

Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext

Catherine C. Marshall and Frank M. Shipman III

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

phone: (415) 812 - 4740

fax: (415) 812 - 4777

e-mail: {marshall, shipman}@parc.xerox.com

ABSTRACT

Hypertexts may be implicitly structured, based on either node content or context. In this paper, we examine implicit structures that rely on the interpretation of node's spatial context. Hypertext authors and readers can perceive and understand these idiosyncratic structures, but, because they are implicit, they cannot be used by the system to support users' activities. We have explored spatially structured hypertext authored in three different systems, and have developed heuristic recognition algorithms based on the results of our analyses of the kinds of structures that people build. Our results indicate that (1) recognition of implicit structures in spatial hypertext is feasible, (2) interaction will be important in guiding such recognition, and (3) the hypertext system can provide layout facilities that will render later systematic interpretation much easier. Found structures can be used as a basis for supporting information management, as a straightforward way of promoting knowledge-base evolution, as a way of solving representational problems endemic to many hypertext systems, or as a basis for collaboration or interaction.

Keywords: implicit structure, spatial hypertext, link automation

INTRODUCTION

In *Literary Machines*, Ted Nelson described hypertext as "non-sequential forms of writing involving links." [Nels84] Augment's links were "textual citations to some specific file item." [Enge84] Later, in his widely-cited survey article, Conklin claimed links to be "the essential feature of hypertext." [Conk87]

Since then, many assumptions about hypertext have relied on this groundwork, expanded over time to include a broader notion of what linking is. Halasz's recently reconsidered definition of hypermedia referred to nodes as being "explicitly or implicitly organized into one or more structures." [Hala91] Thus, links and networks have given way to a newer, more general, notion of structures that may sometimes be implicit. But how are implicit hypertext structures realized?

In this paper, we examine the role of structures in hypertext and some ways and reasons that

hypertexts become implicitly structured. We explore the spatial arrangement of text as an alternative means for creating hypertext, and look at some examples authored in existing hypertext systems to give us insight into the kinds of hypertextual structures we might find in space. Based on these examples, we have developed heuristic algorithms for interpreting spatial hypertext structures. We briefly describe these algorithms and suggest ways they might be applied to find the missing links.

The role of links in hypertext

Hypertext links serve dual functions. Most commonly, they are a mechanism that implements the rhetorical structures of a hypertext. In their most literal form, links are the vehicle for traveling from one segment of text to another, possibly along an already blazed path (as in Scripted Documents [Zell89], Guided Tours [Trig88], or Vannevar Bush's original Memex design [Bush45]), or according to the reader's own interests or internalized construction of what should come next (as in Intermedia [Yank88], Storyspace [Joyc91], or Nelson's envisionment of Xanadu [Nels84]). Because most hypertext systems are geared for the presentation of content, traversal and its rhetorical entailments are the most fully explored aspect of links (see for example [Moul92]); most hypertext authoring tools and hypertexts rely on a traversal-oriented implementation of links.

Links also serve a second crucial function: they are a representational form. They can be used to articulate specific semantics for interconnection. Typed links and nodes in IDE [Russ88], gIBIS [Conk88], and MacWeb [Nana91], relations in Aquanet [Mars91], and typed nodes and links contextualized by activity spaces in Sepia [Stre92] are all good examples. These links make structure explicit, and are thus useful in writing, argumentation, problem structuring, or capturing semi-formal knowledge representations of a domain.

In this paper, we focus on structure not as a means of traversal, but rather as the rhetorical and semantic basis for hypertext.

Where does structure lie? Can it be inferred from the content of nodes or documents, as navigation tools such as Superbook [Egan91] suggest, or is it defined by context -- how the nodes are organized and used, either implicitly or explicitly? At one extreme, content-based structures can be computed on the fly, and are never explicitly present in the hypertext. At the other extreme, in systems that treat structures as first-class objects, explicit connections can be created independent of the nodes that they connect [Gron92].

How is structure constrained? Is it ad hoc, is it constrained by authors, or is it constrained by the system according to a cognitive model of the activity the system was designed to support [Thur91]? We find examples along a continuum. Unconstrained hypertexts are the norm, but embedded constraints -- whether they are defined by authors of the system or authors of the hypertext -- can help effect coherence and consistency. In a few cases like HyperSet's taxonomic sets [Paru91], type constraints define the structure; hypertext structures are determined on the basis of type.

Finally, how is structure conveyed? Does it exist only in the reader's head? Is it generated by the system in the form of a structure map [Bern91], or is it recorded in a spatial layout by the author?

Once again, we can find great variation: systems where structure is never made concrete to the reader, systems where structure is conveyed graphically to the reader, but cannot be directly modified, and systems where graphical forms can be manipulated by the author to modify underlying structures (as in NoteCards browsers [Hala87]). In some cases, the layout is the structure; for example, NoteCards tabletops [Trig88] constrain their constituent objects (in this case, nodes appearing in windows) to be arranged on the screen according to a user's specifications.

In this paper, we examine implicit structures that do not rely on system interpretation of content, but rather on interpretation of spatial context. The structures that we are concerned with are freewheeling, although they are often subject to authors' visual conventions or self-imposed layout constraints. These structures are conveyed graphically, in a manner often orthogonal to a hypertextual framework.

Implicit, extensional hypertext

DeRose introduced a distinction between intensional links and extensional links [DeRo89]. Intensional links follow from the internal structure and content of the nodes; extensional links are idiosyncratically defined. We have also seen a distinction between implicit and explicit links. DeRose applies this distinction to intensional links, but not to extensional links.

Within these dimensions, many hypertexts involve explicit, extensional links; connections are individually defined. Autolinking strategies may yield implicit intensional links (such as in Perseus [Mylo90]). Other forms of virtual linking, such as the Analyst's rule-based linking program Assistant [Anal89], IDE's templates [Jord89], and McCall's virtual structures in PHIDIAS [McCa90], are more in the spirit of explicit intensional links; explicitly defined kinds of connections are computed. The hypertexts we consider use implicit extensional links -- idiosyncratic connections intended by the author, perceived by the reader, but not necessarily stored, nor recognized by the system. Because the connections are not defined explicitly, but inferred by computational analysis of a particular hypertext, these structures extend and sharpen DeRose's distinctions.

