

Toward an Organic Hypertext

Robert Kendall

Writing Program
New School University
66 West 12th Street
New York, NY 10011, USA
E-mail: kendall@wordcircuits.com

Jean-Hugues Réty

Laboratoire de Recherche en Informatique
Université de Paris Sud
91405 Orsay Cedex, France
E-mail: jh.rety@iut.univ-paris8.fr

ABSTRACT

The Connection System is an adaptive hypermedia system for hypertext poetry and fiction. Its adaptive features can help maintain the large-scale structural integrity of the text that emerges during a reading, no matter what local navigational choices the reader makes. Authors can define structural components and specify adaptive behaviors for the textual and navigational elements within them. By establishing criteria for displaying links or text fragments conditionally, authors can encapsulate their understanding of structural possibilities to better guide the formation of the emergent structure without reducing the reader's agency or freedom of interaction. The system models the reader's knowledge of textual components and uses this model to guide adaptive behavior and give the reader a better sense of how structural elements are unfolding. We consider the problems involved with modeling the knowledge of a literary text, and we offer specific examples of how adaptivity can give the reader more control over the reading and make it more satisfying.

KEYWORDS: Hypertext literature, adaptive hypermedia, organic form, theory, hypertext structure

INTRODUCTION

How does form arise in hypertext poetry and fiction? The structure is constrained by the textual and navigational elements the author has created, yet the final configuration of the text emerges only as the reader explores it. Clearly both author and reader have a hand in the structure that is ultimately experienced by the reader.

Many hypertext theorists argue that the reader takes on some of the authorial responsibility. Landow characterizes the hypertext reader as a "reader-author." Citing reader-response theory, he maintains that this role is merely an extension of the normal processes of interpretation that a

reader applies to any text, whether electronic or printed. "This construction of an evanescent entity or wholeness always occurs in reading, but in reading hypertext it takes the additional form of constructing, however provisionally, one's own text out of fragments, out of separate lexias." (p. 195) [19] Bolter makes a similar case and goes on to say that "in the electronic writing space all texts are like dramas or musical scores. The reader performs the text," interpreting or improvising as necessary. (p. 158) [2]

Under most circumstances, the "authorial" activities engaged in by the hypertext reader bear little similarity to the writing processes that actually bring the words into existence. The reader doesn't create something out of nothing. Instead, the reader's role resembles that of an author organizing a final version from preexisting rough drafts and notes. Even this analogy is incomplete in one important respect, though. Generally an author creating the final version of a text depends upon an overall knowledge of the material to make sound structural decisions.

The performers to which Bolter refers must also know their materials before they can interpret or improvise on them. Similarly, when readers create their own versions of the text in the sense understood by reader-response theory, they are organizing and interpreting memories of text that they have already read. In contrast to these situations, when readers of hypertext make interactive choices that affect the overall textual structure, they are dealing with material they have not read and may have little or no knowledge of.

A central question concerning hypertext form then arises. How can we compensate for the reader's limited knowledge of the materials when she constructs her own text out of fragments? We propose a three-pronged approach to the problem. 1) Allow the reader to conceptualize unread material as the potential development or growth of material that has already been read. Making it clearer how unknown text relates to known text parlays the reader's knowledge further. 2) Maximize the impact of what the reader knows about the text by letting the system gauge this knowledge and use it to adapt the emerging text structure in accordance with it. 3) Let the author encapsulate elements of the creative (decision-making) process as behaviors that can be attached to specific components of the text. The reader can then tap into

a surrogate for the author's knowledge when manipulating these structural components.

The goal is to let the reader's knowledge and the author's knowledge better complement each other in the making of structural decisions. Only the author has a sufficient grasp of overall textual relationships to understand the long-range implications of different structural possibilities as they emerge during the reading. Only the reader knows the immediate needs of structural integrity, since they change from moment to moment during the reading as some possibilities are realized and others are not. We apply the techniques of *adaptive hypermedia* (AH) to gauge the reader's knowledge of the hypertext and then allow author-defined algorithms to adapt the text and navigational elements in response to this changing knowledge.

Explicitly encapsulating structural potential as behaviors of content makes it easier for the reader to explore the many structural possibilities that are inherent in the textual materials themselves. She will be better able to do without the traditional linear restraints placed upon a text and also less dependent upon the arbitrary structures that can emerge from navigational choices made with little knowledge of their long-term consequences. This idea of structure growing from content rather than being imposed rigidly or arbitrarily from without has been important in literary criticism since the 19th century. Coleridge described "organic form" that "shapes as it develops itself from within" rather than being simply pressed onto its contents like an inflexible mold. [5] Levertov characterizes organic form in poetry as "the concept that there is a form in all things (and in our experience) which the poet can discover and reveal." [20]

In writing a hypertext, an author is faced with the challenge of creating opportunities for structural growth that are artistically sound as a finished structure. Structural integrity and unity depend upon the collective unfolding of many different unpredictable elements. In considering how we can unify the growth of diverse hypertextual parts, it is useful to turn to another idea of literary organicism. Levertov stresses the goal of relating the needs of individual elements "to the needs of the surrounding parts and so to the whole," [20] echoing Coleridge's description of organic form as "the connection of parts to a whole, so that each part is at once end and means." [5] The AH approach we have implemented in the Connection System lets the parts adapt themselves to better serve the whole as the reader guides their growth. We call this *organic hypertext*.

