Available online at www.sciencedirect.com

SCIENCE DIRECT®

Int. J. Human-Computer Studies 63 (2005) 513–535

International Journal of Human-Computer Studies

www.elsevier.com/locate/ijhcs

Interaction design of tools for fostering creativity in the early stages of information design

Yasuhiro Yamamoto*, Kumiyo Nakakoji

Research Center for Advanced Science and Technology, University of Tokyo, 4-6-1 Komaba, Meguro, Tokyo 153-8904, Japan

Available online 23 June 2005

Abstract

This paper describes our approach for the design and development of application systems for early stages of information design tasks. We view a computational tool as something that provides materials with which a designer interacts to create a situation that talks back to the designer. The interaction design of a tool, that is, the representations a user can generate and how the user can manipulate them with the tool, influences a user's cognitive processes. The tool's interaction design thus either fosters or hinders creativity in the early stages of information design.

Our approach toward the interaction design of a tool for fostering creativity is first to understand the nature of early stages of information design tasks. We discuss four issues in support of the early stages of design based on theories in design and in human–computer interaction: (1) that available means of externalizations influence designers in deciding which courses of actions to take; (2) that designers generate and interact with not only a partial representation of the final artefact but also various external representations; (3) that designers produce externalizations to express a solution as well as to interpret the situations; and (4) that a design task proceeds as a hermeneutic circle—that is, designers proceed with projected meanings of representations and gradually revise and confirm those meanings.

The above theoretical account of early stages of information design tasks has led us to identify three interaction design principles for tools for the early stages of information design: interpretation-rich representations, representations with constant grounding and interaction methods for hands-on generation and manipulation of the representations.

*Corresponding author. Tel./fax: +81 3 5452 5286.

E-mail addresses: yxy@computer.org (Y. Yamamoto), kumiyo@kid.rcast.u-tokyo.ac.jp (K. Nakakoji).

1071-5819/$ - see front matter
doi:10.1016/j.ijhcs.2005.04.023

To illustrate our point, we take ART#001, a tool for the early stages of writing, to apply the interaction design principles and examine how the interaction design of the tool fosters creativity in the early stages of information design. The paper concludes with a discussion of how we generalize the approach and build a framework to design and develop application systems for fostering creativity in the early stages of information design.

Keywords: Information design; Creativity; Interaction design; Cognitive tools; Representations

1. Introduction

The role of information technologies has expanded to serve as tools for the creation of art and knowledge (Shneiderman, 2000). In supporting information creation, application systems have mainly been used to digitally express information artefacts. Examples are word-processing software, image-processing software, or spreadsheet applications.

Few systems are designed for users to think of what to express as information artefacts—what we call the early stages of information design. Early stages of information design tasks—for instance, writing academic papers, producing analysis reports, designing Web sites, or editing movies—involve human creativity and cannot be automated. Supporting psychological creativity is not a luxury but a necessity in this context (Candy and Hori, 2003).

In the early stages of information design, the designer is primarily engaged in conceptual design (Edmonds and Candy, 1993). In conceptual design, a designer starts with a vaguely understood design requirement and gradually develops an understanding of what the problem is and what the solution should be.

We take a pragmatic account of design by viewing information design as "being engaged directly in a specific design situation" (Fallman, 2003). We view a computational tool as something that provides materials with which a designer interacts to create a situation that "talks back to the designer" (Schoen, 1983). In order to foster creativity in the early stages of information design, our research has focused on the interaction design of such tools by considering what representations a designer needs to generate and how to manipulate them using the tools.

Different people use the phrase *interaction design* in slightly different manners. It has been defined quite generally as "the design of interactive systems" (Preece et al., 2002) or rather specifically as "the selection of behavior, functions, information and their presentation to the user" (Cooper, 1999). In this paper, we use the phrase *interaction design* to mean "determining the representations and operations of an application system by considering what representations the user needs to interact with, through what operations."

The interaction design of a tool influences a designer's cognitive process. Tools for fostering, not obstructing, creativity need to be designed around the understanding of what representations a user needs to interact with.

The research issues we address in this paper involve understanding what representations designers need to interact with in fostering their creative process in the early stages of their design task, and determining how we can conduct the interaction design based on this understanding.

To address these issues, our approach is first to develop a theoretical account for what designers do with external representations. We then derive interaction design principles from the theoretical account and last, we examine existing tools in terms of these principles and develop a general framework for the approach.

Section 2 looks at theories in design and in human–computer interaction to understand the nature of early stages of information design tasks. Four issues are elaborated as a theoretical account in supporting designers with computational tools as materials to interact with. Section 3 identifies three interaction design principles for a tool to support such design tasks based on the theoretical account: interpretation-rich representations, representations with constant grounding and interaction methods for hands-on generation and manipulation of the representations. Section 4 examines an existing tool, ART#001, in terms of the principles. ART#001 is a tool for early stages of writing using spatial hypertext as an interaction method. The paper concludes with a discussion of how we generalize the approach and build a framework to design and develop application systems for fostering creativity in early stages of information design.

2. Nature of the early stages of information design

Tasks in the early stages of information design have a "wicked", ill-defined, open-ended, argumentative nature (Schoen, 1983; Rittel and Webber, 1984; Winograd and Flores, 1986; Simon, 1996). In wicked problems, a designer does not have a clear understanding of what to produce and has only a vague goal in mind in the beginning of the task. There are no clear rules or schemes to judge whether the design task is finished; the task is finished only when the designer knows that it is finished. In addition, no clear rules or schemes exist to judge whether intermediate design steps being made are progressing or retreating.

We have identified four issues for computational tools to support the early stages of information design in terms of the nature of such design tasks. They are derived from the existing design studies and cognitive studies as well as human–computer interaction theories and design theories.

2.1. (I-1) *Available means of externalizations influence designers in deciding which courses of action to take*

Existing studies have demonstrated the importance of external representations in ill-defined problem-solving tasks. Different externalizations result in variations in problem-solving performance because such representations change the nature of the problem (Zhang, 1997). People use visual features and physical constraints in solving problems, such as in comparing objects and remembering situations. Represented differently, mathematically equivalent problems become dissimilar problems for people because solving them requires various cognitive processes and resources.

Following de la Rocha's cottage-cheese study (de la Rocha, 1985), Shirouzu and colleagues asked college students to indicate two-thirds of three-fourths of the area of a square sheet of paper (Shirouzu et al., 2002). They found that a large number of subjects immediately folded the paper without first calculating the area (which could have resulted in a much simpler task—to indicate a half of the area), and that they strongly depended on externalized traces such as creases on the paper.

An application system is a cognitive tool (Norman, 1993) in which the tool defines the possible space of what a user can externalize and how the user can manipulate the externalized objects through the representations and operations provided by the tool; that is, through the interaction design of the tool. The above findings indicate that the interaction design influences people's problem-solving processes because it changes the nature of the problem. A certain interaction design may lead some people into taking a certain problem-solving strategy even if other, more efficient strategies are available.

In talking about the cognitive notations for programming, Green argues that all mental operations are often implicitly assumed to be equivalent in cost, but that it is certainly not true. "Certain notations require users to perform operations that are notoriously difficult" (Green, 2000). Tools fostering creativity should not give users such high cost through their poor interaction designs.

2.2. (I-2) Designers generate and interact with not only a partial representation of the final artefact but also various external representations

Architects prefer sketching rather than detailed CAD representations when thinking of design (Lawson, 1994). Chemists use plastic molecular models, not computer-generated 3D graphical representations, in making molecular reaction predictions (Nakakoji et al., 2003). Both are easy-to-handle, low-fidelity representations. What are the roles of such representations?