LINKS RECONSIDERED

We again emphasize that besides being a traversal mechanism to guide a reader through linearly disconnected pieces of text, links are also expressive. Links can capture the semantics of interconnection or explicit rhetorical intent. More critically, links make it possible to explore the role of text in different contexts, to use "any unit of text as a new element in an expanding vocabulary of signs." [Bolt91]

For this purpose, we can conceive of hypertext as using contextualizing forms other than links. Nelson introduced the idea of "stretch text" [Nels84] as one way of presenting text in context, but in this paper, we will look at mechanisms for implicit structure in another way. Spatialized text, explored in some depth by Bolter in his ECHT '92 keynote address [Bolt92], has already demonstrated its ubiquity in many of today's hypertext systems (as well as other presentational forms [Tuft90]).

In our own experiences with Aquanet, a hypertext system that allows authors to create graphical/

textual objects in a shared information space, we found spatialized text to be a primary mode of creating hypertext; people created spatialized text in preference to using Aquanet's relational model for semantically and graphically expressing interconnections [Mars92].

How does hypertext become implicitly structured?

If hypertext can be thought of in terms of textual objects arranged in space, where a textual object can recur by appearing more than once in the same space, hypertext can have structure without conventional links. As noted in [Bern92], recurrence is neither a defect, nor a vice, but rather a method of allowing multiple encounters with the same textual content.

In spatialized text, a reader/author can perceive the intended structures in space, just by noticing their geometrical relationships ("if this text is close to that text, then the two must be related"), visual characteristics ("if this text is in a twelve-point italicized serif font, it must fill an annotative role similar to the other twelve-point italicized text that's an annotation"), and recurrence ("this segment of text means something completely different when I read it over here"). If this structure is perceived by heuristic algorithms, it can then fill the same function as explicitly represented structure.

Why does hypertext become implicitly structured?

In practice, people experience difficulty trying to articulate why they've linked two things (or even deciding if what they're trying to express is a node or a link) [Ship93]. But they are accustomed to arranging text in space, both in computational media and, if somewhat less facily, in the physical world. Computational tools like text editors, structured graphics editors, outliners, hypertext systems, and even multiple window displays (where different documents or different portions of the same document can be juxtaposed) all support spatialized text. Sometimes links are noted graphically in these media -- lines are drawn from one piece of text to another, or boxes group several textual items -- but often interconnections are left implicit; ambiguities in interpretation are a desired consequence of not defining links explicitly. Many current hypertext linking mechanisms have no simple way of expressing ambiguity or lack of certainty -- "this node may be part of this structure, or it may be part of its neighbor." But such uncertainty can be conveyed using spatial relationships such as distance and collocation, as it is in some automatic content-based clustering algorithms.

LINKLESS STRUCTURE IN THREE HYPERTEXTS

To explore how people have spatialized text in hypertext systems -- sometimes in creative ways at cross-purposes to developers' hopes -- we gathered data about the textual objects appearing in layouts created in three different systems, NoteCards, the Virtual Notebook System (VNS), and Aquanet. Eight samples were selected, each the result of a long-term information management or analysis task.

We encoded the data in a uniform representation to capture spatial and visual aspects of the text objects; relative planar location and extent of each object was recorded. Each object was also assigned a type based on distinguishing graphical characteristics. For example, to encode the NoteCards sketches, we assigned one of three types to each textual object according to whether its font was regular, bold, or italic, and calculated its approximate extent according to number of

lines and maximum line length. VNS objects were encoded according to representational type, font, and conversations with the author about their intent. Aquanet objects required no additional interpretation to fit into our data analysis framework.

Sketches in NoteCards: spatialized text in preference to hypertext

NoteCards [Hala87] provides a very general node-link model for creating hypertext networks. It supports hierarchical filing of nodes, and a customizable system-generated graphical overview of the hypertext network. Yet there were NoteCards users who made little use of the system's network traversal model, and instead, either used its link icons as a way of spatializing text (by embedding them, for example, in a sketch to form a flat hyperspace, with all the references coming from the same place), or gave up on the hypertextual aspect altogether, and used NoteCards sketch cards to manipulate bits of text in space (so NoteCards served as a hierarchical file system for storing sketches).

The NoteCards samples we collected resulted from an anthropological data analysis task that took place over slightly less than a month. Since the author confined himself to text and a few graphical notations like circles and arrows, the three examples pertaining to this application provided us with what can literally be called spatialized text; all structure (including the application of font styles, alignment, drawn annotations, and organization of text items) was created ad hoc by the author and never recorded as hypertext.

Figure 1 shows a schematized version of one of his NoteCards sketches; text has been replaced by boxes that cover the same area as the text. Shading indicates whether the text was bold, italic, or plain. Graphical notations have been preserved.

This example contains 35 individual pieces of spatialized text. One recurrence -- the two boxes with the heavy borders -- is the only text object-to-text object link. It is noted graphically as well, by a two-headed arrow. A second graphical link connects two of the higher-level structures. The clusters of objects are both apparent from layout density (we see 5 distinct areas), and they are noted graphically by drawn ellipses (as well as a rectangle and lasso).

If we consider layout constraints like alignment, we can spot several lists or sets. The group of objects in the box in the lower right corner is very probably a named set or labelled list, since the aligned elements all appear to be of the same type. If we consider the visual typing, we can see that the top three encircled groupings are probably a type of composite -- the layout pattern "a crosshatched box above some number of gray boxes above a striped box" is repeated in all three. If we examine the lassoed objects, we might infer that text replaced by diagonally striped boxes refers to the text it is next to (shown as gray boxes).

So in our first example, structures using layout constraints, reference, and type constraints are all created outside the hypertextual framework provided by NoteCards.

A VNS page: structure orthogonal to the hypertext

VNS [Ship89] is a page-oriented hypertext system that implements the electronic analog of a shared lab notebook. VNS pages support the display and manipulation of a variety of types of information objects including text, images, video, audio, page links, and action links.