The Connection System is an AH system designed specifically for Web-based poetry and fiction. Most AH systems are intended for educational materials or online help and are therefore not ideal for literary work. (For a survey of adaptive hypermedia applications, see [3].) Most Web-based adaptive systems (such as InterBook [4], AHA [7], and AVANTI [9]) rely on CGI or some other server-side technology, which also poses a serious problem for hypertext literature. Many if not most hypertext poets and fiction writers have no access to a server that supports CGI or server

extensions. Many literary publishing sites also lack this support. A practical delivery system for literature must forgo server-side resources, so the Connection System relies exclusively upon JavaScript.

The system code is contained in an external library that is accessed by JavaScript function calls inserted in the HTML documents. The function calls can be added manually to documents or by means of the Connection Toolkit, an authoring facility that can either be run from a browser or installed as a set of extensions for Macromedia Dreamweaver. The Toolkit lets authors specify adaptive elements and parameters in a dialog box and then automatically generates the appropriate JavaScript code. Since the system scripts don't interfere with existing HTML, authors can use the system in tandem with any Web authoring tools.

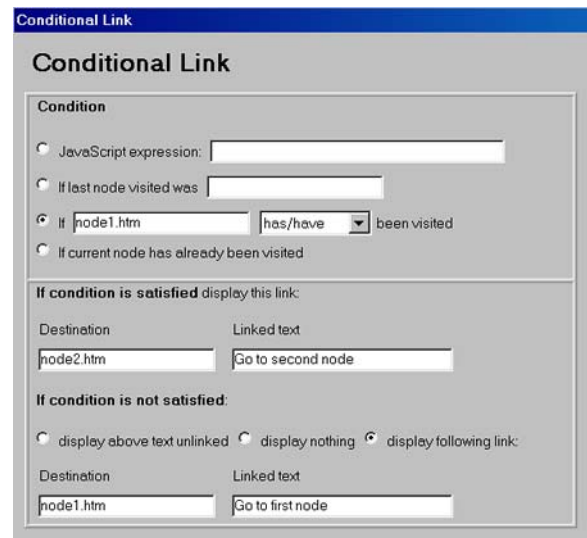


Figure 1: A Connection Toolkit screen.

The JavaScript implementation allows authors to customize and extend the features of the system. JavaScript's ASCII format should make it easier for system elements to be analyzed and understood by anyone studying a work that uses the system or by a future editor attempting to convert system elements to another format. This is an important archival consideration for literary work.

THE STRUCTURE OF READER KNOWLEDGE

AH systems designed for informational documents typically categorize the hypertext's contents into topics or concepts that the reader must learn. This approach works well for education and online help, but the materials of a literary work can rarely be categorized so neatly, nor can the literary reader's goals be defined so clearly. Literary structures are often deliberately ambiguous, and the process of how they reveal information to the reader is as important as the information itself. Elements such as suspense, foreshadowing, and deliberately induced uncertainty make the reading process more than just a simple matter of knowledge acquisition.

This isn't to say, though, that the literary reader's knowledge is undefinable. To understand how user modeling and AH can be applied to literature, we must examine how the reader develops a knowledge of the hypertext story or poem and isolate elements of that knowledge that we can usefully model. Then we will consider how this knowledge can be tapped by the system and used to influence the emergent text structure.

When we talk about the reader's knowledge of a hypertext, we are really talking about several related factors: 1) The memory of the realized text as it has unfolded during the reading. 2) The memory of the interface elements (links, maps, and so on) that enabled this unfolding. 3) An understanding of the conceptual structure that underlies the text. 4) A memory of the process that formed this knowledge of the conceptual structure.

Let's start by considering the work's underlying conceptual structure, which was created by the author. However complex or indeterminate it may be, it will generally include a number of interrelated but identifiable components, such as plot lines, characterizations, themes, unifying metaphors, symbolic elements, and so on. The reader gradually builds a knowledge of these conceptual components through reading the text, and many of them may become clear only through contemplating or analyzing text elements in retrospect. Even if the reader can't quote a single line from the work just read, she is likely to be able to discuss plot events, characters, and other conceptual components from it.

Genette provides a detailed discussion of how the reader extracts a particularly important conceptual component from a work of fiction: The story (*histoire*), or actual sequence of events related, may be presented in the narrative text (*récit*) in a fragmentary fashion. Flashbacks, references to past events in dialogue, and other "anachronies" may require the reader to mentally reconstruct chronologies and other relationships. [10] Genette is discussing printed fiction, but the reconstruction process he describes is even more relevant to hypertext, in which the story is particularly likely to be fragmented. Walker describes the process of reconstructing the story in Joyce's *afternoon*. [25] [11]

The story thus develops as a distinct knowledge component in the hyperfiction reader's mind. If there are several subplots, each will maintain an individual identity for the reader. All of the text relating to a specific character may also cohere as a knowledge component as the reader puts together the often widely scattered bits and pieces to develop a sense of that character's personality and motivations. The reader may also mentally categorize settings, points of view, philosophical themes, types of events, symbols, and moods. A particular text passage may contribute to several knowledge components simultaneously. For example, a plot incident may have symbolic significance and advance characterization. The contents of components will therefore often overlap with one another. The reader is likely to sense parallels and symmetries between various components as themes echo each other in different subplots

and one character emerges as a foil for another. Smaller components will also combine to create larger ones.

Once knowledge components are identified, they reveal a number of different properties. The most important for our immediate purposes is whether the elements that make up a component have a progressive or an associative relationship to one another. If a logic of chronology or exposition dictates the order of elements, they are related progressively and the reader will conceptualize them as a sequence. Even if the text passages underlying the formation of such a component—say a plot line—were encountered out of order during the reading, the reader will try to piece them together retrospectively in her mind to follow the narrative or the exposition. The exact details concerning how the thread was originally fragmented and what separated these fragments from each other may be too complex to remember. The reader may instead retain only what she has pieced together of the thread, along with a general sense that it has as one of its properties a patchwork quality. Of course, the degree of original fragmentation may affect the accuracy of the reader's reconstruction, and pieces of the thread may retain associations they picked up by originally being juxtaposed with other material.

