R.D. #1 Box 236
Grindstone, PA 15442
E-mail: jr@amanue.com
Visible Language 30.2
Jim Rosenberg, 102-117

The Interactive Diagram Sentence: **Hypertext as a Medium of Thought**

Consideration of my work in poetry over more than twenty-five years begins with an analysis of the difficulties of juxtaposition for the poet. A diagram syntax notation provides a method for juxtapositions to be included in larger structures; the accessibility of structural elements in a diagram allows for such constructions as internal relationships and feedback loops. Juxtaposition itself, with no sacrifice of intelligibility, is achieved through an interactive device called a simultaneity. Finally the interactive diagram sentence is explored as a vehicle for hypertext as a medium of thought: this is a truly "native" mode of entirely non-linear thought.

Jim Rosenberg

Diagrams: a separate channel for syntax

To begin with the elemental, the "structural zero," juxtaposition: the act of simply putting an element on top of another, with no other structural relation between the two elements except that they are brought together, is the most basic structural act, the most fundamental micromaneuver at the heart of all abstraction. But consider the problem of the poet in bringing this about. When a sound is played simultaneously with another sound, the result is a sound. When a painter places a bit of colored space on top of another bit of colored space, the result is a bit of colored space. A mathematician would say that the domains of the composer or visual artist are *closed* with respect to the operation of juxtaposition: the result of juxtaposing two elements from the domain is another element from the domain. But what happens when we juxtapose words? Whether it is done by means of sound — either via simultaneous readings by multiple performers, or by overlaying magnetic or digital media — or visually, the result of juxtaposing words — in the almost palpable physical sense of putting them directly on top of one another — is likely to be sheer *unintelligibility:* one will be lucky to make out any of the words at all. How is the poet to achieve juxtaposition with no sacrifice of intelligibility?

But it gets worse: how can direct juxtapositions of words be *used* in larger structures? It is not hard to work in modes that give up such structures as syntax. One simply does without. Asyntactic poetry is a large and fruitful domain in which to work. On the other hand, giving up all possibility of structure is giving up a great deal indeed. Syntax is at the heart of how we normally structure words. How does one achieve such structuring and yet still have complete freedom to use juxtaposition wherever it is artistically important? How does one designate the *structural role* of a juxtaposition in a larger structure? One could put this question a bit more crudely by asking: What is the part of speech of a juxtaposition? The composer John Cage once criticized the

twelve-tone system as having no zero.[1] One could say that syntax "has no zero": in a sentence every element has its structural role with respect to the syntax diagram, or parse tree; there is no way to have words in a sentence whose syntactical relationship to one another is the *null relationship:* nothing at all except that they are brought together. How can the poet have her cake and eat it too? How can one keep both syntactical null relationships and much more elaborate relationships, in which juxtapositions act as elements?

These are some of the formal problems that have motivated my work going back more than twenty-five years. A method for approaching the second problem — how to incorporate null structures as structural elements — became apparent long before I realized how juxtaposition could actually be implemented. By devising an explicit visual structural vocabulary — separating syntax out into its own channel, so to speak — structural roles could simply be directly indicated. The elements occupying those roles might be words or word clusters or other structural complexes. Thus began a long series of works called *Diagram Poems*.