In wicked design tasks, what the problem is and what the solution should be are dependent on each other; understanding the problem and the solution co-evolve (Fischer and Nakakoji, 1994). Based on a partial understanding of the problem, a designer can decide what to produce as a partial solution. This partial solution then helps the designer further understand what the problem is. To pursue the design task in such a manner, a designer uses representations both for the problem and for the solution because externalization allows a person to take an outside view in reflecting on his/her mental images, lessening the cognitive load of remembering mental images (Bruner, 1996). The designer can then reflect on a partially constructed solution as well as a partially framed problem (Nakakoji and Fischer, 1995).

Many existing application systems for information design have as the primary goal for the user to express and fabricate a final form of the information artefact to be designed. The final form of the information artefact is a representation of a solution. Such application systems often do not provide representations for anything else, such as representing problems or annotating the current situation.

For example, a typical What-You-See-Is-What-You-Get (WYSIWYG) word-processing tool allows a user to always look at how a document being designed would appear in final form (such as in a two-column, line-filled form with inserted figures). With such a tool, no means are provided to represent anything other than a solution, such as what is missing; users may temporarily insert blank lines to remind them something needs to be added. This, in turn, would cause undesirable modifications of a solution representation.

Representations for problems are necessary to consider design problems. The need to represent thinking has been recognized in the domain of programming. Paul Graham points out that a programming language is "for thinking of programs, not for expressing programs you've already thought of" (Graham, 2003). By introducing the notion of cognitive dimensions, Green (2000) argues that a notation, a representation for designing, should be designed from the point of view of programmers thinking of programs. His cognitive dimensions provide a set of principles in designing a notation. For instance, considering generative ordering, the generative order is decoupled from the order in the final source code, allowing any part of the design to be developed at any time.

Other studies also emphasize the importance of representations that do not directly compose solutions. A design rationale is a representation that "expresses elements of the reasoning which has been invested behind the design of an artefact," representing a process behind the designed solution (Buckingham-Shum and Hammond, 1994). An annotation is another type of representation to augment the solution (Marshall, 1998; Correia and Chambel, 1999). Both design rationale and annotations are produced and stored in terms of a solution; they are to be shared to understand the design solution and to be re-designed. They are not primarily for a designer's self-reflection during the design process.

2.3. (I-3) Designers produce externalizations not only to express a solution but also to interpret situations

By observing architectural designers being engaged in sketching, Schoen modeled the design process as "having a reflective conversation with the material" (Schoen, 1983). Without having conscious images of what to draw, a designer's hand, holding a pencil, moves on a sheet of paper, producing lines. The designer sees the emerging lines and shapes, and listens to the situation talking back to him/her. While listening to the situation, the designer continues drawing. The design proceeds as a seeing-drawing-seeing cyclic process. Schoen calls this "reflection-in-action", differentiating it from "reflection-on-action" (1983). The former refers to reflection during the act of externalization, whereas the latter refers to reflection on a resulting externalized representation.

Thus, drawing sketches is not merely externalizing a designer's mental images (Goldschmidt, 1999). Drawing is a means of interpreting and understanding the situation. A designer produces a representation, which is not always associated with a pre-assigned meaning. The meaning is generated through interpretation as the designer interacts with the representation.

This interpretive process is necessary due to the wicked nature of design problems. Formal representations therefore do not serve this type of externalization. Informality and ambiguity of the representations become essential in this respect. Ambiguous representations open up a space for interpretation leading to an understanding of the problem. In studying the role of ambiguity in design, Gaver and colleagues distinguish three broad classes of ambiguity (Gaver et al., 2003): ambiguity of information, ambiguity of context and ambiguity of relationship. The possible space of interpretation exists in the three classes accordingly.

Owing to the mathematical nature of computer systems, software tools have been considered unable to handle ambiguity and informality well. Software systems have been regarded as rigorously governed by univocal rules and primarily limited in their potential to produce senses (Capurro, 1992). From the software development point of view, ambiguity should be avoided.

Although computer tools have no flexibility in constructively interpreting ambiguous representations, they can, however, store and display representations that are ambiguous to people, leaving room for people to interpret them. Tools dealing with informality, such as hand-drawing interfaces (Gross and Do, 1996; Landay and Myers, 2001) and spatial hypertext systems (Marshall and Shipman, 1995), take the approach of allowing users to represent ambiguous, informal representations while the systems partially formalize the representations where possible.

2.4. (I-4) Design proceeds as a hermeneutic circle; designers proceed with projected meanings of representations and gradually revise and confirm those meanings

Snodgrass and Coyne view design as a hermeneutic circle in which parts and the whole co-evolve (Snodgrass and Coyne, 1997). An information design task consists of a gradual production of parts. Parts need to be constructed in order to obtain the whole. While gradually producing parts, a designer thinks of how the parts are related to each other and how the whole would appear. An emerging understanding of the whole determines the meaning of each part. In this sense, parts can be viewed as framing a solution, and the whole as framing a problem.

This cycle proceeds toward the goal by designers who make projections and expectations of the meaning of parts and the whole.

> This preliminary projection is continually revised as the reader or listener penetrates deeper into the meaning of the parts. The projection, at first unclear and only existing in outline, plays back into the interpretations of the parts, requiring their revision even as the projected meaning itself is continually revised in the light of the interpretation and increasing understanding of the parts. By this process of to-and-fro reflection the understanding of the whole gradually emerges. (Snodgrass and Coyne, 1997, p. 76)

In the design process, a designer has a partial understanding of the design situation and produces an externalization as a partial solution by projecting the meaning of the design situation. By interacting with the externalization, the designer interprets the situation in terms of the projected meaning and revises the meaning, having a better understanding about the situation. Thus, "understanding arises by a process of constant revisions" (Snodgrass and Coyne, 1997, p. 86).

In information design proceeding as a hermeneutic circle, projection, anticipation and expectation of the meaning play crucial roles. A designer projects a tentative meaning of the design situation to proceed to the next phase of externalizing a representation and interacts with the representation to interpret the situation based on the projected meaning. A designer proceeds by playing such "what-if" games.

This circular process has been recognized as essential in a creative design process, and computational tools have been developed for the process. Terry and Mynatt found that graphic designers using image-processing software repeat a cycle of applying and undoing commands and comparing the different effects. This observation has informed the design of SideViews, which they developed as a previewing tool for an image-processing software. SideViews allows users to compare possible consequences of applying different commands by exploring multiple potential future realities (Terry and Mynatt, 2002). They have also developed ParallelPies to remember commands applied in the past, which allows users to manipulate multiple versions at a time (Terry et al., 2004). Lunzer has developed the notion of a subjunctive interface, which means an interface to "embody simultaneously all the what-if situations that the user wished to explore, giving users freedom to keep options open, ease of viewing a range of available outcomes and ease of comparing outcomes and selecting preferred ones" (Lunzer, 1999).

Although those systems allow designers to compare different projected results, they do not provide representations to externalize how sure the designers are about the projection. When a designer becomes certain about the projection of the meaning, he/she makes a commitment. Representations for delayed commitment as well as representations for various degrees of commitment would be necessary for the process.

3. Principles for tools for fostering creativity

The previous section outlined the four issues in supporting the early stages of information design with computer tools as media for externalization. In recognizing these issues, this section identifies interaction design principles for tools to foster creativity in such design tasks.

3.1. Interaction design of tools for design

Let us first discuss the implications of each of the four issues (I-1, -2, -3 and -4) outlined in the previous section.