If elements of a component are clearly connected in content but don't inherently demand any particular ordering, they have a purely associative relationship to one another. For example, details of a locale or philosophical musings may be related associatively. A component made up of this sort of material will probably be retained as an unordered cluster or constellation of elements. Even though the reader may have encountered the elements of a constellation in a particular order, if their ordering seems unimportant, it is likely that she won't remember it. Some relationships within a constellation may seem stronger than others, however, perhaps establishing subgroupings in the reader's mind. The original proximity of any two elements of a constellation during the reading will probably affect the strength of the perceived relationship between them.

Another property of a knowledge component is its implicit potential for future growth. If a plot line is left hanging, the reader will sense that it will probably be developed further later on in the text. Similarly, if an interesting character is introduced but then drops from sight, the reader may expect that character to reappear.

Knowledge components are by nature highly flexible and they continuously grow and change during the course of the reading and during the process of critical analysis. Not only can new elements be added to a component, but elements can be shifted from one component to another, as when, for example, a character turns out to be other than what he first appears. New relationships may arise continuously among different components as the reader discovers material that is common to both of them. Two discrete components can grow together to form one, as when the reader discovers intervening incidents that connect two plot sequences.

The way that knowledge components grow in relation to one another during the reading affects the reader's overall conception of the whole. How the knowledge structure is formed is often as important as the knowledge itself. As the structure grows, it creates changing patterns, rhythms, symmetries, and proportions among its components that have an aesthetic affect. The reader is aware of whether the growth process is unified or chaotic, varied or monotonous, rich in surprise and suspense or predictable, fast or slow paced.

MAPPING CONCEPTUAL COMPONENTS TO KNOWLEDGE COMPONENTS

The first task of the Connection System is to track the development of reader knowledge. Recording the reading history helps gauge the reader's memory of the realized text. Gauging reader knowledge of the conceptual structure is more complex. The Connection System does this by means of an *overlay user model*. [3] It lets the author define the important conceptual components of the work and then it monitors the reader's growing knowledge of each defined component. The user model is the continuously updated record of this reader knowledge. Each component of the work's conceptual structure becomes a modeled knowledge component in the user model. As the reading progresses, these components in the user model acquire values indicating the reader's knowledge of the corresponding components in the text. The user model persists from one session to another, so when a reader returns to an unfinished reading days or weeks later, she will be immediately taken to her current place in the text with all the stored knowledge values intact.

The user model is a comparative model that relates the reader's current knowledge of the text domain to the entire domain itself. It indicates not just current knowledge but also the potential for new knowledge. Unlike the actual knowledge components stored in the reader's mind, the modeled knowledge components have explicit rather than implicit properties for potential growth, indicating exactly how much of the component is yet to be read.

There can't be an exact correspondence between the conceptual components as defined in the hypertext and the knowledge components that a reader derives from them. Knowledge components are built not so much from the actual text of nodes or portions of nodes as they are from a reader's memory of the salient aspects of the nodes' content or the general impressions they made. But generally there will be some direct relationship between each element of the knowledge structure and an actual part of the text.

The Connection System relies on automatic user modeling, which updates the values of modeled knowledge components solely by tracking the nodes a reader has visited. Brusilovsky points out that for most AH applications, automatic user modeling is not sufficiently reliable. [3] That a user has read a section of text doesn't guarantee she understands it or will remember it. The standard approaches to cooperative user modeling, in which the user provides direct input for the user model, [12] would generally not be very useful for literature, however. The goal in reading a story or poem is

not to learn the material in the way it might be for an informational text. Periodically quizzing the reader about her understanding of the work would be obtrusive and possibly even insulting. What's more, a reader may not be *meant* to fully understand everything in a literary text.

Since the user model must persist from one session to the next, the Connection System stores all the relevant data in a set of browser cookies. Most AH systems maintain their user models by storing explicit knowledge values. To minimize the amount of data that needs to be stored in cookies, the Connection System stores only the complete history record and calculates knowledge values from it as necessary.

This approach has yielded satisfactory performance with the hypertexts that currently use the system, but it's likely that performance problems will arise with very large hypertexts and very complex user models. To handle such cases, a future version of the system might store explicit user model values as variables in a persistent frame or window. These explicit values could be calculated from the history at the beginning of each reading and then maintained during the reading to minimize computational overhead, but they would not need to be stored between reading sessions.

Currently the Connection System can accommodate approximately 8,000–16,000 entries in the history record using Netscape Navigator (depending upon the length of HTML filenames) but only about 400–1,000 entries using Microsoft Internet Explorer. A possible future enhancement would be to allow aliases for node names, which would expand the capacity of the history record to about 1,400 entries for Internet Explorer, regardless of filename length. We are also considering a version of the system that relies at least partly on CGI or server-side Java, which would ultimately be necessary to handle larger history records and improve performance with very large hypertexts.

DEFINING TEXT COMPONENTS

By letting authors define conceptual text components, the Connection System gives them a means for more clearly articulating the structural possibilities inherent in the material, which can then be selectively realized by the reader. These text components can represent any structural element that encompasses one or more nodes.

