Reflections on Spatial Writing in Place

Jim Rosenberg
555 Davidson Road
Grindstone, Pa 15442
E-mail: jr@amanue.com

ABSTRACT

A spatial hypertext system which allows word objects to play in place avoids the need for document authors to extrapolate run-time effects when authoring. This is most notable at small-scale granularities in bottom-up writing. Support for grouping needs to be flexible and support degrees of extent to which a group is finished; grouping by spatial parsing has aesthetic consequences that may not be under the control of a document author. Grouping structure may need to be invisible, and grouping may be needed more as a convenience of an author to preserve spatial relationships than to exhibit structure. There is a powerful feedback effect on what is composed next from materials already composed; this feedback is drastically influenced by having the object play in place. Among the ways that feedback operates are time sequence choices, persistence of state, and persistence of aesthetic effect. Allowing the object to play in place can greatly assist an author to estimate cognitive load of individual word objects during the authoring process.

INTRODUCTION

Spatial hypertext is often used to create documents whose ultimate destination is some other medium. E.g. VKB [7] makes an excellent brainstorming tool, and ART [8] was explicitly designed as a spatial editing environment for creating linear texts, which would presumably be consumed using conventional methods. Elsewhere [6] I have described the Frame Stack Project: a light-weight set of classes for the programming environment Squeak [1] which provides me with an authoring environment for interactive word objects where any distinction between "authoring" and "run-time" environments is a granular state property of individual objects. In this authoring environment, word objects "play in place" without having to "execute" a separate browser or run-time player. This paper, rather than focusing on the details of that environment, reflects on the experience of using that environment over a period of about a year and a half to actually write a number of interactive poems, and contrasts this experience with earlier experiences using other kinds of environments.

A caveat: The Frame Stack Project, by design, has been aimed at creating a *personal* authoring environment. It reflects requirements of my own rather peculiar artistic practice, and reflected my inability to meet those requirements with the off-the-shelf multimedia authoring environments I had tried. This paper is even more personal — exploring my own personal experience at using this intentionally personal authoring system. To those accustomed to usability studies, the value of such results on a sample of one may seem inherently dubious. To this I can only reply that personal empowerment has always been an important goal of tool builders, and at a time when toolkits for creating interactive word objects that play in place are not common, someone has to go first and actually write in such an environment. A document may have many readers, but typically has only a small number of authors. The authoring experience is — almost by definition — an intensely personal one.

GRANULARITY

At what scale does writing begin? One can ask this question in general, or with respect to a particular session. If the question is directed to a particular session, then presumably there are some prior word objects that have been created, so the granularity will be influenced by what has been left "pending". Writers vary enormously in how they actually write. Historically, poets have had a longstanding fondness for working at least in part bottom-up: the poet may "hear" a line, which is collected as a scrap and placed in a notebook; several such scraps may then come together into a larger unit. As a poem proceeds, the granularity may shift back and forth, from extremely small-scale units: the word, or the line, to larger units, such as a stanza, and ultimately to the poem as a whole treated globally.

It is at the smallest level of granularity that an environment enabling spatial writing in place can make the most dramatic difference. As the amount of text in a word object diminishes, the "mechanism" of that object tends to loom at a larger scale with respect to the text itself. Consider a simple spatial cluster of multiple elements. Prior to using spatial writing methods, I might have indicated this in the act of composition by a kind of intermediate notation, so that a cluster with phrases 'life splay ratchet' and 'sentience tarp' inserted into the phrase 'to name blank escrow __ parole' might have

been indicated this way:

to name-blank escrow <life splay ratchet : sentience tarp> parole

Of course this intermediate notation has nothing about it which is spatial. (And certainly nothing which is even slightly interactive!) Even worse, at small-scale granularity, the "mechanism" of the intermediate notation (< ... : ... >) tends to become extremely obtrusive with respect to the words themselves.

In an intermediate step, I wrote some pieces using VKB as a prototyping environment. (VKB was not suitable as the final destination environment for any of my pieces, because it does not support the kind of behaviors my work required — specifically the frame stack behavior. See [5] for a description of this behavior.) Figure 1 shows how a prototype of this phrase might have been rendered as a "mockup" in VKB. Figure 2 shows a screen capture of the actual interactive fragment as composed "in place" in the Frame Stack Project environment.