3.1.1. (I-1) Available means of externalizations influence designers in deciding which courses of action to take

The first issue is related to the essence of our approach. Due to the influence of externalizations, tools need to be designed from the viewpoint of what representations a user wants to generate and interact with through what operations in the early stages of design. In the design and development of computer tools for supporting design, we take the interaction design-based approach in which tools are specified from the point of view of a user: what representations a user can interact with and how. The interaction design-based approach is in contrast with the conventional software development approach in which necessary functionality is first determined, followed by the design of an interface as a means to access the functionality.

With an appropriate interaction design, an application system as a cognitive tool guides, encourages, and permits a user in taking a certain course of action or state of mind fostering his/her creativity. With an inappropriate interaction design, an application system may distract, discourage and prohibit a user from taking a certain course of action or state of mind, thus obstructing creativity.

Note that whether a particular interaction design is appropriate for fostering creativity cannot be judged out of context. We argue that there is no single tool for creativity that universally works for everybody in every possible situation. Different designers might prefer different types of systems designed with different interaction styles. Although our interaction design-based approach is generally applicable for the design and development of tools for fostering creativity, a specific interaction design instance may work better for a certain group of designers than for others.

3.1.2. (I-2) Designers generate and interact with not only a partial representation of the final artefact but also various external representations

This second issue is the reason the interaction design of tools for fostering creativity in the early stages of information design becomes a real challenge. Designers need representations both for problems and for solutions. Designers need representations for partially constructed parts as well as for the emerging whole. Fabricating a final form as a representation in a tool is a quite straight-forward task. If the design domain has representations for the early stages of design that have been traditionally used within a practice, such representations can be replicated in a tool. In contrast, if the targeted information design domain does not have such traditionally exercised representations for the early stages of design, or if one wants to go beyond the current practice, designing a new representation and an interaction style can be quite difficult. The remaining two points provide guiding principles for the design of representations.

3.1.3. (I-3) Designers produce externalizations not only to express a solution but also to interpret situations

3.1.4. (I-4) Design proceeds as a hermeneutic circle; designers proceed with projected meanings of representations and gradually revise and confirm those meanings

The third and fourth issues are related to the role of externalizations and the designer's interactions with externalizations in the early stages of information design. The third point states that a designer produces a representation in order to interpret the situation. The fourth point states that a designer produces a representation based on the projected meaning of the situation.

These two statements might initially seem to be contradictory. If we view a designer as repeating a seeing-drawing-seeing process, however, the two statements simply refer to the cyclic process of interpreting the situation and producing a representation. For representations to drive the cycle going forward, the representations must be easily interpreted in a wide variety of ways as well as easily remembered with their projected meanings.

3.2. *The three interaction design principles*

The above discussions have led us to identify three interaction design principles for tools for fostering creativity in the early stages of information design. They are (P-1) *interpretation-rich representations*, (P-2) *representations with constant grounding* and (P-3) *interaction methods for hands-on generation and manipulation of the representations*.

These principles are derived by taking the notion of cognitive load as a central concern. Creative design is a cognitively intensive human activity. Cognitive resources have limitations, however. Designers need tools to generate representations and interact with them, but using tools consumes some of the cognitive resources, which then becomes cognitive overhead. The more cognitive resources the use of tools asks for, the fewer cognitive resources designers can spend on their creative thinking. Our approach toward the design of computer tools for supporting creativity is to diminish, if not completely remove, cognitive overhead in using tools so that designers can spend more cognitive resources for their creative thinking, such as listening to the situation talking back and projecting a design context.

3.2.1. *(P-1) Interpretation-rich representations*

Designers produce representations and interact with them in order to interpret the situation. The process is not to narrow a possible design space but rather to discover new aspects of the situation and explore them through interacting with the representation.

Representations for such an activity need to easily generate a variety of interpretations. Designers need representations that are interpretation-rich; they should be able to be perceived and recognized in various ways by designers. Examples include graphical representations, spatial positioning of objects and metaphors.

Graphical representations, as used in sketches, have been used in creative design practices (Arnheim, 1969; Lawson, 1994; Gross and Do, 1996). When looking at a sketch, a designer not only perceives visual representation, but actively finds meaning from the representation (Arnheim, 1969). This constructive perception process is regarded as a driving force for a designer's creativity (Suwa, 2002). Sketches afford ambiguity in their symbolism, context and relevance (Gaver et al., 2003) by expressing "something like this" or "around here".

Spatial positioning of objects is another example of representations that afford rich interpretations (Marshall and Shipman, 1995). People use the desktop space and icons on a computer display to arrange files. Spatial relationships among freely positioning objects in a space may be viewed as linearly ordered, spatially arranged, or grouped together. Spatial positioning of objects provides informal means to express emerging relationships among the positioned objects.

Yet another example of interpretation-rich representations is the metaphor. Metaphors are capable of giving a new understanding of experience (Lakoff and Johnson, 1980). In the field of poetry, the creation of metaphors is regarded as "a production of sense, expanding language within language, or a heuristic function, discovering new possible aspects of reality" (Capurro, 1992).

3.2.2. (P-2) Representations with constant grounding

Designers project the meanings of the situation, produce a new representation and interpret the emerging situation. The projection is then revised, leading to a better understanding of the design situation. Representations for this type of activity therefore need to easily remind the designer of what the projected meanings are, and easily express meta-comments (Kintsch and van Dijk, 1978), representing how much one is sure about the projection, or the degree of commitment.

As a designer interprets the situation by interacting with a representation, the designer needs to remember the projected meaning on which the interpretation was based. Representations therefore need to be easily mapped to a projected meaning leading to a design solution.

It is therefore necessary for representations to have constant grounding. Although representations should be interpretation-rich, they also have to provide constant grounding so that designers can use the representational grounding to express their projected meanings, various degrees of commitment about the projected meanings, and remaining ambiguities and uncertainties about the representation.

Visual and perceptual properties of representations are often used to provide constant grounding. For instance, a two-dimensional layout of a sketch drawn on a sheet of paper may be mapped to a floor plan. A circular mark drawn in the layout may refer to the area where the designer thinks something is necessary. Two objects with visual resemblance may be easily understood to resemble each other in their semantics. Roughly drawn lines may refer to less commitment than carefully drawn lines.

The important point here is that the designer decides how to use representations for what he/she means. In doing so, the designer needs assurance that the meaning of the representational properties remain the same by being governed by constant grounding schemes. Although such grounding schemes need not be formally derivable, in these cases, computational tools as media for externalization can go beyond traditional representations, such as pencil-and-paper drawings, clay models, or plastic models.

3.2.3. (P-3) Interaction methods for hands-on generation and manipulation of the representations

Designers need to be able to express representations as they intend to, not in terms of semantics but in terms of perceptual properties. As argued before, using tools requires cognitive load. Ideally, a designer should be able to represent what he/she wants to represent and to interact with representations in a way he/she wants to interact with them. An example is again hand-drawing sketches. The power of sketching for creative design resides not only in its externalized representation, but also in the act of producing the representation (Nakakoji and Yamamoto, 2003). In hand-drawing sketching, without having conscious images of what to draw, a designer's hand, holding a pencil, moves on a sheet of paper, producing lines. The designer, seeing the emerging lines and shapes, reflects and proceeds with the design. The designer can adjust the manner of producing a representation in a very subtle manner; for instance, with the angle between a pencil and paper, or the force put on the pencil against the sheet of paper. By looking at the effect of what is drawn, the designer can interact with the representation, such as drawing thicker lines or retracing already drawn shapes. Using a pencil and paper to produce desired effects is not an easy task; humans learn how to do it when they are small children, but they do not necessarily learn it well. With training, however, a person can always produce a desired visual effect in a consistent manner. This is why some designers have to use a specific type of the pencil with a specific size of the paper.