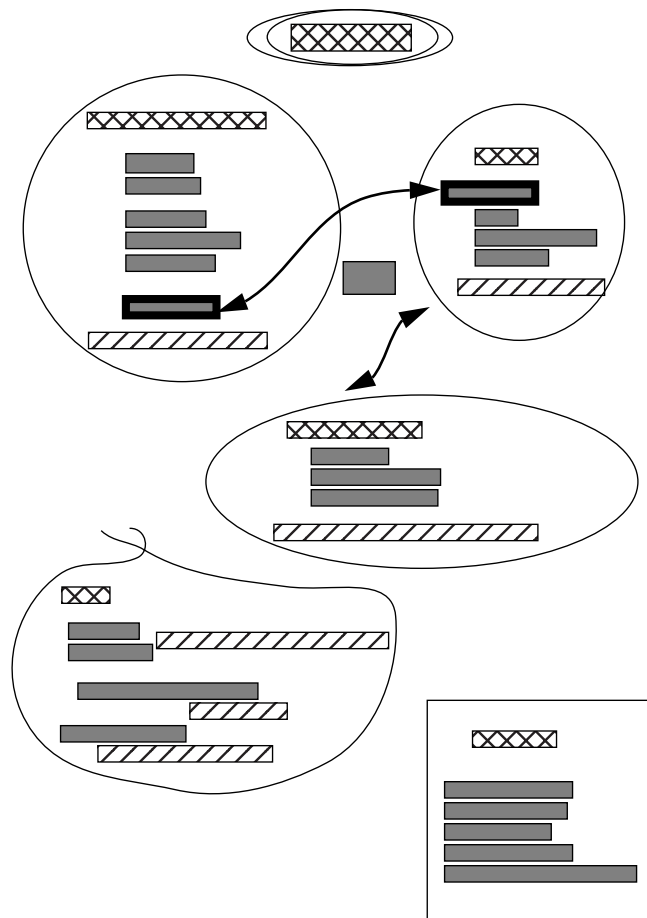


Figure 1. Spatialized text in NoteCards.

In our VNS example, 7 or 8 collaborators used the system's hypermedia capabilities to track system bugs; Figure 2 shows an encoded version of the page. The page contains 28 textual/iconic link markers showing links to other VNS pages, 10 image objects that implement a tabular layout, column headings that are a composite of numbers and text, and miscellaneous blocks of text and link markers (near the top of the schematized version shown in Figure 2). A total of 63 objects were arranged on the page when we collected this data, after six months of use. We could perceive six distinct types.

In this example, we see how the multiple authors have spatialized text to create recognizable structures orthogonal to the hypertext. All of the boxes in Figure 2 represent some kind of text, whether it's within a link marker, or just a textual object. The link markers are mostly in the columns (although some appear in a separate list at the top of the table). Because the authors have agreed upon a conventional structure -- a table -- the space retains mutual intelligibility to the many authors. Some graphical notation is used to delineate the rows and columns.

So in our second example, reference is still done hypertextually, but structures are composed through spatial layout and graphical notation.

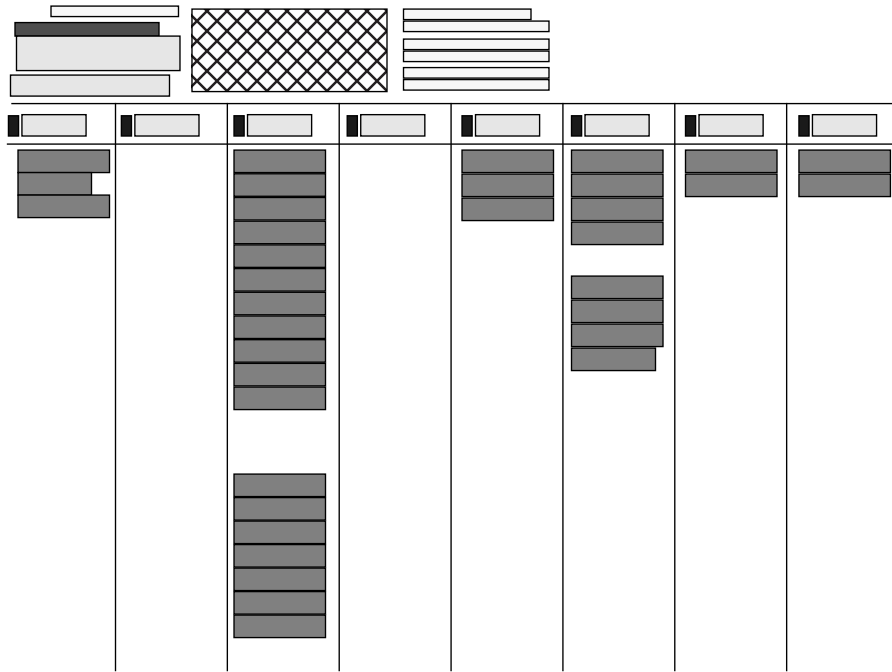


Figure 2. Spatialized text in the Virtual Notebook System.

Aquanet discussions: the space is the hypertext

Aquanet is a collaborative hypertext tool that combines elements of frame-based knowledge representation and graphical presentation. Users manipulate instances of typed information objects in a shared space. Like many other spatially-oriented systems, Aquanet's information space can contain multiple references to the same object.

We originally looked at four different Aquanet discussions (although subsequently we have run the structural analysis algorithms we describe on other Aquanet layouts). The most complex was the analysis of recent activities in the field of machine translation, an application we describe in detail in [Mars92]. Others include a bug tracking discussion very much analogous in function to the VNS example we described earlier, a collaborative look at Spanish-English machine translation software (a portion is shown in its encoded form in Figure 3), and a discussion we used to organize topics for a paper. Subsequent applications that we evaluated involved the same types of structures that we found in our eight examples.

Two of the four Aquanet examples had the unique property of having grown large enough to have distinct structural regions and many complex higher-level structures (structures formed of other structures). Figure 3 shows one of the simple cases of this. Note that there are 16 roughly similar composite structures. These 16 composites can also be seen as a single set, a higher-level structure.

Aquanet does not support "drawing on the space," so there are no graphical hints about how the space is organized. Because there's no alignment or gridding, the structures take on a very informal appearance (in contrast with our other examples of spatialized text). Our more complex Aquanet examples use recurrence to provide reading context for the objects.