There are several notable differences between the approaches shown in the two figures. Even though the approach shown in Figure 1 is to some degree spatial it is not fully spatial. This is because the "final" spatial relationships of the word elements in the word object being assembled are not there in the prototype. VKB does allow transparent text, and does allow objects to be placed on top of one another, but doing this can make the words illegible, and there is no user interface behavior to navigate among the layers. (It is exactly this problem — navigation to achieve legibility among objects placed on top of one another — that the frame stack concept is designed to achieve.) Of course, it goes without saying that the interactive behaviors present in the final word object are not there in the VKB prototype.

What has become apparent to me after several months of writing spatially in place in the Frame Stack Project environment — which was certainly not obvious to me when I was writing using the prototyping or intermediate notations methods above — is that these prototyping environments *induce a bias* in how I would write. Because the prototyping mechanism itself becomes so much more obtrusive at small-scale granularity, these prototyping methods tended to skew the content of what I would write to a larger scale granularity, in which there were simply more words in the leaf-nodes of the structure. I certainly did have very granular objects from time to time in my earlier writing — e.g. clusters with one word per layer — but they tended to be rare.

GROUPING

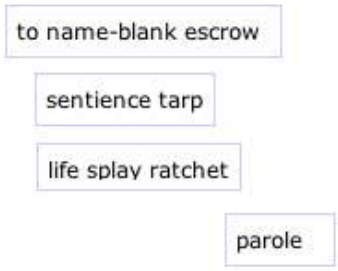

Figure 1

The cluster <sentience tarp : life splay ratchet > is embedded in the phrase to name-blank escrow __ parole, as prototyped in VKB. Much is missing from this mockup: final font, exact spatial relaltionships, and the "unity" of the parts of the phrase "to name-blank escrow __ parole". (Spatial relationships cannot be "final" because VKB makes awkward navigation of objects placed on top of one another.)

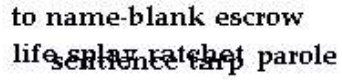

Figure 2

An actual fragment of a cluster captured from its "in-place" appearance in a work composed using the Frame Stack Project. The two-element cluster with the words 'life splay ratchet' and 'sentience tarp' is "live": unless playing is turned off, the frame stack behavior, in which individual layers become visible, will occur when the mouse approaches the cluster.

Grouping is an important function present in nearly all graphics software. Historically, spatial hypertext has had a tendency to take a different approach to grouping than your typical drawing program. Grouping in commercial graphics software tends to have the following characteristics:

- Objects are grouped interactively by muti-selection followed by the action of grouping.

- There is typically *no* distinctive user interface for a group.

When a group is selected, handles appear around the entire group, and menu selections pertaining to groups may become enabled. The appearance of these handles is the same as the appearance of handles on "atomic objects" like a line or a rectangle; when the group is not selected its "objecthood" is invisible.

By contrast, grouping in spatial hypertext sometimes relies on spatial parsing, which is "automatic" rather than interactive.

For artistic work, such as poetry, the technical issues pertaining to grouping take place against the following backdrop:

• Structural groupings may be a natural part of the work itself.

Historically, poetry has involved such units as lines and stanzas; the units of interactive electronic poetry may be different, but may still be present on a range of scales.

• Of immense artistic importance is the concept of whether a unit is *finished.*

As composition proceeds, units may exhibit a great variety in their degree of being finished. Artists frequently change their minds about whether a unit is finished. When a unit is not finished, an additional piece may be a *candidate* for inclusion in the unit. This means that at some stage in the composition process, membership of the additional piece in the unit may be uncertain or ambiguous. Of course, it exactly to be able to model such ambiguities that is one of the prime purposes for which spatial hypertext was created in the first place.