When it comes to having media for externalization on a computer system, a designer has to be able to feel as if he/she interacts with the representations rather than with the computer system; computer systems need to stay invisible to the user. However, in many of the current user interfaces, this is not easy to achieve.

User interfaces have been designed based on the model of a user's cognitive process: to specify a goal, make a plan to achieve the goal, execute the plan and evaluate the result (Norman, 1988). This type of cognitive process is not plausible when we look at designers producing representations and interacting with them in the early stages of information design tasks. The goal is not to produce a representation but to interpret emerging meanings from the representation. The goal of what to represent, therefore, changes as the designers go through reflection-in-action processes.

This misfit between the traditional user interface design framework and necessary interaction designs for tools for fostering the early stages of information design has resulted in a situation in which designers encounter difficulty in making sketches with software tools for drawing even when using touch-screen displays with styli. With most software tools for drawing, a user first has to decide which palette to use to draw which object with which color, and with which thickness of a brush. Once a user decides which object to draw and starts drawing, he/she cannot freely change the decision without stopping the drawing process. This obstructs the flow of creative thinking.

The command-based, turn-taking operation style is thus not appropriate for tools as media for creative design. Interaction needs to be designed so that a user has a discourse with a representation, not with a computer system. Notions developed in human–computer interaction, such as direct manipulation (Shneiderman, 2000) and instrumental interaction (Beaudouin-Lafon, 2000) need to be taken into account to enable hands-on generation and manipulation of the representations.

Direct manipulation is the basis of an interaction model using a graphical user interface (GUI). The user's operation (such as mouse movement) is interpreted by a system to produce an effect (usually visual) in a consistent manner. The system gives feedback to the user within a few hundred milliseconds after the user's operation, so that the user feels that there is a cause–effect relationship between what the user does and what the system does. Direct manipulation goes from the very low-level interaction (e.g., moving a mouse causes cursor movement) to the quite high-level interaction (e.g., specifying a database query with a number bar causes visual data display updates). In instrumental interaction, the instrument is a computational manipulatable object often displayed on a screen to manipulate objects of concern. Using a scroll bar to scroll the contents of the window is an example of instrumental interaction.

These three interaction design principles are not meant to be applied individually. They have to be taken into account as an integral whole in the design and development of tools for fostering creativity in the early stages of information design.

The next section illustrates these three principles in existing tools for design.

4. Reinterpreting ART#001 from the interaction design principles

This section examines how these three interaction design principles, P-1, -2 and -3, are materialized in ART#001 (Fig. 1), an existing tool for the early stages of writing.

The foundation of ART#001 was designed in 1998 (Yamamoto, 2001). This tool uses spatial hypertext as an interaction method to create a document. Despite of a number of updates since 1998, the fundamental interaction model has not been changed from its initial version.

The tool has been available through our Web site,[1] and has been used by dozens of actual users. We have conducted user studies of the tool, by means of videos and an eye-tracking system, and have demonstrated how the tool helps writers in their early stages of the writing task. The results and discussions on those results are reported in the literature (Nakakoji et al., 2000; Yamamoto, 2001; Yamamoto et al., 2002a).

ART#001 was developed as a part of the ART project, which stands for Amplifying Representational Talkback (Nakakoji et al., 1998; Yamamoto, 2001; Nakakoji and Yamamoto, 2001). The ART project, which we have been carrying out since 1997 (Nakakoji et al., 2002; Yamamoto et al., 2002b), is an interaction design-oriented application development project with a strong emphasis on the ART concept.

In interaction design, there is no best solution but only satisficing ones. Interaction design needs to resolve conflicting requirements and constraints. To identify

[1] http://www.kid.rcast.u-tokyo.ac.jp/systems/ARTware/

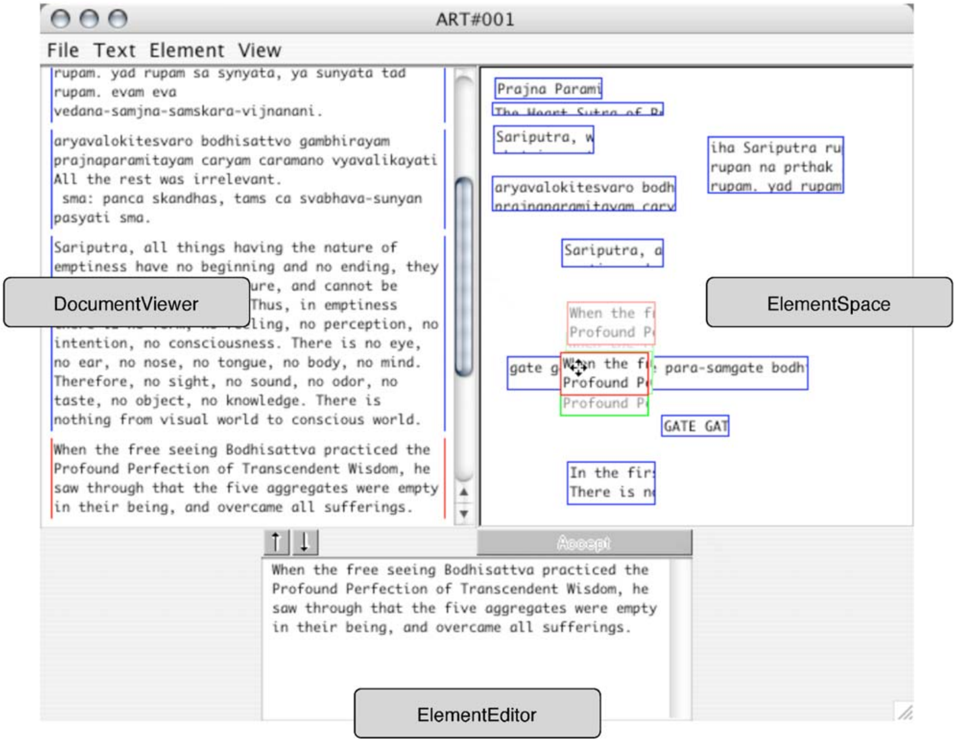

Fig. 1. ART#001, a tool for early stages of writing. The tool comprises three parts, ElementEditor, ElementSpace and DocumentViewer.

relationships among such requirements and constraints, and to prioritize them, it is necessary to apply a design concept throughout the project so that the design decisions would result in a coherent whole (Preece et al., 2002). The ART concept stresses Schoen's view of design as a conversation with the material (Schoen, 1983). Two concerns are related to the ART concept. First, to make representations so that a user can easily externalize what he/she wants to externalize with minimal cognitive load, and second, to make representations so that a user can easily understand and interpret them in a way he/she likes (Yamamoto, 2001). The interaction design of ART#001 had been conducted before we identified the above interaction design principles for fostering creativity in early stages of information design. The goal of this paper is not to argue for the validity of ART#001, but to reinterpret the tool from these interaction design principles.

4.1. An overview of ART#001

The basic idea behind ART#001 is to view writing as the design of linear textual information. A document is a linearly ordered sequence of text chunks of various sizes. The early stages of writing are viewed as consisting of the following activities:

- to create or modify a text chunk that would constitute a document to be written,
- to represent emerging relationships among chunks,
- to represent the emerging function of a chunk with regard to the whole,
- to reflect and confirm how chunks flow in a linear order, and
- to identify missing chunks.