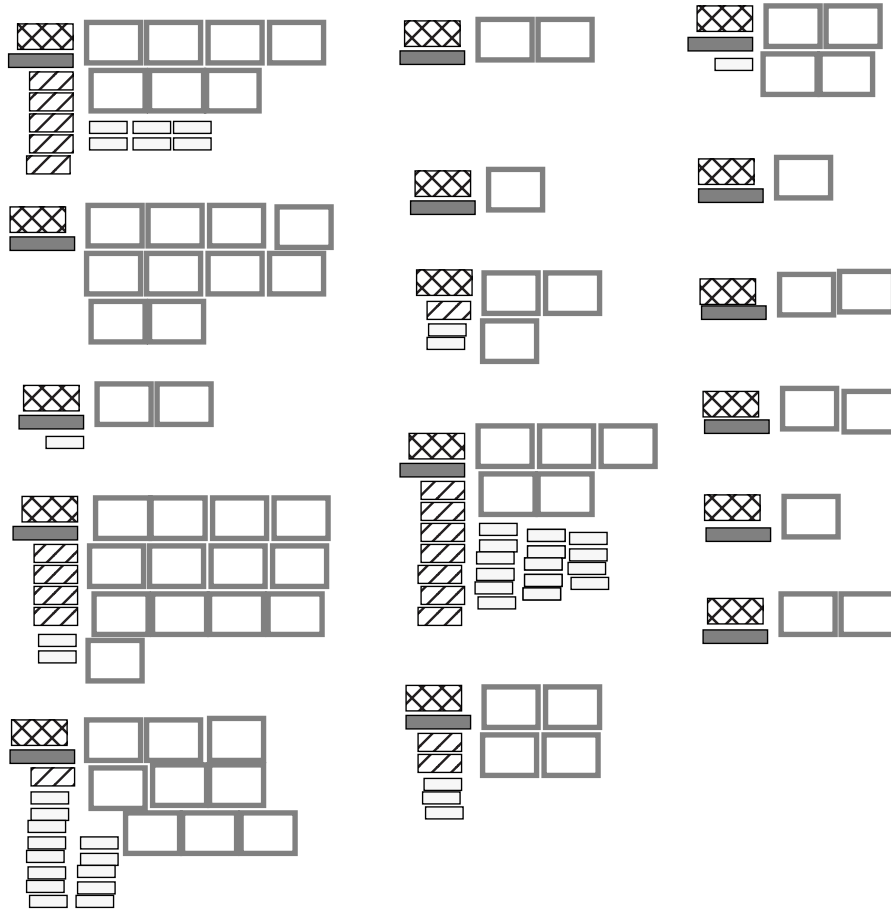


Figure 3. Spatialized text in Aquanet.

So in our third example of spatialized text, typing and reference are done within the framework Aquanet provides; constraints on spatial layout are enforced informally, by the user.

PERCEIVING HYPERTEXT

We can readily see structure in our examples, both very fine-grained local structure, and some larger higher-level structures. These examples suggest that we can perceive specific types of structures based on visual properties and spatial layout.

Spatial hypertext structures like aggregates can be perceived strictly on the basis of proximity. Other kinds of hypertext structure, such as ABC's lists [Smit91], can be perceived without links by relying more on conventions of spatial orderliness. So some hypertext structures can be identified strictly on the basis of how they are laid out.

Spatialized, HyperSet's taxonomic sets [Paru91] might be perceived through proximity, but they could be identified visually, based on apparent homogeneity and heterogeneity of objects. More general kinds of hypertextual structures like composites rely on the distinguishability of multiple types, in addition to uniform relative proximity.

Finally, perception of referential structure can come from graphical cues such as enclosing or

connecting arcs and arrows. The lack of such graphical cues does not necessarily mean there is no referential structure, but that any such structure must be discovered by considering larger patterns of proximity and typing.

Layout-based hypertext structures

Often hypertext systems that use text analysis or other information retrieval techniques present the results of their analyses in a plane as collocated aggregates (see for example [Lelu92]). Not surprisingly, we have found that people also use this visualization technique of placing textual objects close together to imply structure to themselves and other users.

We define aggregates to be textual objects near each other or piled on top of each other, items that share a spatial context. Figure 4 shows two examples of proximity-based structures, abstracted from our Aquanet examples. In practice, proximity analysis of all of our sample spatial text arrangements extracted useful kinds of structure.

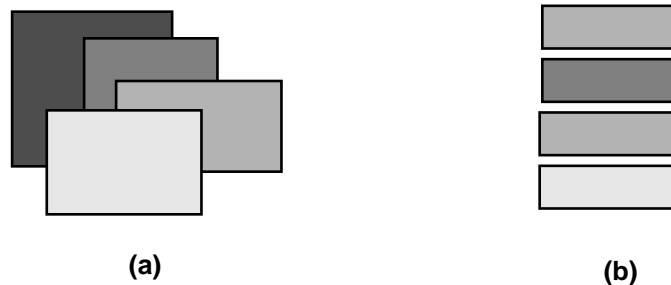


Figure 4. Proximity-based structures: (a) an aggregate, and (b) a list.

Identification of proximity-based structures can also take advantage of different layout characteristics. For example, if all the textual objects are aligned vertically or horizontally, then we might have a list. If they are aligned both vertically and horizontally, we might have a table (of the sort demonstrated in [Mars87]). We located meaningful, intentionally aligned spatial structures in all eight of our samples.

Because all of these textual objects are treated equally under proximity analysis, they can be created without any preliminary distinctions, not even between links and nodes; all explicit elements have become equal.

Structures that use layout and type

Other available visual properties can also prove to be useful. In all of our examples, we can readily pick out differences in the way objects look -- in Aquanet, these differences may be based on graphical appearance (which in this case has a one-to-one correspondence with type), in VNS, on font, implementational type (images, link icons, text blocks all look different), or functional roles (a numeric label, for example, has a distinctive appearance), and in our NoteCards samples, on font properties. So, when we're looking at the spatialized text, we can ask ourselves, "Is this object the same kind of object as that object?" Distinctions of this sort allow us to establish homogeneity and heterogeneity of objects.