In my own practice I have found it most useful to make a group where: (1) a unit is *probably* finished; (2) I am almost certain to want to move the components together, not separately. I.e. the group is a set of spatial elements where I want the internal spatial relationships of the elements to persist. If I am thinking about adding a new element to the unit, I would tend to place it "nearby". It is important to note that as composition proceeds, a unit that is not finished may be *drastically* not finished. For instance, a phrase may have holes in which the words haven't been chosen yet. (In my own practice I tend to "write" these holes explicitly, often with an indication of what kind of prosody they should have, but without the specific words.) Of course a unit which is "drastically unfinished" may actually end up being discarded (or cannibalized for its parts!). In this situation I am personally very unlikely to want to include in a group a unit which is drastically unfinished.

I.e. the act of grouping is itself an artistic act, with artistic consequences. This has direct ramifications for spatial parsing. If an algorithm is forming groups — virtual or otherwise — on behalf of the author, that algorithm has artistic consequences. It could happen, of course, that the author of the spatial hypertext system has happily chosen an algorithm that is exactly in concert with the aesthetic of the document author, and the document author is getting just what she wants from the algorithm. But, realistically, the opposite might also happen. This suggests that if grouping is done by a spatial parser, at a minimum, tuning parameters for how that algorithm works must be accessible to document authors, and ideally the entire algorithm itself should be accessible to allow modification. Historically, spatial hypertext systems have tended not to be open source. Where algorithms have artistic consequences but the source is closed, hypertext system authors are giving document authors only the option of *turning the algorithm off*.

One possible compromise where open sourcing the entire spatial hypertext system is not an option is a form of plug-in architecture. If a spatial parser is implemented as a plug-in, with a clearly documented interface, in theory at least document authors who are unsatisfied with the off-the-shelf spatial parser could write their own. It might also be possible for just a spatial parser plug-in to be open sourced, even though the entire spatial hypertext system for one reason or another cannot be open sourced.

History vs. Structure

An interesting issue for grouping concerns the history of how the group is constructed. Consider a group made of three elements, a,b,c. Now a fourth element, d is constructed, and the document author decides that all four elements need to be grouped. There are two ways to do this: (1) the element d can be added to the group {a,b,c}. (This may require a, b, and c to be ungrouped first, and then reselected, which can sometimes be cumbersome. Or perhaps d can simply be dragged into the group. This case would require a "heavy-weight visibility" for the group boundary — which may have undesired aesthetic consequences.) (2) A new group can be created as {{a,b,c},d}. This is likely to be much faster than method 1, but can cause some surprises if the document author needs to edit the group later and forgets the history of how it was created. Of course these are two drastically different structures! The question is: does it matter?

There is a strong tendency in the hypertext community toward heavy-weight attention to structure, even to the point of placing structure at the center of the universe (Structural Computing [2].) For those from this point of view, the idea that such a drastic structural difference "may not matter" is likely to sound like heresy. But consider the case where (1) grouping is done for the convenience of the document author; (2) the purpose of grouping is simply to preserve internal spatial relationships among the elements; (3) the group as a structure is not visible to the document reader. In this situation, the document reader *cannot distinguish* the two cases above. If the two cases appear identical to the reader, then we are left with simple author convenience to determine which case should apply. It should be noted that the kind of grouping typically found in graphics software does tend to have all three of these properties.

However: some spatial hypertext systems seem to make the assumption that structure always matters. E.g. in VKB, it is difficult to group objects except by making them members of a collection. The boundaries of a collection cannot be made invisible. VKB collections are not well suited to implementing structural ambiguity.

FEEDBACK

All writers know that the writing environment influences what is written; writers tend to be very particular about their work circumstances. In the days before personal computing, that might mean a very specific kind of pen, an exact brand of notebook, a particular individual typewriter, etc. We have barely begun to explore how this process works in "new media" forms of electronic writing. As a writer works, the parts of the composition already present exert a powerful *feedback* on what gets written next. Particularly in artistic or literary writing done with a strong "bottom-up" methodology, an immensely important part of the composition process is deciding whether a possible "extension" of an element already composed *fits*. (An extremely significant strain of poetics prominent in American poetry in the latter half of the 20th century, and highly influential on a whole generation of poets, was known as *Projective Verse* [3]. A major component of this poetics may be described as composition by induction. (Robert Duncan described this when speaking about poetry using Olson's term *Composition by Field*.) The rough idea is that if a continuation occurs to the poet which works in a localized area, that continuation should be accepted, without having to match it against a "global scheme". Thus Projective Verse placed great importance on the question of whether "that next bit" — induction — fits.) When writing using either a prototyping environment or an authoring environment which is distinct from the run-time environment, the writer needs to use a kind of *extrapolation* process to imagine the result from "inside" the writing environment. To understand the implications of *having* to extrapolate — which requirement is of course absent from an environment that supports spatial writing in place — I will next explore some specific issues related to this extrapolation process.