Many literary hypertexts treat a group of nodes as an integral unit for structural purposes. Storyspace [24] offers support for stringing nodes together into paths, and the effect of paths can also be achieved in Web hypertexts by assigning path names to links. Other hypertexts create node groupings by means of menus or spatial representations on a map. The structural purposes that these techniques can serve are very limited, however. The Connection System provides a wider variety of options for letting the reader interact with, explore, and evaluate text components as coherent structural entities. The author can even create a link directly to a component and let the system automatically determine the most appropriate destination node within that component.

To create a Connection System text component, the author defines a set of nodes that have some sort of meaningful relationship to one another. She can choose to treat the set as either a sequence or an unordered group, which affects how the relationships among the nodes are realized as links or used to inform other navigational elements.

Components with ordered contents can define any group of nodes that have a progressive relationship to one another, such as narrative or expository threads. Components with unordered contents have a broader application and can define any group of nodes that have an associative relationship to one another. Examples are nodes that all involve a particular character, locale, object, or event or simply share a philosophical outlook or style of writing. Even nodes that share nothing more than moments of high narrative tension can form a component, making it easier to use narrative tension as a structural element.

Components can also define certain relationships, such as chronology or causality, on a large scale. For example, nodes can be grouped according to the position they occupy in the narrative timeframe, so that chronological relationships can be taken into account by functions during the reading.

The author can define navigational as well as conceptual components. Navigational components are higher-level groupings of nodes that don't necessarily represent inherent textual relationships but serve mostly to make navigation easier. A default path or "guided tour" can constitute a navigational component. Such a path may be designed to let the reader sample the most important parts of the hypertext by traversing portions of different conceptual components. Another example is a component containing only nodes that would serve as good starting points for jumping into the reading. Often an author will provide an image map with links to such a group of nodes, to which the reader can return periodically to find fresh entry points.

While letting authors clarify structural elements, text components don't force them to sacrifice flexibility, but rather increase the potential malleability of the text. Components created by the author can be overlaid onto any preexisting HTML structure of nodes and links, making the system unobtrusive for users who wish to apply adaptive methods only to certain elements of a work. The author can define all the textual material as components, create components for only a small portion of it, or forego component definitions altogether. Nodes that aren't included in components are still monitored by the system and properties can be extracted from them.

An author may wish to leave some conceptual components of a hypertext ambiguously or unclearly defined. To accommodate the blurred boundaries often required in literary structures, the system supports alternative versions of components. The content of components can overlap with no limit on the number of different components that can simultaneously contain a particular node.

The current version of the system does not allow components to be explicitly contained within other components, though functions can be performed on groups of components at run time. A new version now in prototype will allow multilevel component containment. This will accommodate any structural hierarchies an author wishes to create, including alternative versions of hierarchies. It will also allow partially ordered components by grouping ordered and unordered subcomponents together. Thus a path could present most of its nodes in a predefined sequence yet contain stretches where nodes appear in a different randomly determined order every time the path is followed.

PROPERTIES OF MODELED KNOWLEDGE COMPONENTS

AH systems typically assign a single knowledge value to each knowledge component in the user model, indicating how well the reader knows the material it represents. Modeled knowledge components in the Connection System have a number of different properties that together indicate reader knowledge.

The most important factor in determining a reader's knowledge of a node is whether or not it has been read. In the case of a component, the primary concern is how many of its member nodes have been visited. We need to determine, however, not only whether a reader has read a node but also whether she may be able to get more out of it by returning to it. Hence there are several other factors that affect a reader's knowledge of a component.

A node's *age*—that is, how long ago it was last visited—bears upon reader knowledge. As discussed in [17], if a node was visited much earlier in the reading, the reader may have forgotten many of its details. Furthermore, a well-aged node is more likely than a recently visited one to have new significance accrue to it upon rereading, since the second reading may benefit from much more background information than was available upon the first reading. Thus the greater a node's age, the greater the chance that there will be something new for the reader to find in it. The system can compare the ages of different nodes in order to help guide the reader back to the oldest material. It would be useful to have an age index for an entire component, based on a value such as the average or minimum age of all visited nodes within it, but this feature has not yet been implemented.

The system tracks the number of times a node has been visited, which is also significant. A reader may find it useful to return once to a node but may not tolerate having it recur three or four times during a reading.

Context is relevant to a reader's knowledge of components. Rereading a node in different contexts can sometimes imbue it with new significance. (See [25] for examples.) Therefore, the reader's knowledge of a node may not be complete unless she has read it in all the possible contexts that may affect its meaning. This is especially true if the node contains context-sensitive variable text. The system can therefore determine whether a visited node was preceded by another specified node in the history.

Another factor bearing upon reader knowledge of components is integrity. The reader will have a much clearer picture of a progressive sequence of nodes if they were all read in the correct order one after another than if they were read out of order and at widely scattered points in the reading. Calculating the degree of a component's integrity would be somewhat complicated, but this capability may be added to a future version of the system.

If a previously visited node contains links to unvisited material, this might also warrant a reader's return to it. Currently the system can find visited nodes that provide access to unvisited material, but the technique requires a fair amount of JavaScript programming on the part of the author. This approach has been implemented in *Penetration* [18].

ADAPTIVITY TO SUPPORT STRUCTURAL GROWTH

There are two types of hypertextual adaptivity defined by Brusilovsky, and the Connection System employs both of them: *Adaptive navigation support* allows links to be displayed or hidden conditionally and provides the reader with information about link destinations. *Adaptive presentation* allows the contents of nodes to change conditionally. [3] Organic structural growth has a number of specific needs that can be served by these adaptive methods, as outlined below.