A user of ART#001 edits, modifies, or manipulates each text chunk and manipulates them in two-dimensional space. The interaction with the representations is supported through direct manipulation of the chunks.

The interaction model of ART#001 comprises three components (in the description, text chunks are called elements):

1. *ElementEditor*: for creating and modifying an element.
2. *ElementSpace*: for specifying relationships among elements.
3. *DocumentViewer*: for viewing a document under construction made up of elements that have been created.

In ART#001, a user produces a text chunk (i.e., words, sentences, paragraphs) as an element in ElementEditor. A constructed element appears as an element in ElementSpace. Contents of each element positioned in ElementSpace are appended from top to bottom and displayed in DocumentViewer. When selecting an element in ElementSpace by clicking on it, the DocumentViewer area scrolls so that the corresponding element becomes visible in the DocumentViewer. Dragging elements in ElementSpace changes the vertical relationships among the elements, which will dynamically move the corresponding elements in DocumentViewer. A user may modify any element by selecting it in either ElementEditor or DocumentViewer. Multiple elements can be merged together, or a single element can be split. The content of the document displayed in DocumentViewer can be saved as plain text or in the HTML format.

4.2. Application of the principles to account for ART#001

A number of users have been using ART#001 since 1999. Fig. 2 shows actual screen images of representations produced by different users. The eye-tracking system employed in our user studies has shown that users were engaged in a reflection-in-action process interacting with elements in ElementSpace (Fig. 3) (Nakakoji et al., 2000).

This subsection takes the interaction design principles identified in Section 3 and applies them to ART#001. As stated earlier, this paper is not about demonstrating how ART#001 works to foster creativity, which was a topic in other papers (Nakakoji et al., 2000, 2002; Yamamoto et al., 2001, 2002a, b). Rather, the goal of

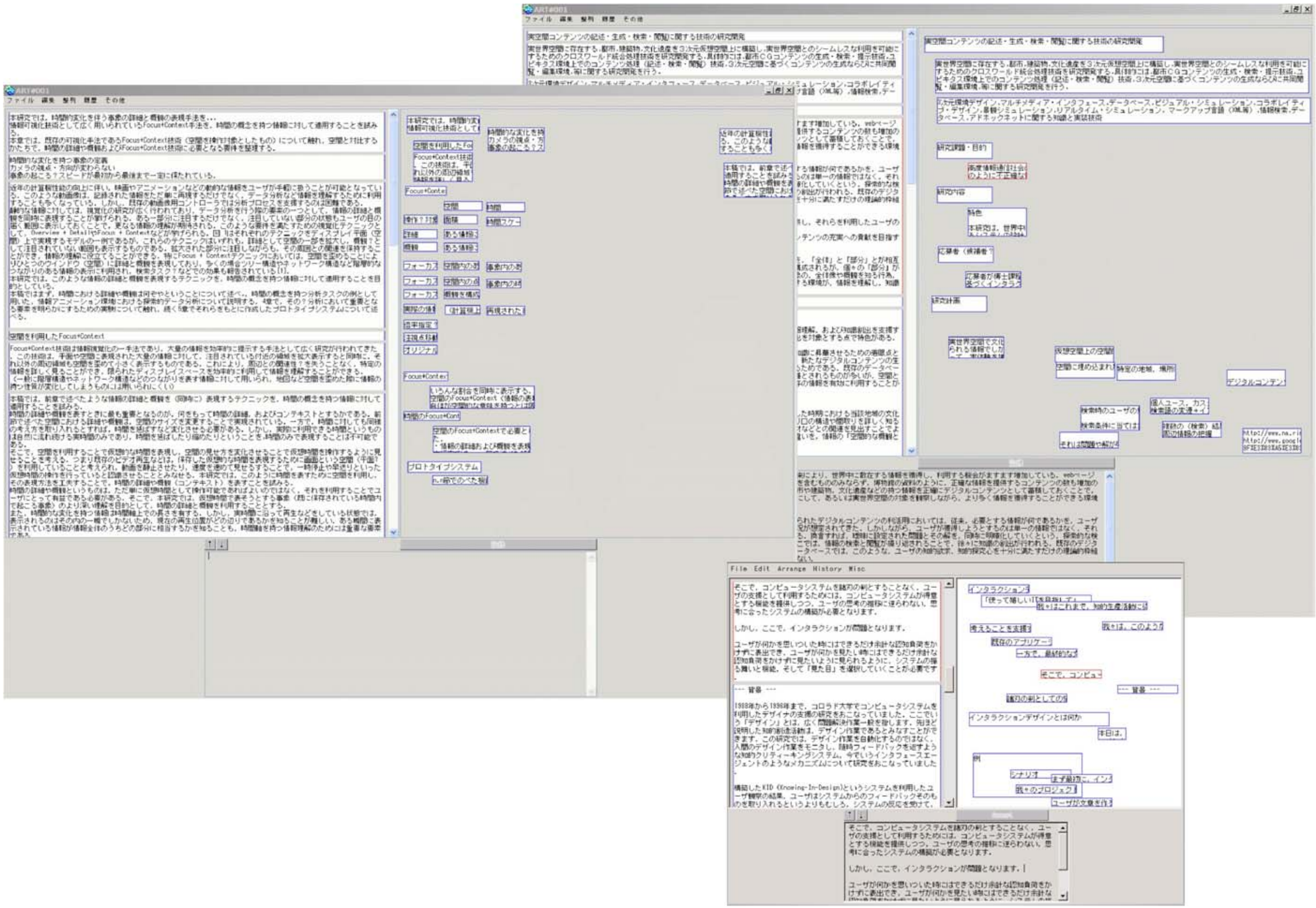

Fig. 2. Representations produced in ART#001. Different users used ElementSpace as a spatial hypertext medium and produced a variety of representations. The figure includes three different versions of ART#001.

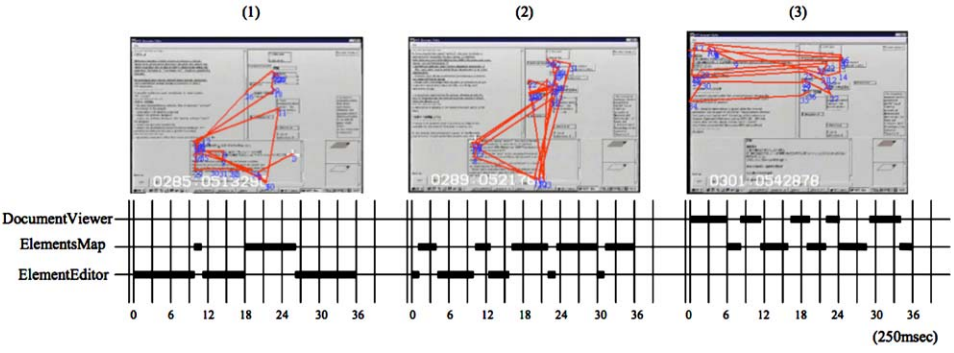

Fig. 3. Eye-tracking data on the use of ART#001 (from Nakakoji et al., 2000).

this subsection is to demonstrate the power of interaction principles to describe why ART#001 works for users in early stages of writing.

4.2.1. (P-1) Interpretation-rich representations

ElementSpace of ART#001 allows a user to produce spatial positioning of objects, which serve as interpretation-rich representations. The size, shape, or location of an element in the space and relative relationships among elements can be interpreted in a variety of ways.

One of the user studies found that subjects keep slightly moving elements and changing shapes of elements in considering what the order of the elements should be. As Marshall and Shipman and other spatial hypertext researchers have found (Marshall and Shipman, 1995), spatial positioning of objects serves as representations that are interpretation-rich, fostering creative exploration.