Time Sequence Choices

Hypertext (and spatial hypertext is no exception in this regard!) tends to place interactive devices amid the words. Even if the structure used is inherently conjunctive [4], there will be some history of use in operating these devices: the user has a particular time sequence of having made various "moves" operating the interface. In the case of a cluster where the elements are meant to be peers, there will nevertheless be some element which in a particular case was visited first, and some which was visited last. When using an environment that allows writing in place, the time sequence in question as experienced by the writer simply is exactly the time sequence experienced by the writer acting as reader. Where the word objects do not play in place, however, this time sequence — along with other behaviors — must be extrapolated. How many such time sequences has the author extrapolated? In the case where elements are not reordered dynamically, the document author is likely to have a "favored order" in which elements are extrapolated. The feedback effect of "playing the objects" in a different order will not be experienced. Of course the document author can obtain this feedback by interrupting the composition process, in the case where "playback" does not occur in place, but this interruption may be disruptive. The feedback may occur, but may not be "invited".

Persistence of State

When an interactive word object is played in place, it is left in some particular state. That state is likely to be a state that persists from state that was dynamic while the object was playing. By contrast, for an authoring system where the objects do not play in place, state within the authoring system is simply not dynamic at all, but static. Persistence of state is thus another aspect that a document author must extrapolate when trying to compose with an authoring system where objects do not play in place. Consider a multiplicity where one element "drowns out the others" — an effect that the document author does not desire. This may only be apparent when the word object is played. If feedback during the composition process is used to compose further elements, this drowning out effect may not have played out. A compositional continuation may have been written based on a voice which "did not sing as loud" as the voice during object playback.

My own practice has for many years involved putting words on top of one another while allowing the layers to be read legibly — an activity to which I am deeply committed. When the resulting cluster is "closed", there will be a lot of variation in the legibility of individual words. Some will be obscured completely, while others will be completely legible. In some cases all the words in the cluster can be read easily once the reader has read them in individual layers; in other cases almost nothing can be read, but an occasional word may "stick out". This is a persistence of state: when the object is closed, it stays closed until it is activated interactively. The varying weight given to the words by persistence of state is only observed once the word object has been assembled. It could be argued that there is no reason to have the word object play in place to be able to observe this: just the graphical assemblage displays persistence. The answer, of course, is that there is a vast difference between state

that persists after the experience of a dynamic state, and a completely static state. The word that "sticks out" from a phrase you have read will have a different resonance from the same instance where the context of the entire layer has not been experienced.

It goes without saying that such persistent state effects are nearly impossible to extrapolate when a work is composed by a prototyping process where the ultimate graphical look of the word object is simply not available until "later". This is a form of extrapolation that is nearly impossible to do.

Persistence of Aesthetic Effect

When a word object is "played" (or perhaps "operated" is a better term) there is an aesthetic effect. How long does this last? How resistant is it to "noise" — in particular the noise of having to transition from an authoring environment to a run-time environment and back? If the aesthetic effect of playing a word object in a separate run-time environment is significantly dissipated in the transition back to the authoring environment, this is likely to create no harm for the ability of the author to evaluate the word object — in effect in isolation. But it may severely inhibit the ability of that word object to exert feedback on the composition process.