These distinctions, used in conjunction with layout characteristics (proximity and alignment), can

form the basis of the systematic detection of structures like taxonomic sets (sets where all the members belong to the same category). Figure 5a shows an example of a taxonomic set identified based on homogeneity and proximity; of course, proximity is inessential to identifying taxonomic sets if there is a well-developed notion of type.

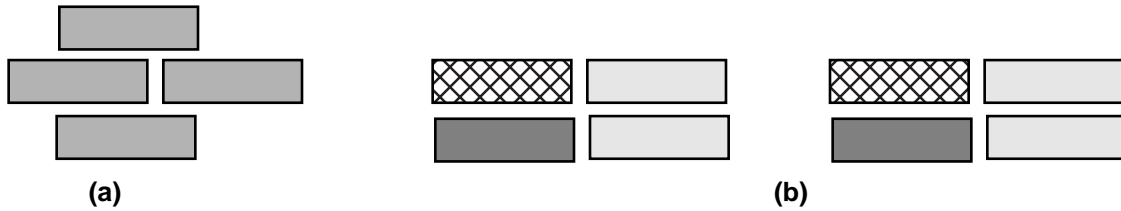


Figure 5. Layout and type-based structures: (a) a taxonomic set and (b) two instances of a composite.

Considering relative spatial positions in conjunction with type allows us to readily identify composites. By Halasz's definition, composites are higher level hypertext constructs that include (rather than reference) other hypertextual objects [Hala88]. Gronbaek and Trigg advance this definition by proposing structured composites [Gron92]. In our spatial framework, we take this to mean that composites can be defined with both layout constraints and slot constraints; in other words, an instance of a particular type occupies a known position in the composite structure. Figure 5b shows two instances of a particular type of composite; the patterns of type and layout are the same in each.

Why is it so important to recognize composite objects? Finding composites can be an important step in solving the problem of premature commitment to structure; they make certain kinds of decisions ("Is this a property of an object, or a separate thing?") less far-reaching. Surprisingly, all of our sample layouts had at least some examples of implicit composites.

Reference-based structures: graphical notations and signified recurrence

Graphical composability provides us with the ability to form referential structures in space. One driving idea behind Aquanet networks is that chaining (two relations sharing a single object) supports visually interlinked structures. As we have noted in [Mars92], explicit visually interlinked structures were the exception rather than the rule.

Our examples show us two ways to perceive reference-based structures: signified recurrence, (e.g. highlighting the use of a visual reference in more than one place in a structure), and informal graphically noted connections (e.g. fences that corral groups of objects and lines to signify standard hypertextual reference). When working within a limited domain, a specialized graphical vocabulary, like that provided by XNetwork's palette of computer network devices, can support the construction of more shades of referential meaning [Ship93a]. An example is that text placed near a particular design unit or network connection may imply a reference that other designers will understand.

We also find that recurrence and graphically noted structures are useful adjuncts to spatialized text, since the informal arrangements are as volatile as they are expressive. Collaboration and self-annotation are difficult when so much of the structure is expressed implicitly, in the layout; just moving an object around changes the meaning of the reference, and the other objects that

refer to it. Drawing on the space (and interpreting drawn notations) gives authors a way to extend and stabilize the meaning of volatile layouts.

AN IMPLEMENTATION THAT FINDS SPATIAL STRUCTURES

Our data analysis supports the idea that automatic perception of implicit spatial structure is possible, and that it can identify several useful kinds of hypertext constructs. We built a prototype recognizer to test some ways of parsing spatialized text into hypertext structures. Unlike Lakin's visual language parsing in vmacs [Laki87], our algorithms did not assume that we could unambiguously recover their underlying syntactic structure. Unlike Pictoral Janus, a visual programming environment that bases connectivity on assessments of inside, connected, or touching [Kahn92], our purpose is not to "debug" formal visual/spatial structures, but rather to tease out some hypertext features from structured, spatialized text.

Why recognize implicit structure?

People can easily perceive that there is structure in each of the three examples we discuss in the previous section. Why is there a need to develop heuristic algorithms to find it?

First, the found structures can be used as the basis for supporting simple but repetitive information management tasks; they can help authors interact with ad hoc organization. For example, if objects are identified as implicit members of a taxonomic set, an operation on one may cause the system to ask the author if the change -- possibly reorganization or a change in internal structure -- should be propagated to the others.

Second, allowing structures to emerge gradually, then be recognized, reduces some of the representational problems introduced by node-link models. Most node-link models of hypertext introduce a variety of authoring requirements -- for example, text segmentation, deciding whether an object is a node or link, deciding whether a characteristic of an object should be part of the object itself (as a property or slot) or whether it warrants the creation of a separate object that is linked to the first, and other impediments to the task at hand.

Heuristic structure recognizers can solve many of these problems. Spatialized text allows the articulable content to be expressed as objects, and the more difficult to express connections between objects to be identified later, thus ameliorating the problem of premature structuring. It also allows properties that are expressed as separate objects to later be folded back into objects in the form of the recognized composites described in our discussion of structures that use layout and type; the distinction between internal and external structure is usefully fused.

Third, structures that are perceptibly regular or can be identified by simple heuristics can provide an important means for communicating about how a space is used; they can promote mutual intelligibility [Trig86]. Collaborators can use found structures as a basis for explaining implicit organization. For example, if the heuristic algorithms find clustered objects, it may be vital that the author of the cluster explain to her collaborators why the objects have been pushed together. Found structure can thus be the basis for interaction between humans, not just human-computer interaction.

Finally, recognizing structure is an important method for helping users notice and express the

regular structure of their domain and maintain its consistency. This identification is especially important if a more formal knowledge base is a desired end of the task or is of use in supporting the task. Recognized structures can be used to support the incremental formalization of domain information and thus support knowledge-base evolution [Ship93a].

How structure is analyzed

To keep the spatial structure analysis algorithms system-independent, they use only the visual and planar characteristics described earlier: object position, extent, and a very simple notion of type. The heuristic algorithms, implemented as communicating experts, perform a bottom-up parse. Found structures are re-parsed according to the same rules to identify higher level structures (like sets of sets, or sets of like composites).

