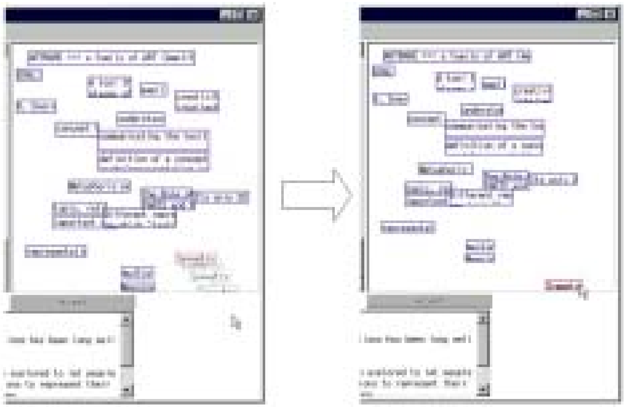


Figure 5: Zooming-out-by-Dragging

In implementing zooming-out and zooming-in, we had to decide how we change the size and relative positions of existing objects in ES. In following the ART principle, it has been decided to minimize the change in relative relationships among the sizes of elements and among the positions of the elements. ART#001, therefore, when zooming-out or zooming-in, maintains the aspect ratio for each element, as well as the ratio between the horizontal and vertical distances among elements. The top-left corner of ES is used as a fiducial point.

Finally, to support the user to go through all the elements positioned in ES in a serialized order, the two arrows located on top of EE allow the user to parse each element by selecting one by one, one for going toward the beginning and the other is going toward the end, serving as support for proof-reading the linear information being authored.

3.2.6 Features Intentionally Avoided

In order to achieve consistency in its interaction and to avoid unnecessary complexity, we have intentionally not provided some functionality in ART#001. Such functionality includes coloring of elements, explicit linking representations, and grouping of elements.

The beauty of spatial hypertext resides in its continuous representation. The size and position can be an analog representation, which allows people to represent structures implicit and informal [14]. Currently, we do not have any mechanisms that allow people to do coloring in an analog manner through direct manipulation, and it is not included in the current implementation of the system.

Explicit linking also affects the power of spatial positioning as a representation. Links either exist or not, forcing the user to make a commitment for a discrete decision.

Grouping and substructuring mechanisms did exist in the previous version of ART#001. It had a layering mechanism, with which a user could group elements in ES and place them on different layers [26]. The layers were translucent, and the user could see through overlapping layers in ES. The user could manipulate each group of elements through the use of the layer, such as toggling a visible/invisible mode. The layers were provided in order to support grouping elements. However, in the user studies using the system, we have found that users got often confused in understanding how layers were related to each other and which element belongs to which layer. We have identified a strong need for structuring layers. In redesign of ART#001, therefore, we have decided that ART#001 only deals with elements that belong to a single group. At the same time, we have started to develop a system that is for structuring and designing hierarchies. Our plan is that when finished, our ART#001 system will be integrated with the hierarchy-designing system: ART#001 for creating a group of elements, and the hierarchy-designing system for structuring multiple groups. A user should be able to going back and forth between the two systems.

4. THE OTHER ART SYSTEMS

The previous section described in detail how ART#001 supports writing as a linear-information authoring task, and discussed design rationale for interaction designs.

As we discussed above, we have implemented three other systems for linear-information authoring that are all based on the same interaction model as illustrated in Figure 2: ART#002 for notes summarization, ART#003 for multimedia-data analysis, and ART#004 for movie editing.

With each of the three systems, what the user does in authoring linear information is to create an element in EE, and to drag and drop it in ES to determine where to put the new element in the currently composed linear information presented in DV. Once positioned, the size and location of elements can be changed in ES. The size of the space of ES (zooming scale) can also be changed by dragging elements in ES toward one of the edges. When selecting an element either in DV or in ES 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, as a QuickTime movie, or in HTML format so that the user may further work on the document with other tools.

In ART#002 for notes summarization (Figure 6), the middle window shows a scanned image of a user's hand-written note. The user drags the note image to ES and positions the thumbnail of the note image produced by the system. The user can then produce text by looking at the note image; the text content is associated with the note image. The user may position multiple images of notes in the ES. All the positioned elements in ES, each of which is associated with text content, are serialized from top-to-bottom and shown in DV in a tabular format. The content of DV can be saved in an HTML file.