4.2.2. (P-2) Representations with constant grounding

What distinguishes ART#001 from other spatial hypertext tools is that the vertical relationships of elements in ElementSpace are always interpreted in the linear relationships among elements in DocumentViewer. When a user drags an element in ElementSpace, the corresponding element in DocumentViewer follows the movement to maintain consistency between the vertical relationships within ElementEditor and those in DocumentViewer. ElementSpace does not have a scroll bar, so all elements in ElementSpace are always visible. When more area is needed to position a text chunk, the user simply drags the element toward the right or bottom edge of ElementSpace; ElementSpace then zooms out until all the elements, including the one just dragged, become visible in ElementSpace while maintaining the relative positions among the other elements in ElementSpace.

A user of ART#001 can freely position text chunks in ElementSpace, except the vertical relationship that is interpreted by the system as the order in linearlizing chunks. The horizontal relationship, size, or even the vertical distance between two elements do not affect the content of DocumentViewer, allowing users to manipulate the spatial layout of elements in the space as representations affording rich interpretations. For instance, one can leave room between two vertically aligned elements in ElementSpace to represent that something is missing between the two elements.

Thus, ElementSpace, together with DocumentViewer makes it possible to have both interpretation-rich representations and representations with constant grounding.

A few more design features help a user of ART#001 to easily provide grounding for the representations of text chunks, including the visual representations of text chunks in ElementSpace, DocumentViewer and in ElementEditor.

In ART#001, only 10 percent of the text in the same font size is shown by default as an element in ElementSpace, not the entire text content. This gives the user an approximation of which text chunk each element in ElementSpace represents. It is possible to show the entire content in an element in ElementSpace by using a much smaller font size, but it might become more difficult to read. Also, if the user changes the content of an element, the look of the element in ElementSpace might completely change, losing its visual identity. It is possible for a user to label each element, but that would require additional cost for the user to understand the correspondence between each element in ElementSpace and in DocumentViewer.

ElementEditor allows the user to edit the content of the text element. Currently, DocumentViewer and ElementEditor use exactly the same font size with the same column width, making sure that the line break occurs in the same position within the text. Again, the equivalence of the visual appearance of elements in ElementEditor and DocumentViewer is important to ensure that the user easily understands the relationships among the representations.

4.2.3. (P-3) Interaction methods for hands-on generation and manipulation

With ART#001, a user simply grabs an element in ElementSpace and drags it to change the position of the element within the document. This is a different interaction method of manipulating text compared to a traditional editor, for which a user selects a region of text to move, scrolls down or up while dragging the selected text to the insertion point and pastes it into position.

Generation and modification of an element in ElementEditor seem to give a representational distance compared to directly editing text within DocumentViewer or in ElementSpace. In contrast, the content of text within DocumentViewer or in ElementSpace does not change until a user pushes the "Accept" button, when the changes are committed to update the content. This allows a user to edit the content of an element by comparing it with the content of the current state.

4.3. Toward a general framework

The previous section demonstrated that the interaction model underlying ART#001 conforms to the design principles outlined in Section 3. In fact, the interaction model consisting of ElementEditor, ElementSpace and DocumentViewer provide representations and interactions applicable not only for the text domain but

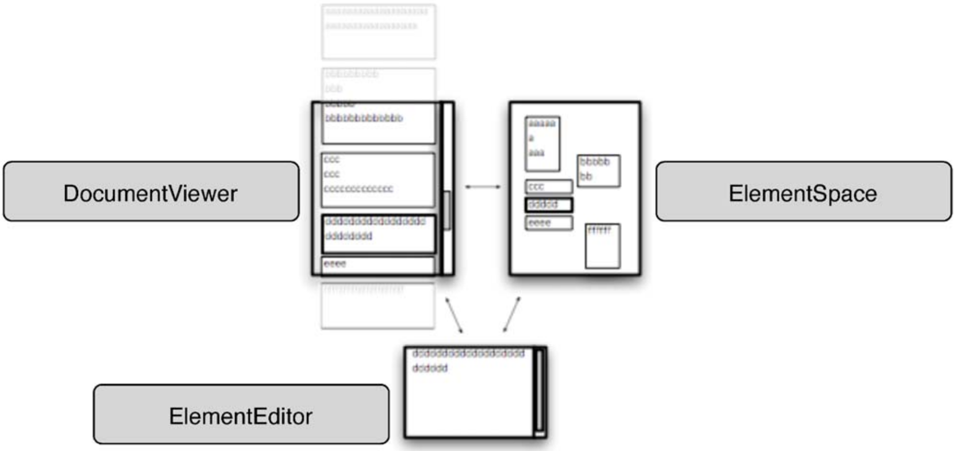

Fig. 4. An interaction model for a tool for early stages of linear information design.

also for other linear information design domains. Fig. 4 illustrates the general framework for the interaction model.

By looking at text chunks as elements of the information artefact in general, the same interaction model can be applied to other information design domains. That is, a user using the system based on this interaction model produces linear information by creating element-by-element using ElementEditor. All the elements created in ElementEditor are located in ElementSpace. Elements in ElementSpace are appended by the vertical order in ElementSpace. Thus, a user can express the location of the element in terms of the whole document by positioning the element in ElementSpace. Horizontal relationships among elements in ElementSpace as well as the size of element in ElementSpace do not affect the document to be designed. A user can associate any semantics to the horizontal positioning of elements and resizing elements. Selecting an element in ElementSpace will identify the corresponding element in DocumentViewer and vice versa. Dragging elements in ElementSpace results in the dynamic reordering of elements in DocumentViewer. As mentioned earlier, ElementSpace always shows the document as a whole, so it is not equipped with any scrollbars. A user may zoom out the space by dragging elements toward the edge of ElementSpace. A user thus directly manipulates elements in ElementSpace, and the document under construction is edited by using the spatial positioning of elements in ElementSpace as an instrument (Yamamoto, 2001).

In place of text chunks, the ART#004 (Fig. 5) treats movie segments as elements. A user of ART#004 can decide which part of an original movie to use for a design task, and can decide in what order to append the segmented movies. ART#004 uses the horizontal relationships among elements positioned in ElementSpace. Thus, DocumentViewer is now located at the top of the window, and ElementSpace is located at the bottom of the window. ElementEditor has an interface so that a user can browse any movie material and decide which part to use by inserting two

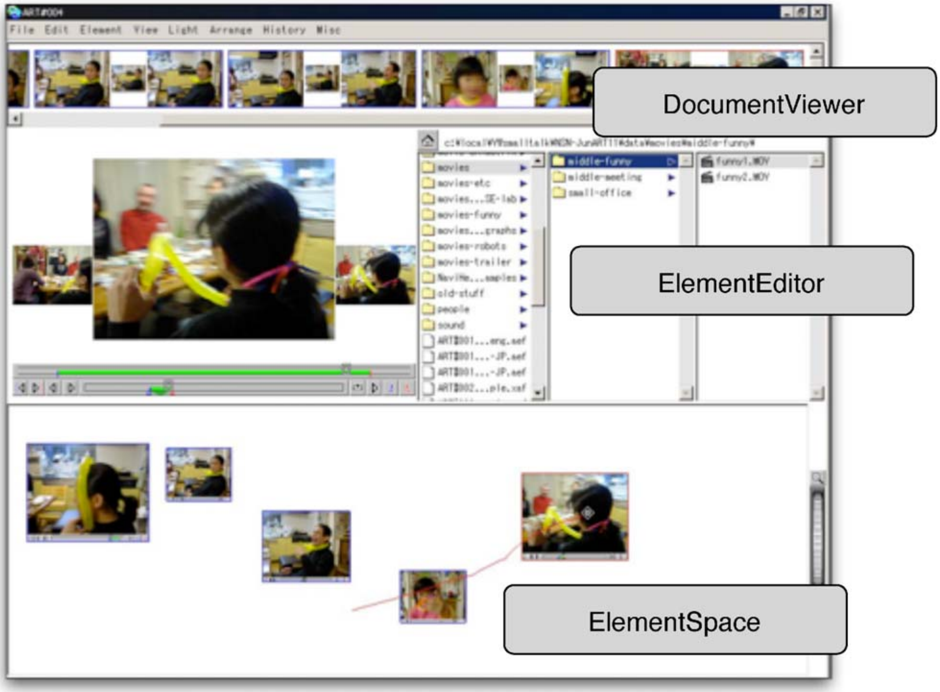

Fig. 5. ART#004, a tool for early stages of movie editing.