As we pointed out earlier, while these structures are not explicit, they are extensional; they have meaning to authors and readers even though these meanings may be fraught with ambiguity. Thus, parses are only partial, and it's no surprise that they are sometimes not well aligned with human intent.

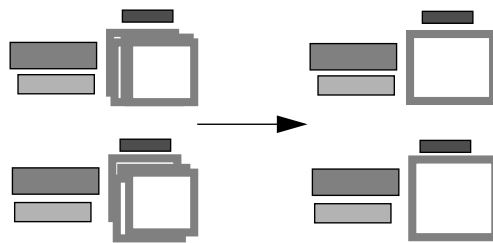
We experimented with recognition order, and found that different spatial layouts have different preferred parsing orders. Thus, interaction should guide how these heuristic algorithms are applied -- in what order and when. In general, the parses were fairly successful when aggregates based on overlap were found first, sets based on homogeneity and alignment (possibly vertical or horizontal) were located next, and composites were located after that. Identification of heterogeneous structures (piles of dissimilar objects and composites) generated a new type; structures formed from homogeneous objects assumed the type of their constituents. By repeated application of the recognition algorithms using previously inferred structure, higher levels of structure are perceived.

Figure 6 simulates a parse that the recognition algorithms can perform; this example is a segment of a structure built in Aquanet. For the purposes of the figure, when a parsed structure inherits a type from its constituents, it is represented the same way; if a new type has been generated, an arbitrary new visual representation is shown.

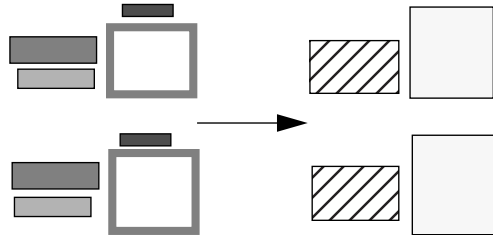
First, the piles of similar overlapping objects are parsed into a single structure that has the same type as its constituents. Then the two composite substructures, one on the right, and one on the left, are parsed as new types. The two composite substructures are joined into one. Finally, the set of two complex composites is parsed, resulting in a single structure.

Using our examples of spatialized text as test cases, we found that this kind of parsing was fairly successful. The recognition algorithms were not tuned to any particular style of layout, since they ran independently of the hypertext systems that produced the spatialized text. Specific structures that were identified included aggregates, aligned taxonomic sets, and composites. The structure identified was often consistent with the authors' intent, although the higher the level of the perceived structure the greater the occurrence of incomplete recognition.

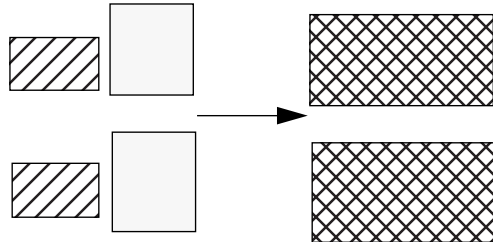
Figure 7 shows a sample parse of an Aquanet discussion. Different structures found by the recognition algorithms are shown with different shading. The right-most column of objects is one



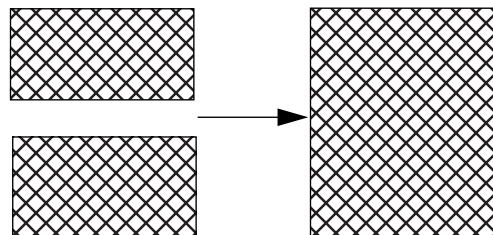
Step 1: Overlapping homogenous objects are grouped, retaining the same type.



Step 2: Two different composites are located. New types are assigned.



Step 3: Higher-level composites are formed. Once again a new type is assigned.



Step 4: A set of two similar composites is located. The type is retained.

Figure 6. Parsing a structure in spatialized text.

of many ambiguous situations in this relatively simple spatial layout. In this case the layout could be interpreted as a list of items with a larger-than-normal gap or, as the recognition algorithms decided here, there could be two separate shorter lists. Neither alternative is necessarily the correct one; the spatial freedom allows the author to leave such determinations implicit, and left up to the interpretation of the reader.

From this automatic heuristic parsing experiment we can conclude that (1) such parsing is feasible; (2) interaction will be important in guiding this kind of recognition (especially since the structures can be both idiosyncratic and ambiguous); and (3) that the hypertext system can provide facilities for spatialization that will render later systematic interpretation much easier.

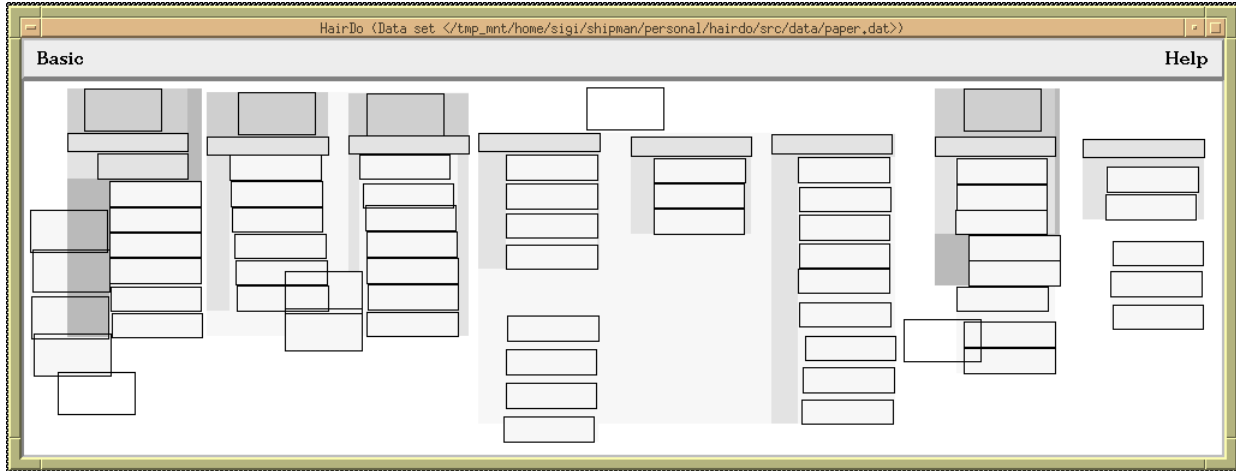


Figure 7. The heuristic algorithms parse a structure created in Aquanet.