Flexible Growth

The first criterion for structural growth is flexibility. To ensure maximum flexibility in the arrangement of nodes, the Connection System allows any link to be displayed or activated conditionally depending upon properties of the user model. Conditional links have proven effective in *Lust, afternoon, Victory Garden*, and other hypertext fiction created with Storyspace. [1] [11] [22] But Storyspace only allows link anchors to be made active or inactive conditionally, and it supports a limited range of conditions. The following sections will discuss applications of a very broad range of conditional linking options.

For flexibility at finer levels of granularity, the system uses what Brusilovsky calls *fragment variants* [3] to change the content of individual nodes dynamically. These variable elements can contain any configuration of HTML or JavaScript code to present text, graphics, or multimedia content. They can be embedded anywhere in a node, can be nested inside one another, or can constitute the entire node. A small collection of them can be stored independently of individual nodes for use in any node within the hypertext. Properties of the user model can determine the display of fragments within a particular node, or they can be chosen at random.

Fragment variants can serve two valuable purposes in literary hypertexts. First, they can alter a text so that it will better fit into multiple contexts, creating syntactic or narrative transitions as needed. This allows the reader more options for textual rearrangement.

Second, fragment variants can make node recurrence more interesting and meaningful. As discussed above, rereading

nodes is most fruitful for the reader when it reveals something new in the text. Context-sensitive variant versions of a node can ensure that reading the node in different contexts reveals new meaning in it. Random variations can create a more unpredictable type of variety upon rereading. Sometimes authors create variant versions of a node simply by storing the different versions as separate nodes. In her discussion of Joyce's *afternoon*, Walker points out several instances of this, which she calls false recurrence. [25] Having to create a separate node for each possible configuration of variable text elements limits the author's possibilities and makes revision more difficult. [13] (See *A Life Set for Two*, *Dispossession*, and *Penetration* for implementations of fragment variants in hypertext poetry. [14] [15] [18])

Fragment variants in the Connection System aren't treated as individual components, as they are in some AH models such as the Adaptive Hypermedia Application Model. [8] They aren't tracked in the user model and properties cannot be extracted from them. Nonetheless, the relationship of fragment to node parallels to some extent the relationship of the node to the larger hypertext. The fragment thus brings the multiple textual possibilities of hypertext into the node itself down to whatever level of granularity the author desires.

Sustained Growth

The reading can progress only so long as the reader's knowledge of the work continues to grow. This means continuously presenting the reader with new text or opportunities to find new meaning or structural significance in already familiar parts of the work. Adding a large number of links to each node will increase the chances that a reader will always have unfollowed links available, but this approach can also lead to confusion and lack of direction. An alternative is to make each individual link farther reaching.

The Connection System does this by letting links express large-scale structural relationships rather than just local ones. A link can be made directly to any text component rather than to an individual node. The author can then specify an *entry behavior* for the component by invoking a function that examines relevant properties of the user model and determines the appropriate destination node accordingly.

In the current implementation of the system, specific behaviors cannot actually be attached to text components. A behavior must be defined each time a link is made to a component. A new version in prototype, however, treats components as true objects to which the author can attach behaviors (as JavaScript methods) and properties. For example, an author can create an entry behavior and assign it to the *next()* method of a component called *Ennui*. Creating a link to *Ennui.next()* instead of to a specific node would then invoke the stored entry behavior to select the appropriate node from the component. Additional methods (such as *previous()*, which would enable backtracking through the contents of a component) could also be defined.

A number of different entry behaviors are possible, including randomly selecting an unvisited node or finding the first

available unvisited node in a sequence. A link can even be made to multiple components simultaneously. The constituent nodes of all the destination components can then be pooled in the selection process, or hierarchical filtering parameters can be established to search components in succession for a suitable entry point.

Figure 2 presents an example of linking to an unordered component consisting of six nodes (N1–N6). The reader first accesses the component by following a link (the arrow labeled 1) from a node outside the component. The link leads directly to the component, not to a particular node within it. This first time the component is accessed, none of the nodes have been previously visited, so the reader is sent arbitrarily to N3. Then the reader follows a link from N3 to the component itself and ends up at another randomly selected unvisited node (N1). A static link from N1 takes the reader back to the core hypertext, where she may explore many other nodes. She then returns to the component twice (arrows 2 and 3), invoking the entry behavior until all the component's nodes have been visited once.

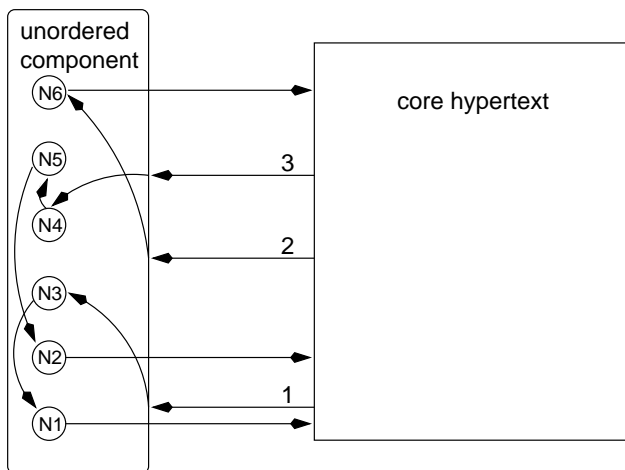


Figure 2: Linking to an unordered component.

This component behavior can be useful in a wide variety of circumstances. If a link anchor attached to a text passage evokes a particular theme or narrative element, the reader's selection of that link can be treated broadly as a desire to build upon the current knowledge of that theme or element rather than merely explore a purely local associative relationship. The same link can reveal different aspects of this textual relationship every time the reader returns to it.