1 See for instance Cage, John. 1961. "45' For A Speaker." *Silence.* Cambridge, Massachusetts: The MIT Press.

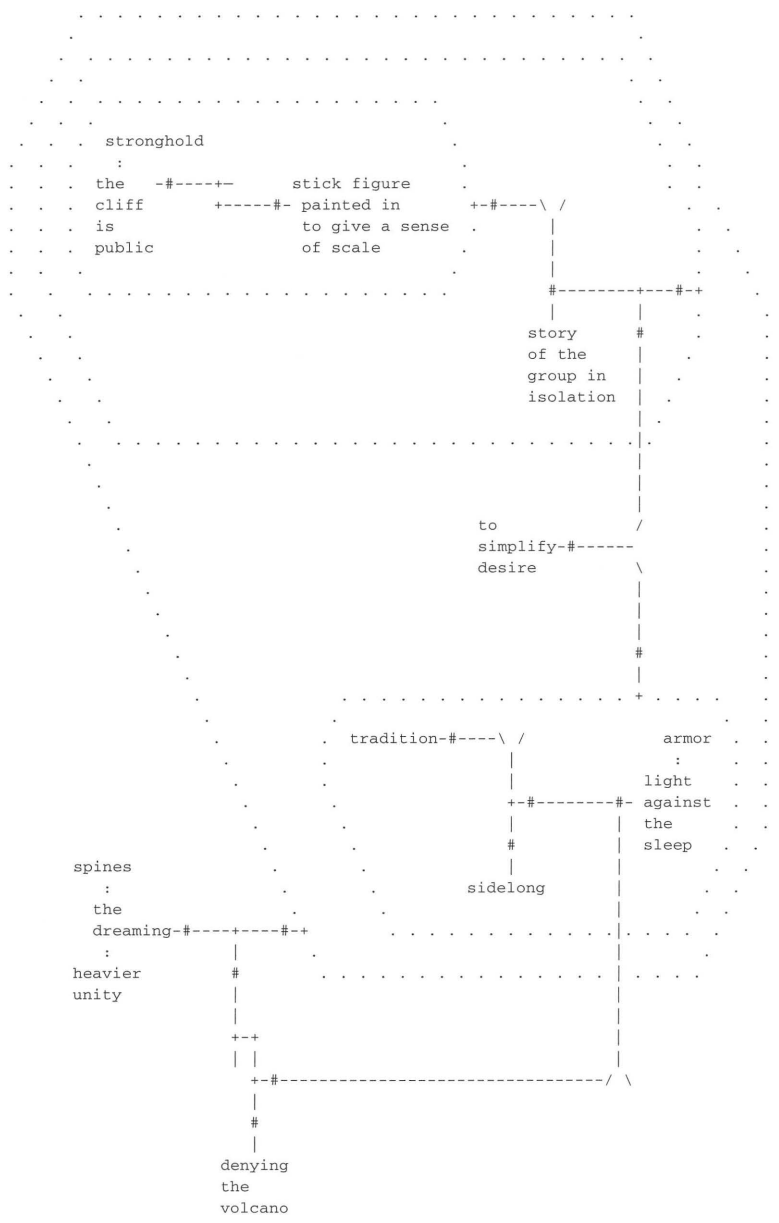


Figure 1: A diagram poem from *Diagrams Series 3*

Figure 1 shows a poem from *Diagrams Series 3*.[2] It illustrates many of the facilities provided by the diagram notation in a variety of works spanning a large number of years. The configuration:

```
x -#----+--
        +-----#-y
```

shows a simple modifier relationship where x is modified by y. The configurations:

```
x -#\  /-#-y          x-#---\  /
     |                       |
     |                       |
     |                       |
     #                       #
     |                       |
     z                       y
```

show verb relationships; in the left case above, z acts as the verb relating x and y, in the right case above, y acts as the verb and x acts as the subject.

These relationships can be built up into complexes in two ways: where a "node" in a relationship is a loop of dots, the element participating at that node is the entire contents of the loop; where a node terminates in the graphical part of a relationship, the element at that node is *the act of making* that relationship. A number of interesting things happen when syntax is "externalized" in this way. Syntax came about originally in conjunction with speech, where speaker and listener are constrained by: 1) the requirement that the listener "decode" the message approximately synchronized in real time with the speaker; and 2) the aid of only whatever "temporary storage" the listener has available in short-term memory. One might say that the function of syntax is to pre-code the message with *storage cues* so that the listener will know how to park pieces of the message in short-term memory so that they can be properly assembled in the logical relationships desired by the speaker — all in more or less real time without getting behind the speaker.

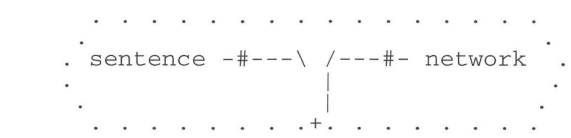
2 Rosenberg, Jim. 1979. *Diagrams Series 3*. Grindstone, Pennsylvania. Excerpts appeared in *Interstate 14*, 1981.

Writing, however, changes the picture completely. Obviously, the real-time constraints are absent: the reader may take as much time as desired, may revisit parts of the message as many times as are necessary and may even browse the message "out of order." In addition, a written document may be said to *provide its own storage*. In contrast to speech, where whatever parts of the message not properly stored in short-term memory by the listener are simply (and irretrievably) *gone*, the written message *persists*: it stores itself, it stores its structure, it stores its own logical relationships.