An experiment with planar cut-up

To further test our recognition algorithms, we performed a variant of cut-up [Burr73] on our original spatial structures. Would the recognition algorithms find structure that was not readily available to human perception?

We distributed the objects in each layout randomly, such that the entire layout occupied the same planar extent as did the original; also, the type and individual extent of each object was preserved. In some of the new layouts, random placement of objects conformed to a grid (i.e. the objects appeared to be aligned); in others, objects were placed freely on the limited plane.

When the recognition algorithms were applied to these randomly generated layouts the number and complexity of the structures that were identified was significantly reduced compared to the original layouts. Most of the structures that were found involved a small number of objects, and were likely to be two member aggregates or sets. No spurious composites were identified, and no higher-level structures (structures composed of other structures) were parsed. Only one of the machine-generated spatializations -- the one whose human-created counterpart was least amenable to this type of parsing -- appeared roughly equivalent to its original counterpart. Thus, disorganized structures that are not readily perceived by human interpreters were not mistakenly perceived by the algorithms.

CONCLUSIONS AND DIRECTIONS

Spatialized text can usefully augment the staccato traversal suggested by a node-link model of hypertext. It allows authors to create volatile, implicit extensional hypertext, and it allows readers to interpret intertextual relationships according to perceptual conventions.

But spatialized text can also make us recall why the node-link model became normative: It enables us to refer from one textual entity to another; it supports comprehensible, navigable structuring of a complex space; and it prescribes and constrains interconnection where it is useful.

The ability to find and use implicit structures can support authors and readers in a system that

extends the node-link model with spatial hypertext. As we noted in our discussion of reasons to recognize implicit structure, this kind of found structure can be used as a basis for supporting information management, as a straightforward way of promoting knowledge-base evolution, as a way of solving representational problems endemic to many hypertext systems, or as a basis for collaboration or interaction.

We have also found that spatial hypertext is necessarily idiosyncratic and ambiguous; people use implicit structures of this sort to maintain the fluidity of how textual content is organized and interpreted. Thus any sort of heuristic recognition should be guided by human interaction.

Where should this work lead? Our heuristic algorithms are just a start; additional structure finders can be developed, including algorithms that can interpret graphical as well as spatial notations. Drawings on the space can thus have hypertextual meaning.

In situ investigations also need to be performed. Currently, this work analyzes pre-existing structure; integration with a spatial hypertext system would require more attention to interactive, incremental recognition of structure. It may not be sufficient to retrofit current hypertext systems with these capabilities; a more promising approach involves designing for spatial hypertext. More extensive use of spatial hypertext will allow us to observe how this work scales; our largest analyzed space so far contains 2000 nodes; it is interesting to contemplate how these techniques may be applied to a much larger information space.

REFERENCES

- [Anal89] Analyst User Guide Volume II. Pasadena, CA: Xerox Special Information Systems (1989).
- [Bern91] Bernstein, M., Bolter, J.D., Joyce, M., and Mylonas, E. "Architectures for Volatile Hypertext." Proceedings of Hypertext '91, San Antonio, TX (Dec.16-18, 1991), pp. 243-260.
- [Bern92] Bernstein, M., Joyce, M., and Levine, D. "Contours of Constructive Hypertext." Proceedings of ECHT '92, Milano, Italy (Dec. 1-4, 1992), pp. 161-170.
- [Bolt91] Bolter, J.D. Writing Space: The Computer, Hypertext, and the History of Writing. Lawrence Erlbaum Associates, Hillsdale, NJ (1991).
- [Bolt92] Bolter, J.D. "Writing on the World: Virtual Reality and the Future of Hypertext." ECHT '92 Keynote Address, Milano, Italy (Dec. 4, 1992).
- [Burr73] Burroughs, W.S. Exterminator! Penguin Books, New York (1973).
- [Bush45] Bush, V. "As We May Think." Atlantic Monthly, (Aug. 1945), pp. 101-108.
- [Conk87] Conklin, J. "Hypertext: An Introduction and Survey." IEEE Computer 20, 9 (Sept. 1987), pp. 17-41.
- [Conk88] Conklin, J. and Begeman, M.L. "gIBIS: A Hypertext Tool for Exploratory Policy

- Discussion." MCC Technical Report Number STP-082-88, Austin, TX (1988).
- [DeRo89] DeRose, S.J. "Expanding the Notion of Links." Proceedings of Hypertext '89, Pittsburgh, PA (Nov. 5-8, 1989), pp. 249-258.
- [Egan91] Egan, D., Lesk, M., Ketchum, R.D., Lochbaum, C.C., Remde, J.R., Littman, M., and Landauer, T. "Hypertext For the Electronic Library? CORE Sample Results." Proceedings of Hypertext '91, San Antonio, TX (Dec. 16-18, 1991), pp. 299-312.
- [Enge84] Engelbart, D.C. "Authorship Provisions in Augment." Proceedings of 28th IEEE International Conference, San Francisco, CA (Feb. 27- Mar. 1, 1984), pp. 465-472.
- [Gron92] Gronbaek, K. and Trigg, R. "Design Issues for a Dexter-Based Hypermedia System." Proceedings of ECHT '92, Milano, Italy (Dec. 1-4, 1992), pp. 191-200.
- [Hala87] Halasz, F.G., Moran, T.P., and Trigg, R.H. "NoteCards in a Nutshell." Proceedings of the ACM CHI + GI Conference, Toronto, Ontario (Apr. 5-9, 1987), pp. 45-52.
- [Hala88] Halasz, F. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems." Communications of the ACM 31, 7 (July 1988), pp. 836-852.
- [Hala91] Halasz, F. G. "'Seven Issues': Revisited." Hypertext '91 Keynote Address, San Antonio, TX (Dec. 18, 1991).
- [Jord89] Jordan, D.S., Russell, D.M., Jensen, A.-M.S., and Rogers, R.A. "Facilitating the Development of Representations in Hypertext with IDE." Proceedings of Hypertext '89, Pittsburgh, PA (Nov. 5-8, 1989), pp. 93-104.
- [Joyc91] Joyce, M. "Storyspace as a Hypertext System for Writers and Readers of Varying Ability." Proceedings of Hypertext '91, San Antonio, TX (Dec. 16-18, 1991), pp. 381-388.
- [Kahn92] Kahn, K. "Concurrent constraint programs to parse and animate pictures of concurrent constraint programs." Proceedings of the International Conference on Fifth Generation Computer Systems, June 1992.
- [Laki87] Lakin, F. "Visual Grammars for Visual Languages." Proceedings of AAAI 87, Seattle, WA (July, 1987), pp. 683-688.
- [Lelu92] Lelu, A. and Francois, C. "Hypertext Paradigm in the Field of Information Retrieval: A Neural Approach." Proceedings of ECHT '92, Milano, Italy (Dec. 1-4, 1992), pp. 112-121.
- [Mars87] Marshall, C.C. "Exploring Representation Problems using Hypertext." Proceedings of Hypertext '87, Chapel Hill, NC (Nov. 13-15, 1987), pp. 253-268.
- [Mars91] Marshall, C.C., Halasz, F.G., Rogers, R.A., and Janssen, W.C. Jr. "Aquanet: a hypertext tool to hold your knowledge in place." Proceedings of Hypertext '91, San