It is difficult to describe in words, but there is a very strong aesthetic feedback effect from graphically manipulating a word object that you have just played. Consider the case of a mostly finished cluster, and a possible candidate cluster which the document author may wish to include. These two objects can be played, dragged close to one another, then played again in a nearly seamless act where the time it takes to spatially "indicate candidacy" is within the persistence of aesthetic effect. This provides the author with a strong preview of what it will be like to combine the two objects. It is important to emphasize that such "compositional moves" *may yield failure*. From the persistence of aesthetic effect, the author might decide that the combination being considered simply doesn't work. It can save a document author considerable disruption to be able to reach such a judgment without going through the heavy-weight process of creating and then evaluating a combined object and then deciding the result was not successful and should be undone.

Software designers typically believe that their job is to create systems that are engineered for success. But in the arts, creative success only happens amid many failures that are discarded. Thus software which is designed to support artistic endeavors must also be engineered to support failure. Spatial hypertext has long been associated with such concepts as ambiguous structure, emergent (or incipient) structure, and other structural concepts where structuring is not "completely done". The result of such incompletion may go both ways: a structure may emerge, or the document author may decide that structure among a set of elements has failed, and the set is dissolved. Spatial hypertext research needs to devote as much energy to understanding and supporting the process of structure failure as it does to structure emergence.

Cognitive Load

This is a very difficult issue — the elephant at the dinner table for cybertext authors. Is the cognitive load that the word object imposes on the reader manageable? Of course a document author can conduct usability studies to try to answer this question, but a usability study is hardly feasible as part of the composition process at a granular level. As a first resort, a document author is most likely to just play the object and ask herself: is this too hard. It is quite aesthetically risky to do this by extrapolation. But, document authors will be tempted to extrapolate to avoid compositional disruption if objects do not play in place. How often will the document author click "play in browser"? Every few words? Play the whole piece every few words? Not likely. This has an intimate relationship with the subject of granularity. When the word object does not play in place, the document author's evaluation of cognitive load is likely to be made on large-scale granularities. This means that small-scale pieces that have too high a cognitive load in a context where overall cognitive load is manageable might be missed. Another effect is that evaluation of cognitive load is just less available as part of the compositional feedback process.

REFERENCES

1. Ingalls, Dan, Kaehler, Ted, Maloney, John, Wallace, Scott, and Kay, Alan. "Back to the Future: the Story of Squeak, a practical Smalltalk written in itself", *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications,* ACM, New York, 1997, pp. 318-326.

2. Nürnberg, Peter J., Legget, John J., and Schneider, Erich R., "As We Should Have Thought", *Hypertext 97*, ACM, New York, 1997, pp. 96-101.

3. Olson, Charles, "Projective Verse". *Poetry New York*, No. 3, New York, 1950. Reprinted in Allen, Donald M., *The New American Poetry,* Grove Press, New York, 1960; also in Olson, Charles, *Human Universe and Other Essays,* New York, Grove Press, 1967.

4. Rosenberg, Jim, "And And: Conjunctive Hypertext and the Structure Acteme Juncture", *Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext '01)*, ACM, New York, pp. 51-60.

5. Rosenberg, Jim, “User Interface Behaviors for Spatially Overlaid Implicit Structures” First Workshop on Spatial Hypertext, Århus, 2001, http://www.csdl.tamu.edu/~shipman/SpatialHypertext/SH1/rosenberg.pdf.

6. Rosenberg, Jim, “Hypertext in the Open Air: A Systemless Approach to Spatial Hypertext”, Third Workshop on Spatial Hypertext, Nottingham, 2003, http://www.csdl.tamu.edu/~shipman/SpatialHypertext/SH3/rosenberg.pdf.

7. Shipman, Frank M. III, Hsieh, Haowei, Maloor, Preetam, and Moore, J. Michael, “The Visual Knowledge Builder: A Second Generation Spatial Hypertext”, *Hypertext '01: Proceedings of the 2001 ACM Conference on Hypertext*, ACM, New York, 2001, pp. 113-122.

8. Yamamoto, Yasuhiro, Nakakoji, Kumiyo, and Aoki, Atsushi, “Spatial Hypertext for Linear-Information Authoring: Interaction Design and System Development Based on the ART Design Principle”, *Thirteenth ACM Conference on Hypertext and Hypermedia (Hypertext '02),* ACM, New York, pp. 35-44.