Secondly, by externalizing syntax, all points and substructures in the message are *accessible* in ways not normally found in speech. That they are accessible to the reader has already been discussed. Some interesting ways they are accessible to the writer are revealed by figure 1. Note the relationship of the phrase "story of the group in isolation" to a larger whole in which it appears. In an externalized graphical syntax, such a relationship is easy to simply *draw*; joining a part with a larger whole in which it participates is as easy as joining a part with a disjoint part. Relationships between a part and a larger whole in which the part occurs are an obvious logical structure that occurs commonly in the world; yet this is difficult to do in conventional syntax. In addition, the fact that relationships may simply be drawn to parts of the message already laid out allows for complex multiple pathways to be established within even small messages; the message may *feed back upon itself*. Feedback, while a ubiquitous structure in nature, is notoriously difficult to deal with. It violates the principle set theorists call "well-foundedness"; it may induce the potential for infinite loops in computer programs. Where feedback is introduced into the way sound elements are combined in an electronic synthesizer the results may be completely unpredictable: all bets are off. Figure 1 also illustrates this concept of feedback inside the sentence: the "highest-level" logical relationship shown in figure 1 relates the configuration at the very bottom, in which "denying the volcano" is a modifier, with a cluster "already" deep within the message: "armor : light against the sleep."

A feedback loop may seem an inimical structure to a programmer, where the threat of an infinite loop is ever present (and indeed the infinite loop stands out as an archetype "cardinal bug" second only in its fearsomeness to an out-and-out crash); one may say that the threat of an infinite loop stands as the fear at the heart of all programming. (Technically, the theorem that one cannot algorithmically determine whether a general computer program will lead to an infinite loop is known as the halting problem and establishes absolute limits on what is computable.) Yet, when the composer induces feedback into synthesized sound structures, the ear can hear it as a single sound; when a graphical feedback loop is established in a visual syntax, the mind can apprehend *the loop as a whole* as a single gestalt. Of course to do so, *time must not be constrained.* It is difficult to see how an aural syntax, subject to real-time constraints, could accommodate feedback loops.

A diagram syntax is notably non-linear. While this is an important point, one must be careful to avoid going too far in pushing non-linearity as a distinction between a diagram syntax and the conventional speech syntax. The essence of syntax is its ability to convey logical relationships across a distance of intervening words; one might say syntax has been our way out of the bind of achieving complex speech structures in the face of the constraint of linear time. Conventional syntax provides a start toward obtaining full non-linearity from an inherently linear channel; a diagram syntax can break free completely to non-linearity without restraint. Non-linearity is freed to extend far down into the fine structure of language — just barely above the word. Or, to put it slightly differently:



The interactive juxtaposition

But how to actually achieve juxtaposition of words — to place them literally on top of one another — and sacrifice nothing in the way of intelligibility? Too often we think of words simply as whatever comes out of a word *processor* — or perhaps one should call it a word constrainer, forcing as it does the words into the familiar linear chains (with a nod to non-linearity by allowing hypertext links) and certainly *not* allowing words to be one atop another! A graphics program, on the other hand, allows text objects to be placed on top of one another with complete graphical freedom, but the legibility problem remains. Yet the graphics program gives a clue: juxtaposition combined with intelligibility is achieved (at last) by using interactive software. In a construction I call a *simultaneity*, words are placed in the same location — with all the freedom and fluidity a graphics program allows. At first it appears the words are simply overlaying one another — with no solution at all to the problem of overlay plus legibility. In this state the simultaneity may be called *closed*. The act of *opening* the simultaneity consists of moving the cursor using the mouse to a particular "hot spot" on the screen. When the cursor enters this hot spot, all layers of the simultaneity but one are hidden: the one visible layer can be read unimpeded by its partners in the juxtaposition.