If all the nodes in a hypertext are fairly self-contained in textual content, this technique can even guarantee that unread material will be available from every node and that when all the links on screen are set to the vlink color, the entire piece has been read. The author can create a component containing every node in the work, link to that component from each node, and specify an entry behavior of ran-

domly selecting unvisited nodes. (See the hypertext poem *Peggy* for a simple implementation of this approach. [23])

Another method for sustaining growth is to provide easy access to a number of nodes that could serve as fresh starting points if the reader becomes trapped in a loop or otherwise hindered from progressing. Typically such access is provided in literary hypertexts by a central image map or table of contents to which a reader can periodically return, as in *Notes Toward Absolute Zero* and *Victory Garden*. [21] [22] The reader may find seeking out the map a distraction, however, and there may be a limit to the number of entry points a map can accommodate without creating confusion. The Connection System provides an alternative approach by letting the author create a small menu of links that all lead to different randomly selected unvisited nodes in a single large component. This component could contain all the nodes that constitute logical initial entry points to various important areas of the hypertext.

The menu could appear at the bottom of the page or in a frame when a strategic location is reached, such as the end of a path. It could also be displayed conditionally, for example if all the links on the current page lead to visited nodes or if the reader enters a loop and encounters a number of previously visited nodes in succession.

The presentation of new material isn't the only stimulus for structural growth in a hypertext. It is also important that a reader be able to uncover implications in textual passages that weren't apparent upon first reading. The Connection System can use link annotation, as defined by [3], to help guide the reader back to "partially known" nodes that reveal their full significance only when read in more than one context. *Penetration* uses progressive shades of color in its links to denote knowledge values. [18] Dark green indicates an unvisited node. Light green indicates a node previously visited but from a different antecedent node that showed it in a different light. White (the body text color) indicates a node previously visited in a context that is semantically identical or similar to the context in which it would appear when revisited from this link. Alternatively, the system can indicate such knowledge values by icons preceding or following a link or by either pop-up text (similar to ALT text pop-ups over graphics in Version 4 browsers) or status bar messages activated by mouse-overs.

Sometimes the significance of a passage can be affected by whether the reader has encountered particular background material relating to it. Link annotation can indicate a previously read node as "partially known" if it has been read only before the reader's encounter with background material that affects the node's meaning.

Successfully sustaining growth is ultimately important for achieving closure in a reading. Hypertext readers generally stop reading when they feel that their knowledge of the work is no longer growing significantly. Toward the end of reading a complex hypertext, it can be difficult to determine whether there is any remaining potential for growth. As discussed in [16], this final stage of the reading, or *end reading*,

can involve frustrating searches through previously read material to find what is left to read. The Connection System can not only help sustain growth up until the end but can give the reader direct feedback concerning the remaining potential for growth. Details of this feedback will be discussed in the next section.

The end reading has a character distinct from the other stages of reading because during it the reader is trying to decide whether or not to terminate the reading. [16] Her strategy changes during this part of the reading, so it would be appropriate to let interface and content objects adapt themselves to accommodate the special needs of the end reading. For example, when a specified percentage of the entire text has been read, the Connection System can use link colors to indicate not only unvisited nodes but also well-aged nodes that were visited early on in the reading.

Concurrent Growth

A reader's knowledge of a complex work generally develops through the concurrent growth of many separate knowledge components. For example, if a narration alternates among several plot lines that unfold concurrently in the story's timeframe, the reader's knowledge of each plot will develop progressively, almost as if no interruptions had occurred. The reader of fiction also simultaneously maintains knowledge of different characters, and these knowledge components steadily evolve as the characters come and go in the narrative. The Connection System offers explicit support for the concurrent growth of many different knowledge components, making it easier for the reader to mentally juggle all the components of a complex text.

When the reader returns to a component of which she has a partial knowledge, the behavior of that component can take this knowledge base into account. The selection of a link that leads directly to a text component is interpreted as a desire to build upon knowledge the reader already has of the component, not start anew with it. This ensures that a reader can always move forward in her process of exploring a component rather than, for example, merely looping back through the same portion of the component upon successive visits to it. It also lets her more easily seek out a particular component to deliberately resume her exploration of it.

The entry behaviors we examined in the previous section permit the resumed growth of unordered components. If a component is treated as a sequence of nodes, a different behavior is required. The Connection system supports *retentive paths*, which are usually created by means of an entry behavior that takes the reader to the first unread node in the sequence. This will let the reader either pick up where she left off or—if she has read several fragments of a sequence while leaving unread gaps between them—fill in the gaps in her knowledge of a component. This behavior ensures that an entire sequence can be explored by returning to it several times from any point in the hypertext. In some cases it may be more appropriate to return the reader to the last-visited node in the component, so this behavior is supported as

well. Figure 3 illustrates three different visits to a retentive path via links made directly to the component itself.

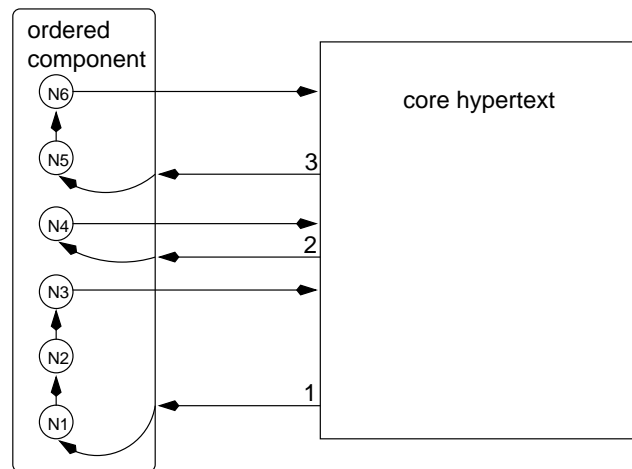


Figure 3: Linking to a retentive path.