segmentation points. When segmented, a user can drag and drop a movie segment from ElementEditor to ElementSpace and position it as an element, which is represented as a thumbnail image of the segmented movie. Each element is represented as a three-tiered thumbnail image in DocumentViewer, consisting of the initial frame, the frame visible when segmented and the ending frame. Positioned elements in ElementSpace are appended from left to right and displayed in DocumentViewer. When a user drags an element in ElementSpace, the change of the horizontal relationships among elements is dynamically reflected in the order of the three-tier thumbnails in DocumentViewer. The appended movie can be browsed in a movie player or saved as a new movie.

ART#003 (Yamamoto, 2001) also uses movie segments as elements (Fig. 6), but helps a user analyse video data, such as in empirical video analysis tasks. In ART#003, a user may segment a part of the original material (video data) as he/she finds interesting portions, and drags it into ElementSpace. Elements are represented as thumbnails in ElementSpace. In addition, a user can textually annotate the segmented movie in the window provided beneath ElementEditor. Elements in DocumentViewer are represented as a set of the movie thumbnails and their associated text annotations. The tabular form of DocumentViewer can be saved in an HTML format and can be browsed through a Web browser.

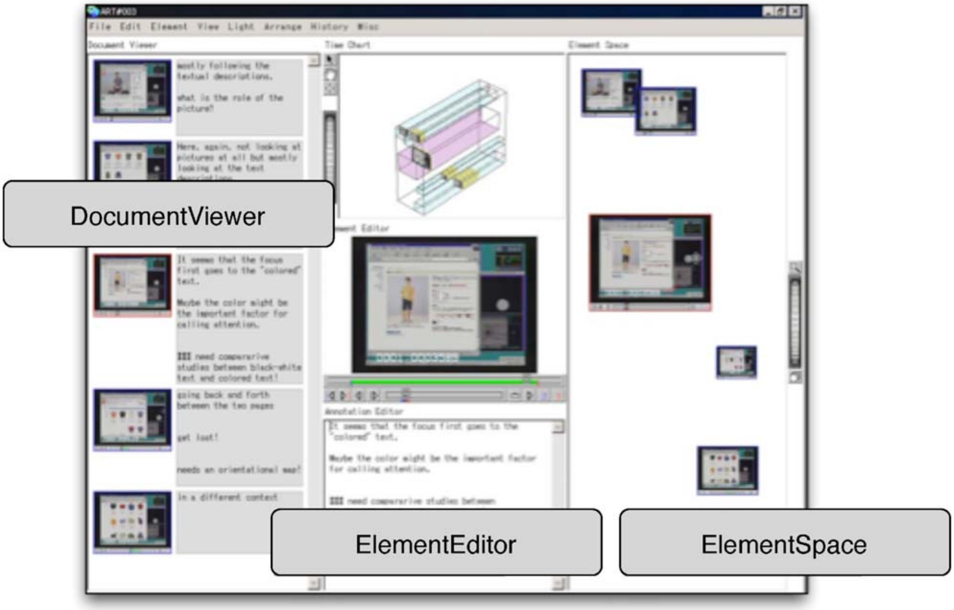

Fig. 6. ART#003, a tool for early stages of video analysis.

5. Conclusions

This paper has presented our view of computer tools as media for externalization, and the interaction design-centered design and development of such tools. We have discussed issues in the design of tools for the early stages of information design and identified three interaction design principles for the design and development of tools for fostering creativity in such design tasks: representations with rich interpretations, representations with constant grounding and interaction methods of hands on generation and manipulation of representations.

The majority of existing research on application systems has been aimed at the development of solutions to research problems. In this framework, research on tools for creativity has explored what properties and functions tools ought to have in order to support creativity. The necessary functionality of the tool has been identified and implemented, and the tool was then evaluated in terms of how subjects perform using it. We examined issues derived from the nature of the early stages of information design as cognitively intensive design tasks. Based on the theoretical account, we argued that the interaction design-centered approach is essential for the development of tools serving as external media with which designers can interact. We then derived three interaction design principles.

Our contribution for the research on tools of creativity resides in the interaction design principles. Our position is that there is no single solution toward creativity support research. Admitting that many tools can exist for the different styles of designers, we argue that such different tools have to be designed by applying the three principles we have identified. ART#001 has been demonstrated as a tool for fostering creativity in a designer's early stages of writing by reinterpreting its interaction design using the principles, but we are aware that ART#001 is not for everybody. Some users do not like the style of element-based writing. People with different cognitive styles should be able to choose from different interaction design instances. This finding will next lead to the development of appeal-driven information technologies (Winograd, 1995).

An application system as a cognitive tool guides or distracts, encourages or discourages and permits or prohibits a user from taking a certain course of action and state of mind, depending on what representations the designer interacts with. This fosters or obstructs a designer's creative thinking process as a flow (Csikszentmihalyi, 1990). Tools are not only something that users use to achieve their goals; they can also be bearers of representations that users generate and interact with. In designing everyday home computing systems, Hallnaes and Redstroem argue that we need to change the view of computer systems from ''efficient use'' to ''meaningful presence,'' where an aesthetic point of view is required as a logic of expression (Hallnaes and Redstroem, 2002). Similarly, we argue for a view of computer systems from ''rich functionality'' to ''creative interaction''. Our approach toward the development of computer tools for fostering creativity needs to be interpreted and further elaborated within this point of view.

Acknowledgements

We would like to thank Jonathan Ostwald, Mark Gross, Yunwen Ye, Frank Shipman, Mark Bernstein, Jim Rosenberg, Simon Buckingham-Shum, Atsushi Aoki, Yoshiyuki Nishinaka, Mitsuhiro Asada, Hiroko Asaoka, Linda Candy, Ernest Edmonds, Akio Takashima and Michael Terry for their intensive discussions on the topic as well as the encouragement to pursue this project. This research is partly supported by MEXT Grant-in-Aid for Scientific Research A(1), 2004, 16200008.

References

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

Beaudouin-Lafon, M., 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. Proceedings of Human Factors in Computing Systems (CHI 2000), 446–453.

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

Buckingham-Shum, S., Hammond, N., 1994. Argumentation-based design rationale: what use at what cost? International Journal of Human–Computer Studies 40, 603–652.

Candy, L., Hori, K., 2003. The digital muse: HCI in support of creativity: ''creativity and cognition'' comes of age: towards a new discipline. ACM Interactions 10 (4), 44–54.