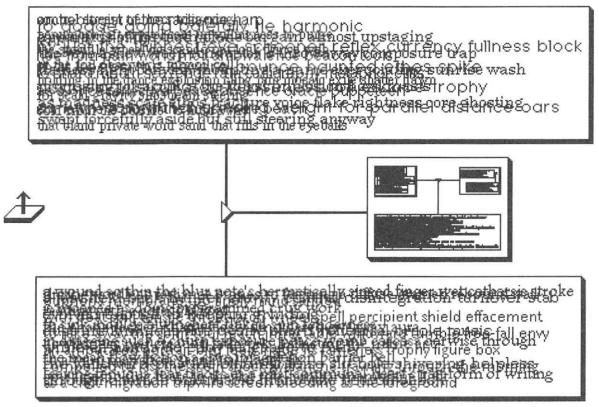


Figure 2a.

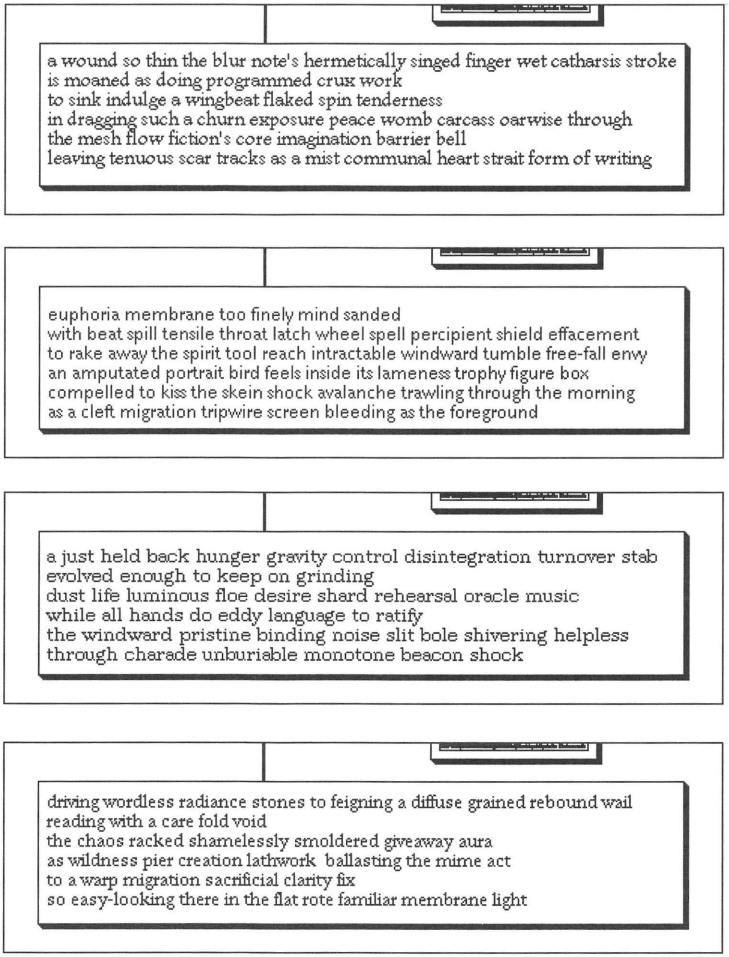


Figure 2b-2e.

Figure 2 shows a simultaneity from *Intergrams*.[3] In 2a the simultaneity is closed and all layers are visible; in the detail views 2b-2e the simultaneity is opened showing each layer. (A static illustration cannot convey the *tactile* aspects of causing the different elements to appear by moving the mouse with one's hand; the reader will have to try to imagine this.)

3 Rosenberg, Jim. 1993. *Intergrams*. Watertown, Massachusetts: Eastgate Systems.

Taking the diagram interactive : **hypertext as a medium of thought**

A diagram is a marvellous instrument for presenting information of great complexity in a small space — to the point that the phrase "Well, you'll have to draw me a diagram" is a stereotype epithet of complaint that something is too complex. There are limitations to diagrams, however. What happens when the space required is not small? How does one manage a diagram comprising *thousands* of elements? Enter hypertext.[4]

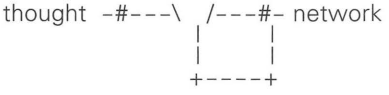
Hypertext is most often thought of as a special kind of computer software — or as the documents produced using that software, but here I would like to consider the idea of hypertext as virtual diagram. In the classical model of hypertext, a document is structured as a network of *nodes* and *links*. The nodes are typically either entire documents, or document regions (known as anchors); a link is a relationship between document places such that clicking on the anchor at the source end automatically takes the user to the destination anchor. If a hypertext is small enough and simple enough, the entire network can be represented by other means than using a computer — on paper, for instance.