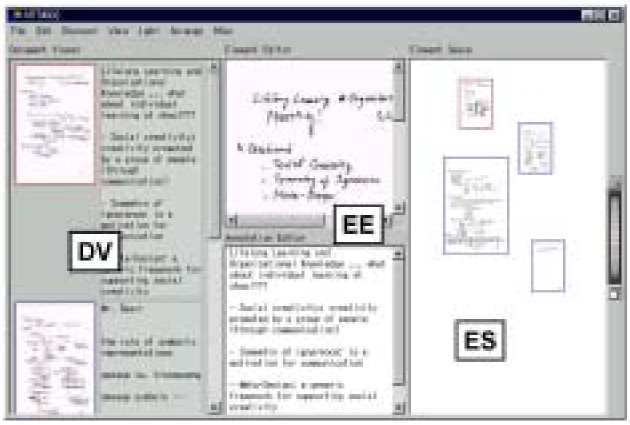


Figure 6: ART#002 for Notes Summarization

In ART#003 for multimedia-data analysis (Figure 7), the user's interaction is the same with ART#002 except that a user views a movie instead of a scanned jpeg image. With ART#003, the user can view a multimedia-data file in EE (which is developed on top of QuickTime), and identify an interesting part of the movie by segmenting (clipping) 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 the ES (a dragged movie segment is shown as a thumbnail image of the focused frame of the segmented movie in ES). The user may textually annotate the positioned element. Each element together with its text 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 also be saved in an HTML file. In order to support the user to identify which element positioned in ES corresponds to what part of which movie, the *TimeChart* window (the top middle of ART#003) is provided to show 3D boxes, the faces of which correspond to the thumbnail images in the ES with the depth representing the lengths of the original movies. The colored sides indicate which part of each movie is segmented from the original movie. The user can change the viewpoint for the 3D space by using the thumbwheel and the hand cursor located on the left side of the *TimeChart* window.

In ART#004 for movie editing (Figure 8), the user interacts with the system in the same manner as with ART#003, but does not textually annotate each element. Instead, the system appends segmented movie elements positioned in ES (which is now located at the bottom of the window) from left to right and allows the user to view the appended movie, helping the user identify necessary parts of movie materials. Triplet-thumbnail images, each consisting of the initial, current, and the end frames of each segmented movie part, are displayed in the DV.