Capurro, R., 1992. Informatics and hermeneutics. In: Floyd, C., Zuellinghoven, H., Budde, R., Keil-Slawik, R. (Eds.), Proceedings of Software Development and Reality Construction, pp. 363–375.

Cooper, A., 1999. The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. SAMS, Indianapolis, IN.

Correia, N., Chambel, T., 1999. Active video watching using annotation. Proceedings of ACM Multimedia '99 Part 2, 151–154.

Csikszentmihalyi, M., 1990. Flow: The Psychology of Optimal Experience. HarperCollins Publishers, New York.

de la Rocha, O.L., 1985. The reorganization of arithmetic practice in the kitchen. Anthropology and Education Quarterly 16 (3), 193–198.

Edmonds, E.A., Candy, L., 1993. Knowledge support systems for conceptual design: the amplification of creativity. HCI International '93 2, 350–355.

Fallman, D., 2003. Design-oriented human–computer interaction. Proceedings of Human Factors in Computing Systems (CHI 2003), 225–232.

Fischer, G., Nakakoji, K., 1994. Amplifying designers' creativity with domain-oriented design environments. In: Dartnall, T. (Ed.), Artificial Intelligence and Creativity Part V. Kluwer Academic Publishers, The Netherlands, pp. 343–364.

Gaver, B., Beaver, J., Benford, S., 2003. Ambiguity as a resource for design. Proceedings of Human Factors in Computing Systems (CHI 2003), 233–240.

Goldschmidt, G., 1999. Design. In: Mark, A.R., Steven, R.P. (Eds.), Encyclopedia of Creativity, vol. 1. Academic Press, San Diego, CA, pp. 525–535.

Graham, P., 2003. Hackers and Painters, May 2003, http://www.paulgraham.com/hp.html

Green, T.R.G., 2000. Instructions and descriptions: some cognitive aspects of programming and similar activities. Proceedings of Working Conference on Advanced Visual Interfaces (AVI 2000) 4, 21–28.

Gross, M.D., Do, E.Y.L., 1996. Ambiguous intentions: a paper-like interface for creative design. Proceedings of Symposium on User Interface Software and Technology (UIST '96), 183–192.

Hallnaes, L., Redstroem, J., 2002. From use to presence: on the expressions and aesthetics of everyday computational things. ACM Transactions on Computer–Human Interaction 9 (2), 106–124.

Kintsch, W., van Dijk, T.A., 1978. Towards a model of text comprehension and production. Psychological Review 85 (5), 363–394.

Lakoff, G., Johnson, M., 1980. Metaphors We Live By. The University of Chicago Press, London, UK.

Landay, J.A., Myers, B.A., 2001. Sketching interfaces: toward more human interface design. IEEE Computer 34 (3), 56–64.

Lawson, B., 1994. Design in Mind. Architectural Press, MA.

Lunzer, A., 1999. Choice and comparison where the user wants them: subjunctive interfaces for computer-supported exploration. In: Proceedings of IFIP TC13 International Conference on Human–computer Interaction (INTERACT '99), 474–482.

Marshall, C.C., 1998. Toward an ecology of hypertext annotation. Proceedings of Hypertext 98, 40–49.

Marshall, C.C., Shipman, F.M., 1995. Spatial hypertext: designing for change. Communications of the ACM 38 (8), 88–97.

Nakakoji, K., Fischer, G., 1995. Intertwining knowledge delivery, construction, and elicitation: a process model for human–computer collaboration in design. Knowledge-Based Systems Journal 8 (2–3), 94–104.

Nakakoji, K., Yamamoto, Y., 2001. What does the representation talk back to you? Knowledge-Based Systems Journal 14 (8), 449–453.

Nakakoji, K., Yamamoto, Y., 2003. Toward A taxonomy of interaction design techniques for externalizing in creative work. In: Stephanidis, C., Jacko, J. (Eds.), Proceedings of HCII 2003, Theory and Practice (Part II), vol. 2. pp. 1258–1262.

Nakakoji, K., Yamamoto, Y., Suzuki, T., Takada, S., Gross, M.D., 1998. Beyond critiquing: using representational talkback to elicit design intention. Knowledge-Based Systems Journal 11 (7–8), 457–468.

Nakakoji, K., Yamamoto, Y., Reeves, B.N., Takada, S., 2000. Two-dimensional positioning as a means for reflection in design. Proceedings of Design of Interactive Systems (DIS 2000), 145–154.

Nakakoji, K., Yamamoto, Y., Aoki, A., 2002. SRA, NAIST & JST. ACM Interactions 9 (2), 99–102.

Nakakoji, K., Satoh, H., Yamamoto, Y., Aoki, A., Asaoka, H., 2003. Interaction design for molecular reaction prediction. Proceedings of IPSJ-SIGHI-106, 79–86 (in Japanese).

Norman, D.A., 1988. Psychology of Everyday Things. Basic Books, New York.

Norman, D.A., 1993. Things That Make Us Smart. Addison-Wesley, Reading, MA.

Preece, J., Rogers, Y., Sharp, H., 2002. Interaction Design: Beyond Human–Computer Interaction. Wiley, New York.

Rittel, H.W.J., Webber, M.M., 1984. Planning problems are wicked problems. In: Cross, N. (Ed.), Development in Design Methodology. Wiley, New York, pp. 135–144.

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

Shirouzu, H., Miyake, N., Masukawa, H., 2002. Cognitively active externalization for situated reflection. Cognitive Science 26 (4), 469–501.

Shneiderman, B., 2000. Creating creativity: user interfaces for supporting innovation. ACM Transactions on Computer–Human Interaction 7 (1), 114–138.

Simon, H.A., 1996. The Sciences of the Artificial, third ed. MIT Press, Cambridge, MA.

Snodgrass, A.B., Coyne, R.D., 1997. Is designing hermeneutical? Architectural Theory Review 2 (1), 65–97.

Suwa, M., 2002. Constructive perception: an expertise to use diagrams for dynamic interactivity. In: Proceedings of 24th Annual Conference of the Cognitive Science Society, p. 55.

Terry, M., Mynatt, E., 2002. Side views: persistent on-demand previews for open-ended tasks. In: Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST 2000). ACM Press, New York (pp. 71–80).

Terry, M., Mynatt, E., Nakakoji, K., Yamamoto, Y., 2004. Variation in element and action: supporting simultaneous development of alternative solutions. In: Proceedings of Human Factors in Computing Systems (CHI 2004), 711–718.

Winograd, T., 1995. From programming environments to environments for designing. Communications of the ACM 38 (6), 65–74.

Winograd, T., Flores, F., 1986. Understanding Computers and Cognition: A New Foundation for Design. Ablex Publishing Corporation, Norwood, NJ.

Yamamoto, Y., 2001. Amplifying representational talkback: interactive systems using spatial positioning to support early stages of information design. Doctoral Dissertation, Department of Information Science, Nara Institute of Science and Technology, Nara, Japan (in Japanese).

Yamamoto, Y., Aoki, A., Nakakoji, K., 2001. Time-ART: a tool for segmenting and annotating multimedia data in early stages of exploratory analysis. Extended Abstract of CHI 2001, 113–114.

Yamamoto, Y., Nakakoji, K., Aoki, A., 2002a. Spatial hypertext for linear-information authoring: interaction design and system development based on the ART design principle. Proceedings of Hypertext 2002, 35–44.

Yamamoto, Y., Nakakoji, K., Aoki, A., 2002b. Visual interaction design for tools to think with: interactive systems for designing linear information. Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2002), 367–372.

Zhang, J., 1997. The nature of external representations in problem solving. Cognitive Science 21 (2), 179–217.