Often hypertext begins (alas) at the level of the document; such documents are fully linear and use completely traditional methods for structuring text internally. Using links, associations are built up among places in these documents. The notation of the diagram poems suggests a different possibility: hypertext built up from scratch using very fine-grained word elements, where hypertext is used to carry the infrastructures of language itself, e.g., syntax. One may speak here of *hypertext as medium of thought:* rather than hypertext serving as an association structure for thoughts that are not themselves hypertexts, an individual thought itself is "entirely" hypertext. To use terminology familiar to computer programmers, hypertext becomes a medium in which one thinks "natively."

4 The term "hypertext" was originally coined by Ted Nelson. The literature on hypertext is extensive; for a bibliography see Harpold, Terence, "Hypertext and Hypermedia: A Selected Bibliography," in Berk, Emily and Joseph Devlin, editors. 1991. *The Hypertext / Hypermedia Handbook.* New York: McGraw-Hill. The best single-source introduction to hypertext is probably still Nelson, Theodore H. 1981. *Literary Machines.* Swarthmore, Pennsylvania: T.H. Nelson.

Why should we do this: construct a morphemic hypertext[5] — hypertext taken into the fine structure of language? Why not make do with the syntax we have? Why not leave hypertext structure to relate "conventional" documents, at the level known in the hypertext literature as the lexia?[6]

To answer this question, let me pose a counter-question: How does a single mind apprehend a complex network? It is becoming more and more clear that not only are networks — in the actual physical sense — becoming more and more important in our lives, the network as a metaphor is becoming increasingly important in dealing with a wide range of aspects of living. What does it mean for thought when an individual thought is itself a network? Does it help in understanding the complexities of life's networks around us, containing us, moving us, to "think native" in a mode that is inherently network? Many seek in art a *refuge* from complexity; indeed, many consider simplicity as such a paramount goal for art that it virtually defines artistic purpose. For others, complexity is taken as a given in this life, and art is seen as an aid that can help us *to live with it* rather than fight it or withdraw from it. To understand the network one *becomes* the network. Thought itself is a network, there is no other-than-network:



5 The term "morphemic hypertext" was applied to my work by the hypertext researcher Catherine C. Marshall (private correspondence).

6 The term "lexia" was borrowed from the writings of Barthes by George Landow to refer to a document piece at a hypertext node; see Landow, G. P. 1992. *Hypertext: The Convergence of Contemporary Critical Theory and Technology.* Baltimore, Maryland: Johns Hopkins University Press.

The obstacles in the way of achieving such a hypertext of thought are many. Among them are:

1) Lack of Tools. Most commercially available hypertext systems are not adequate. Although much attention has been paid in the hypertext research community to a variety of structural models other than the standard "node-link" hypertext model,[7] this has borne very little fruit in tools available for the kinds of computers writers are likely to have accessible. Instead, commercially available hypertext software tends to either adhere too rigidly to a node-link model or require the user to build everything "by hand." Typical hypertext structures are or-based, i.e., disjunctive: from lexia L with links X, Y, and Z one may choose X *or* Y *or* Z. Syntax structures are *and-based*, i.e., conjunctive: a sentence with parts X and Y and Z consists of X *and* Y *and* Z.[8] (Consider the classical phrase structure rule:
S -> NP + VP

A sentence can be rewritten as a noun phrase followed by a verb phrase. One does not get to *choose* which of NP and VP to use; they are both there.) This is not to argue against the use of disjunctive structure, or "classical" hypertext links. Rather, the need is for both to be available as an author requires. Typically, commercially available software has no built-in support for conjunctive abstractions at all.

Another problem with available software packages is too rigid an attitude toward *behavior*. Available hypertext systems typically offer only off-the-shelf behaviors that can't be extended by the user. At the other extreme, systems like Hypercard are fully programmable, but don't allow that programmability to be encapsulated in pluggable objects. (For instance, a Hypercard button has no storage containers!)