- Antonio, TX (Dec. 16-18, 1991), pp. 261-275.
- [Mars92] Marshall, C.C. and Rogers, R.A. "Two Years before the Mist: Experiences with Aquanet." Proceedings of ECHT `92, Milano, Italy (Dec. 1-4, 1992), pp. 53-62.
- [McCa90] McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F., and Wallace, N. "PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext." Proceedings of ECHT `90, Paris, France (Nov. 1990), pp. 152-165.
- [Moul92] Moulthrop, S. "Toward a Rhetoric of Informating Texts." Proceedings of ECHT `92, Milano, Italy (Dec. 1-4, 1992), pp. 171-180.
- [Mylo90] Mylonas, E. and Heath, S. "Hypertext from the Data Point of View: Paths and Links in the Perseus Project." In Hypertext: Concepts, Systems and Applications, N. Streitz, A. Rizk, and J. Andre', Eds. Cambridge University Press, Cambridge, UK (1990), pp. 324-336.
- [Nana91] Nanard J. and Nanard M. "Using Structured Types to Incorporate Knowledge in Hypertext." Proceedings of Hypertext `91, San Antonio, TX (Dec. 16-18, 1991), pp. 329-342.
- [Nels84] Nelson, T. Literary Machines. Sixth Edition, 1984.
- [Paru91] Parunak, H.V. "Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning." Proceedings of Hypertext `91, San Antonio, TX (Dec. 16-18, 1991), pp. 233-242.
- [Russ88] Russell, D.M., Moran, T.P., and Jordan, D.S. "The Instructional Design Environment." In Intelligent Tutoring Systems: Lessons Learned, J. Psotka, L.D. Massey, and S.A. Mutter, Eds. Lawrence Erlbaum Associates, Hillsdale, N.J. (1988).
- [Ship89] Shipman, F.M., Chaney, R.J., and Gorry, G.A. "Distributed Hypertext for Collaborative Research: The Virtual Notebook System." Proceedings of Hypertext `89, Pittsburgh, PA (Nov. 5-8, 1989), pp. 129-135.
- [Ship93] Shipman, F.M. and Marshall, C.C. "Formality Considered Harmful: Experiences, Emerging Themes, and Directions." Technical Report CU-CS-648-93, Department of Computer Science, Univ. of Colorado, Boulder, CO (1992).
- [Ship93a] Shipman, F.M. Supporting Knowledge-Base Evolution with Incremental Formalization. Ph.D. Dissertation, Univ. of Colorado, Department of Computer Science, Boulder, CO (July, 1993).
- [Smit91] Smith, J.B., and Smith F.D. "ABC: A Hypermedia System for Artifact-Based Collaboration." Proceedings of Hypertext `91, San Antonio, TX (Dec. 16-18, 1991), pp. 179-192.
- [Stre92] Streitz, N., Haake, J., Hannemann, J, Lemke, A., Schuler, W., Scheutt, H., and Thuring, M. "SEPIA: a Cooperative Hypermedia Authoring Environment."

- Proceedings of ECHT `92, Milano, Italy (Dec. 1-4, 1992), pp. 11-22.
- [Thur91] Thuring, M., Haake, J., and Hannemann J. "What's Eliza Doing in the Chinese Room? Incoherent Hyperdocuments and How to Avoid Them." Proceedings of Hypertext `91, San Antonio, TX (Dec. 16-18, 1991), pp. 161-178.
- [Trig86] R.H. Trigg, L.A. Suchman, and F.G. Halasz. "Supporting Collaboration in NoteCards." Proceedings of CSCW `86, Austin, TX (December 3-5, 1986), pp. 152-162.
- [Trig88] Trigg, R. "Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment." Proceedings of CSCW `88, Portland, OR (Sept. 26-28, 1988), pp. 216-226.
- [Tuft90] Tufte, E. *Envisioning Information*. Graphics Press, Cheshire, Connecticut (1990).
- [Yank88] Yankelovich, N., Haan, B., Meyrowitz, and Drucker, S. "Intermedia: The concept and construction of a seamless information environment." *IEEE Computer* 21,1 (Jan. 1988), pp. 81-96.
- [Zell89] Zellweger, P.T. "Scripted Documents: A Hypermedia Path Mechanism." Proceedings of Hypertext `89, Pittsburgh, PA (Nov. 5-8, 1989), pp. 1-14.

ACKNOWLEDGMENTS

We would like to thank Tom Moran, Frank Halasz, and Jim Coombs for many helpful discussions about visual and spatial structures and Randy Trigg for invaluable discussions about links. We would also like to thank Fred Lakin for suggesting spatial cut-up and Andy Burger of the ForeFront Group for providing us with the VNS data.