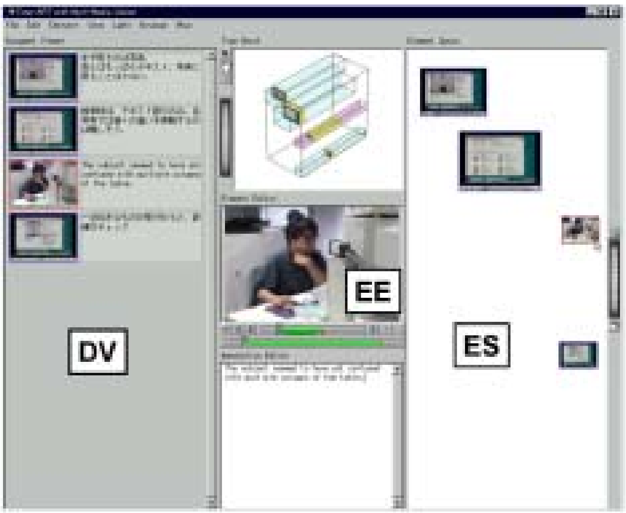


Figure 7: ART#003 for Multimedia-Data Analysis

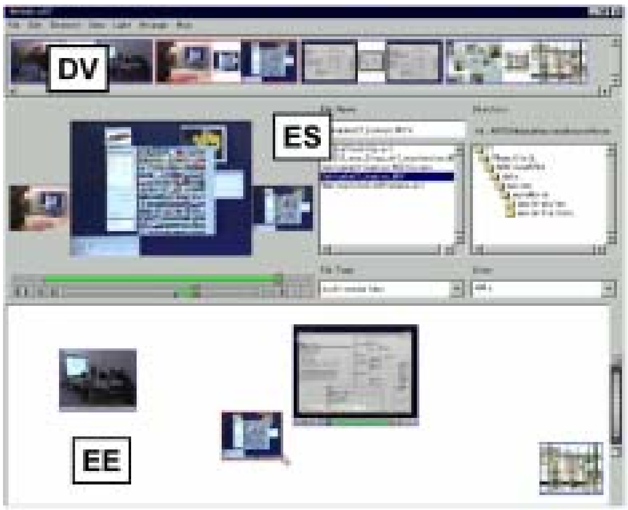


Figure 8: ART#004 for Movie Editing

5. DISCUSSIONS

In this paper, we have described our approach of using spatial hypertext as a means for interacting with early stages of linear-information authoring. This section discusses two views of the spatial hypertext representation that our systems have explored.

5.1 Spatial Hypertext as an Indexical Representation

The hallmark of the spatial hypertext systems has been the flexibility, ambiguity and evolvability of the spatial positioning of objects as a representation [4] [14] [18] [21]. Our approach has taken an advantage of this and used the spatial hypertext representation as a powerful external representation. ART systems use spatial hypertext as a temporary representation to interact-with and think-with, and as a representation to be easily externalized and understood. Unlike many spatial hypertext systems, however, ART systems do not use spatial positioning of objects as a final form of the product. They are designed based on an assumption that patterns of spatial arrangement of objects would not eventually be associated with any meanings other than serialization.

In our user studies, we have found that the spatial positioning of elements allowed users to externalize a variety of *meta-comments* for linear information authoring by using size, a distance between objects, or the way of alignment [26]. Such meta-comments include what elements are missing; how much the user is "sure" about a newly created element; what the role of this element is in terms of the whole document; what the role of this element is in terms of other elements; or which direction the whole linear information design is moving toward and whether the direction is in accordance with the intention behind the authoring task.

In the studies, however, we have not found any definitive relationships between patterns of spatial positioning and types of meta-comments. Rather, people use two-dimensional positioning as *reminders*. While interacting with the 2D space, users do not reflect on the spatial positioning per se; they do reflect on the state of mind that is remembered by looking at a particular aspect of the 2D space.

For instance, in one of the ART#001 user studies, we have found that one subject used the left area and right area of ES differently [16]. She used the left area for positioning elements that "*are important and directly contribute to the final document*", and the right area for positioning elements that "*would either require more attention or were related but not directly useful for my current document.*" This behavior was unexpected and found a little confusing because contents of the elements in ES were all appended in the order of top to bottom and displayed in DV regardless of whether they are on the left side or on the right side. She carefully positioned elements in the left part of EM in terms of the vertical relationship. At the same time, she could completely ignore the vertical relationship of elements on the right side. In the very end of her writing task, she adjusted the position of the large element on the right side so that the content of this element was incorporated in the very end in DV.

This indicates that the subject was very good at looking at positioning in a context that she mentally projected; in this case the left region and the right region. Although there was no clear boundary displayed in EM, subtle spatial information was enough for her to indicate meaning without confusion. A visual representation reminds the user of a particular situation and thought processes that the user was engaged in when created the representation. Two-dimensional positioning served as *indices* for thoughts [15].

The classification of signs that Peirce provides in semiotics, *symbolic*, *iconic*, and *indexical*, has been useful for us to argue for the aspect of two-dimensional positioning serving as indices for thoughts. We view two-dimensional spatial positioning the ART provides is a type of Indexical interface. While describing current user interface approaches using the Peirce classification of symbols, it has been argued that keyboard and command languages were concerned with *symbolic* communications and WIMPs with *iconic* ones, and that *indexical* interface needs to be explored in the coming age of HCI [8]. We argue that the way ART systems use the spatial hypertext representation is one type of such *indexical* interfaces.

5.2 Spatial Hypertext as an Instrument for Interaction

Our approach uses the spatial hypertext representation as a means to interact with linear information. Instead of directly composing linear information, the user interacts with spatially positioned objects to generate a new part and to change the order of the part in the linear information being authored.