Retentive paths can be transparent to the reader. For example, the word “Rachel” within a node might bear an associative relationship to the contents of a path that describes an episode in Rachel’s life. The association might hold equally well between the word Rachel and any of the nodes in the Rachel path. A link from the word Rachel to the first unvisited node in the Rachel path would realize the local associative relationship between the two nodes as effectively as a static link, but it would also ensure that the reader’s knowledge of Rachel would continue to develop.

Readers can also be made aware of links to retentive paths or other large-scale components and given an indication of the component’s potential for further growth. This lets them shift their navigational concerns from the local level to a more global level. A pop-up text message for a link can indicate the destination component’s name and the proportion of unread material it contains. Alternatively, different shades of link color could indicate different knowledge values for a component, such as all nodes visited or most nodes visited.

A more practical method for indicating such subtle differences in value is provided by the system’s graphical gauges, a feature currently in prototype. These are user-created graphics that can be stretched along either the x-axis, the y-axis, or both axes to represent changing values. Thus the size of a graphic linked to a component could indicate how much of it has been read, and the graphic could gradually shrink or grow as the component was explored. The exact numeric value can be treated as ALT text in the tag and thus displayed to the reader. These gauges can also be used without links, for example to create a progress indicator showing how much of the current component or the total hypertext has been read.

Graphical gauges linked to text components can be particularly valuable in central maps or tables of contents. Such maps in hypertext stories or poems typically contain only

links that serve as fixed entry points for important sections of the work. Returning to these static links once they have been followed may offer little help in trying to continue explorations already begun. On the other hand, a map linked to retentive paths can support the concurrent growth of any key components until they have been completely read. Version 2 of *In the Changing Room* offers an example of this. [6] In this work the linked components together encompass the entire hypertext, so a reader can be assured of easily finding every node in the work simply by returning to the map repeatedly.

Sometimes retentive paths by themselves don't provide everything the reader needs to fully resume interrupted knowledge growth. After returning to a partially read sequence and resuming its exploration with an unread node, the reader may want to reread the node or two that comes before this new node in the sequence to put the new material in context. These antecedent nodes cannot be reached by backtracking, however. The reader may also want an idea of what lies ahead. To provide a bird's-eye view of a component's contents and immediate access to all of it, the system supports more detailed graphical representations of this contents and its properties. The author can map out a component using small graphics to represent each node, with alternative versions of the graphics designating the current node and distinguishing between visited and unvisited nodes. Currently the author must build these maps by hand, but we plan to provide automatic generation of these component maps in a future version of the system.

To further facilitate exploring components, we plan to add a Path Navigator, which would let the reader navigate forward or backward along the current path by clicking Forward and Back buttons. As a bonus, it could also let the reader view all the paths that intersect with the current node and choose any one of these as the new default path to follow. Storyspace's Path Browser offers some of this functionality.

Controlled Growth

There are many reasons for a text to foster the concurrent growth of several components during a reading, even when it would be possible for them to unfold sequentially one after the other. Concurrent growth enables interplay among components as they unfold. It can create an impression that events described within two components are occurring simultaneously. It can create variety, surprise, suspense, and interesting textual rhythms. (See [17] for discussion of textual rhythms in hypertext.) In hypertext, it also increases the opportunity for reader interaction and varied textual relationships.

Authors of print as well as hypertext often find the need to interrupt the unfolding of a plot line or characterization and then resume it later. The means of interruption are as important as those of resumption. This sort of deferred growth is often achieved in hypertext fiction by means of breaking up a single sequence of nodes into several separate navigable paths, as in *afternoon* and *Victory Garden*. [11] [22] The paths aren't directly linked together so the reader can never

traverse more than a fragment of the entire sequence at any one time.

One problem with this approach is that while the interruption is guaranteed, the resumption is not. A reader may repeatedly return to one fragment of the sequence but never be able to find its continuation. The Connection System can offer truly deferred growth by combining this conventional sequence fragmenting method with retentive paths. Static links can be used to create the isolated fragments of the sequence, linking together some but not all of the nodes. The entire sequence can then be defined as a retentive path and linked to from various strategic points in the hypertext. Thus each successive visit to the sequence will advance the reader along it, but at the same time each visit will reveal only a portion of the whole. To further accommodate deferred growth, a conditional link can be placed in the last node of each sequence fragment. This link will be inactive the first time the fragment is visited. If a reader returns to the fragment, then the link will lead to the beginning of the next fragment.

The structural growth process in hypertext is highly flexible and adaptable, but sometimes it is necessary for one thing to grow from another. The strategy of deferring growth to create suspense depends upon the growth to be initiated (for example, by introducing a mystery) before it can be deferred and resumed (for example, by revealing whodunit). Some other large-scale elements, such as character development, can be difficult to realize unless they unfold progressively during the reading. Adaptive techniques can control growth when necessary by restricting access to components or the later parts of sequences until all or most of the prerequisite material has been read.

Adaptive techniques can implement mutually exclusive textual elements, such as alternative plot lines. The author can create alternative versions of a component and then remove or hide all links to one version if any nodes unique to the other version have been read.