7 See for instance Marshall, Catherine C., Frank G. Halasz, Russell A. Rogers and William C. Janssen, Jr.. "Aquanet: a hypertext tool to hold your knowledge in place." *Proceedings of Hypertext '91* for a model based on relations; Parunak, H. Van Dyke. "Don't Link Me In: Set Based Hypermedia for Taxonomic Reason- ing." *Proceedings of Hypertext '91* for a model based on sets; and Stotts, P. David, and Richard Furuta. "Petri-net based hypertext: Document structure with browsing semantics." *ACM Trans.Off. Inf.Syst.*, 7:1, for a model based on Petri nets.

8 The concept of conjunctive hypertext was introduced in Rosenberg, Jim. "Navigating Nowhere / Hypertext Infrawhere." *SIGLINK Newsletter* 3:3, http://www.well.com/user/jer/NNHI.html.

2) Reticence to tackle "language itself." There is no gainsaying that the idea of using hypertext to carry the infrastructure of language itself is an extremely radical proposition — one from which many will shrink. One source of objection is the idea that "language itself" is off-limits by virtue of being biologically hard-wired.[9] There are two answers to this: the artistic answer and the engineering answer. For the artistic answer, consider the analogy of dance. No one would dispute that there is a biological basis for how our bodies are put together, for the conformation of bone structure, for the ways that joints work: in short biology places many constraints on how the human body can move. This has not notably abolished the dance. To the contrary: one may say it has *created* the dance: we admire those who can show us what the boundaries are for how the human body can move, who can take us all the way up to those boundaries and perhaps even stretch them. To the degree that syntax is biological, it makes experimentation on the limits of syntactic structure *more* valuable rather than less. For the engineering answer, consider the analogy of computer networks. Again: there is no disputing that neurons are biological objects, and that genetics has a great deal to do with how neurons function individually and how the nervous system functions collectively. This does not diminish the utility or importance of those "externalized nervous systems" we call computer networks. The proposal for hypertext as a medium of thought, for hypertext inside the infrastructure of language, is a proposal for an "externalization" of syntax analogous to the externalization of the nervous system manifested in computer networks.[10] Just as computer networks do not "replace" the biological nervous system, an externalized mechanism of thought does not "replace" syntax; rather it adds to syntax and allows new possibilities.

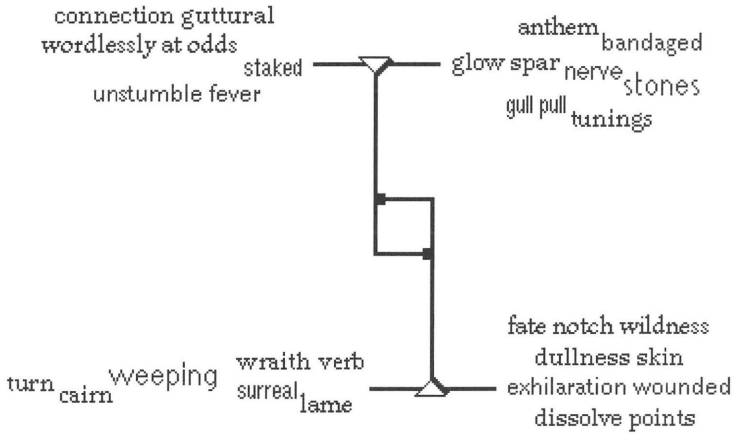
For instance: how do we allow more than one user "inside the sentence"? For a diagram syntax this is almost trivially easy: each user's relationships can be distinctively marked — using color, for instance, or any other form of explicit marking. How is it possible using conventional syntax to construct a "multi-user sentence"? It is exactly in joining

9 For a review of issues pertaining to the biological basis of language see Pinker, Steven. 1994. *The Language Instinct.* New York: William Morrow and Company.

10 Externalization of language is discussed extensively in Donald, Merlin. 1991. *Origins of the Modern Mind.* Cambridge, Massachusetts: Harvard University Press.

multiple users that our biological nervous systems break down and externalized ones show their true value. How does one construct a true multi-user medium of thought? To repeat: a multi-user medium of thought does not mean a multi-user mechanism for bringing together "single-user thoughts" but rather a medium where *the individual thought* can be a multi-user construction. Just as multi-user interactions require an externalization of the nervous system, a true multi-user medium of thought will require an externalization of syntax.

It all interacts:[11]



11 The final figure is a single plane in a simultaneity from Rosenberg, Jim. (In press.) *Diffractions through: Thirst weep ransack (frailty) veer tide elegy.* Watertown, Massa- chusetts: Eastgate Systems.