Beaudouin-Lafon introduces the notion called instrumental interaction [2]. He uses a scrollbar as an example of an instrument for interaction to view the information content larger than a visible window area. Our use of the space has a very similar role. In one sense, ES in ART#001 can be viewed as an "editable" scrollbar for DV. Clicking on an element in ES will scroll the content in DV. The order of serialization can be changed using the elements placed in this scrollbar rather than directly cut and paste text chunks in DV. What spatial hypertext (ES) provides in ART#001 is this role of an instrument to interact with linear information (DV) being authored. In ART systems, the use of spatial hypertext is viewed as an instrument rather than as a medium to represent information.

Through our experience of implementing four systems that use spatial positioning of objects as an interaction method, we have found that what is really essential about spatial positioning is the feeling of "hands-on-ness." With the graphical interface, clicking, grabbing, dragging, and placing an object on a screen has been found a quite natural interaction method. As discussed in 5.2, the spatial hypertext representation has been found a powerful external representation. The use of spatial hypertext as an instrument for interaction is a promising area to pursue for the next generation hypertext research.

6. CONCLUSION

We have presented our approach to use the spatial hypertext representation as a means to interact with linear information being authored. The 2D space is used both to serialize elements (parts) and to provide an overview of the whole (i.e., linear information to be authored). We have developed the interaction model as a framework for the approach, and ART (Amplifying Representational Talkback) design principle has been used to guide the interaction design process for the systems. The use of the spatial hypertext representation as an instrument for interaction rather than a medium to represent information has been found a powerful approach to support cognitively intensive design tasks such as writing.

7. ACKNOWLEDGMENTS

We would like to give special thanks to Mark D. Gross for helping us identify theoretical foundation of the approach, Shingo Takada for the design of early versions of the ART systems, and Yunwen Ye and anonymous reviewers for valuable feedback for early versions of this paper.

8. REFERENCES

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

[2] Beaudouin-Lafon, M. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces

Proceedings of Human Factos in Computing Systems (CHI2000), ACM Press, New York, NY., 2000, 446-453.

[3] Beaudouin-Lafon, M. and Mackay, W. Reification, Polymorphism and Reuse: Three Pcinciples for Designing Visual Interfaces Proceedings of Advanced Visual Interfaces, AVI2000, ACM Press, Palermo, Italy, 2000, 102-109.

[4] Bernstein, M., Bolter, J.D., Joyce, M. and Mylonas, E. Architectures for Volatile Hypertext Proceedings of Hypertext'91, ACM, New York, NY, 1991, 243-260.

[5] Bruner, J. The Culture of Education. Harvard University Press, Cambridge, MA, 1996.

[6] Card, S.K., Mackinlay, J.D. and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. Morgan-Kaufmann Publishers, San Meteo, CA, 1999.

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

[8] de-Souza(Ed.), C.S. Special Issue on Semiotic Approaches to User Interface Design. in Knowledge-Based Systems Journal, Elsevier Science, 2001.

[9] Djajadiningrat, J.P., Overbeeke, C.J. and Wensveen, S.A.G. Augmenting Fun and Beauty: A Pamphlet Proceedings of DARE 2000: Designing Augmented Reality Environments, Helsingor, Denmark, 2000, 131-134.

[10] Do, E.Y.-L. and Gross, M.D. Inferring Design Intentions from Sketches: An Investigation of Freehand Drawing Conventions in Design Proc. of the 2nd Conf. on Computer Aided Architectural Design Research in Asia (CAADRIA'97), Hu's Publishing, Taipei, Taiwan, 1997, 217-227.

[11] Goldschmidt, G. On Visual Thinking. in Design Studies, 1994, 158-174.

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

[13] Lawson, B. Designing with Drawings. in How Designers Think: The Design Process Demystified, Architectural Press, MA, 1997, 241-259.

[14] Marshall, C.C. and Shipman, F.M. Spatial Hypertext: Designing for Change. in Communications of the ACM, 1995, 88-97.

[15] Nakakoji, K. and Yamamoto, Y. What does the Representation Talk Back to You? in Knowledge-Based Systems Journal, Elsevier Science,, 2001, 449-453.

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

[17] Reisberg, D. External Representations and the Advantages of Externalizing One's Thoughts Proceedings of the 9th Annual Conf. of the Cognitive Science Society, 1987.

[18] Rosenberg, J. User Interface Behaviors for Spatially Overlaid Implicit Structures. in Workshop on Spatial Hypertext (Hypertext 2001), Arhus, Denmark, 2001.