As well as facilitating structural growth, conditional links and component behaviors can help ensure that growth is terminated satisfactorily when the time comes. Any link can be defined to lead to a particular node or component once all of the other nodes in the work have been visited, thus creating a true ending, as in *Peggy*. [23]

FUTURE POSSIBILITIES

The Connection System and the literary works that utilize it demonstrate some of the possibilities that adaptive hypertext can bring to literature. Future work should open up many more possibilities in giving the reader greater control over the text, allowing greater variety from one reading to the next, and ensuring more satisfying reading experiences.

Among planned enhancements of the system is support for configuration options that can be set by the reader to influence textual elements on a large scale. For example, the reader might be able to select among several moods, points of view, or thematic focuses. The selected option could af-

fect fragment variants throughout the hypertext and could effectively add components to the structure or remove them from it by globally activating or deactivating all links leading to those components. (See *A Life Set for Two* for examples of globally reconfiguring mood. [13]) The reader might also be able to increase the interplay between selected text components by engaging an option that activates conditional links between many of the constituent nodes of those components. Such options to reconfigure large-scale textual behaviors would allow user interaction at a higher level than is possible merely through making navigational choices.

REFERENCES

1. Arnold, Mary-Kim. *Lust*. Eastgate Systems, Watertown, MA, 1993.
2. Bolter, Jay David. *Writing Space: The Computer, Hypertext, and the History of Writing*. Lawrence Erlbaum, Hillsdale, NJ, 1991.
3. Brusilovsky, Peter. "Methods and Techniques of Adaptive Hypermedia," in *User Modeling and User Adaptive Interaction* (1996, vol. 6, nos. 2–3), pp. 87–129.
4. Brusilovsky, P.; Eklund, J.; and Schwarz, E. "Web-Based Education for All: A Tool for Developing Adaptive Courseware," in *Computer Networks and ISDN Systems* (Proceedings of Seventh International World Wide Web Conference, Apr. 14–18, 1998; vol. 30, nos. 1–7), pp. 291–300.
5. Coleridge, Samuel Taylor. "Mechanic vs. Organic Form," in *Shakespearean Criticism*, ed. by T.M. Raynor. Dent, 1960.
6. Craven, Jackie. "In the Changing Room." Version 2 forthcoming in Word Circuits Gallery. (www.wordcircuits.com/gallery)
7. De Bra, Paul and Calvi, Licia. "AHA: A Generic Adaptive Hypermedia System," in *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia* (Pittsburgh, June 20–24, 1998), pp. 5–10.
8. De Bra, Paul; Houben, Geert-Jan; and Wu, Hongjing. "AHAM: A Dexter-based Reference Model for Adaptive Hypermedia," in *Proceedings of Hypertext '99* (Darmstadt, Germany, Feb. 21–25, 1999), ACM Press, pp. 147–156.
9. Fink, J.; Kobsa, A.; and Nill, A. "User-oriented Adaptivity and Adaptability in the AVANTI Project," in *Proceedings of Designing for the Web: Empirical Studies* (Redmond, WA, 1996), Microsoft Usability Group.
10. Genette, Gérard. *Narrative Discourse: An Essay in Method*, trans. by Jane E. Lewin. Cornell University Press, Ithaca, 1980.
11. Joyce, Michael. *afternoon: a story*. Eastgate Systems, Watertown, MA, 1990.
12. Kay, J. "The UM Toolkit for Cooperative User Models," in *User Models and User Adapted Interaction* (1995, vol. 4, no. 3), pp. 149–196.
13. Kendall, Robert. "Hypertextual Dynamics in *A Life Set for Two*," in *Proceedings of Hypertext '96* (Washington, DC, Mar. 16–20, 1996), ACM Press, pp. 74–83.
14. Kendall, Robert. *A Life Set for Two*. Eastgate Systems, Watertown, MA, 1996.
15. Kendall, Robert. *Dispossession*. Eastgate Hypertext Reading Room, 1999. (<http://www.eastgate.com/Dispossession>)
16. Kendall, Robert. "But I Know What I Like," in *SIG-WEB Newsletter* (1999, Vol. 8, No. 2). Also in Word Circuits Commentary. (www.wordcircuits.com/comment/htlit.htm)
17. Kendall, Robert. "Time: The Final Frontier," in *SIG-WEB Newsletter* (1999, vol. 8, no. 3). Also in Word Circuits Commentary. (www.wordcircuits.com/comment/htlit.htm)
18. Kendall, Robert. *Penetration*. (Unpublished, www.wordcircuits.com/penetration.)
19. Landow, George P. *Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology*. Johns Hopkins University Press, Baltimore and London, 1997.
20. Levertov, Denise. "Some Notes on Organic Form," in Stephen Berg and Robert Mezey, eds., *Naked Poetry: Recent American Poetry in Open Forms*. Bobbs-Merrill Co., Indianapolis and New York, 1969.
21. McLaughlin, Tim. *Notes Toward Absolute Zero*. Eastgate Systems, Watertown, MA, 1995.
22. Moulthrop, Stuart. *Victory Garden*. Eastgate Systems, Watertown, MA, 1991.
23. Réty, Jean-Hugues. *Peggy*. Aix Art Contemporain, 1999. (www.aix-art-contemporain.org/cine_video/imageswebart/animations_f/rety_f/index.html)
24. *Storyspace*. Eastgate Systems, Watertown, MA, 1999.
25. Walker, Jill. "Piecing Together and Tearing Apart: Finding the Story in *afternoon*," in *Proceedings of Hypertext '99* (Darmstadt, Germany, Feb. 21–25, 1999), ACM Press, pp. 111–117.