[19] Schoen, D.A. The Reflective Practitioner: How Professionals Think in Action. Basic Books, New York, 1983.

[20] Shipman, F., Hsieh, H., Airhart, R., Maloor, P. and Moore, J.M. The Visual Knowledge Builder: A Second Generation Spatial Hypertext Proceedings of Hypertext2001, ACM, 2001.

[21] Shipman, F., Marshall, C. and LeMere, M. Beyond Location: Hypertext Workspaces and Non-Linear Views Proceedings of Hypertext'99, ACM, 1999, 121-130.

[22] Snograss, A.B. and Coyne, R.D. Is Designing Hermeneutical? in Architectural Theory Review, Journal of the Department of Architecture, The University of Sydney, 1997, 65-97.

[23] Takada, S., Yamamoto, Y. and Nakakoji, K. Two-Dimensional Positioning as Visual Thinking. Edinburgh, U. ed. Theory and Application of Diagrams (Diagrams2000), Springer-Verlag, Berlin, Germany, 2000, 437-452.

[24] Weed, B. Visual Interaction Design: The Industrial Design of the Software Industry. in ACM SIGCHI Bulletin, 1996.

[25] Yamamoto, Y., Nakakoji, K. and Takada, S. Hands-on Representations in a Two-Dimensional Space for Early Stages of Design. in Knowledge-Based Systems Journal, Elsevier Science, 2000, 375-384.

[26] Yamamoto, Y., Takada, S. and Nakakoji, K. Representational Talkback: An Approach to Support Writing as Design Proceedings of 3rd Asia Pacific Computer Human Interaction Conference, IEEE Computer Society, Kanagawa, Japan, 1998, 125-131.

[27] Zhang, J. The Nature of External Representations in Problem Solving. in Cognitive Science, 1997, 179-217.

APPENDIX A.

Table 1: Integration of the Three Components in ART#001

The user's goal	The user's action	The system's reaction
To change the order of a part	ES: The user drags an *element* and releases the mouse button where the user wants to put it	ES: The initial position and its trajectory are displayed till the user releases the mouse button and specifies the position of the *element*. The *element* remains as selected by having its border red. DV: The DV window shows the *element* being dragged in the document at the vertically serialized position in ES. DV keeps scrolling to make the *element* visible in DV. The element remains as selected by having its sidebar red.
To write new stuff	EE: The user starts generating text without selecting any *elements* and clicks the	ES: The new *element* is created and appears in the center of ES, floating till the user clicks to specify where to put the *element*. DV: DV inserts the new *element* in the document at the vertical order

	Accept button	of the *element* in ES. DV keeps scrolling to make the *element* visible in DV. The element remains as selected by having its sidebar red.
To examine the content of a part by selecting it	DV: The user clicks on text in DV	EE: The content of the selected *element* appears in EE and becomes ready to be edited. ES: The corresponding *element* is shown as selected in ES by changing its border color from blue to red.
	ES: The user clicks on an *element*	EE: The content of the selected *element* appears in EE and becomes ready to be edited. DV: The DV window scrolls so that the *element* is visible in DV. The *element* is shown as selected by changing its sidebar from blue to red.
To reedit the content of a selected part	EE: The user edits the selected *element* and clicks the *Accept* button	ES: The content of the selected *element* is updated, remains as selected by having its border red. DV: The content of the selected *element* is updated, remains as selected by having its sidebar red.
To divide a single part into two parts	EE: The user selects an *element*, selects a certain	ES: The spun off portion of the text appears in the ES as a new *element*, which starts floating from the position where the original *element*
To merge multiple parts	ES: The user selects multiple *elements* clicking on them and evokes the “join” command from the pop-up menu	EE: The content of the appended contents of all the selected *element* appears in EE and becomes ready to be edited. ES: The selected *elements* are animated to be merged into the first (top-most) one among the selected *elements* and shown as selected by having its border red. DV: The DV window scrolls so that the content of the merged *element* is visible. The element is shown as selected by having its sidebar red.
To remove unnecessary parts	ES or DV: The user selects one or more *elements* in ES or in EV and evokes the “delete” command from the pop-up menu	ES and DV: The selected *element(s)* is(are) removed both from ES and DV. EE: Nothing to be shown.