

Storyspace 1

Mark Bernstein

National University of Singapore
Eastgate Systems, Inc.
134 Main Street Watertown MA 02472
+1 (617) 924-9044
E-mail: bernstein@eastgate.com

ABSTRACT

Storyspace, a hypertext writing environment, has been widely used for writing, reading, and research for nearly fifteen years. The appearance of a new implementation provides a suitable occasion to review the design of Storyspace, both in its historical context and in the context of contemporary research. Of particular interest is the opportunity to examine its use in a variety of published documents, all created within one system, but spanning the most of the history of literary hypertext.

Categories and Subject Descriptors

H.5.4 [Hypertext/Hypermedia]: Architectures, Theory, Navigation, User Issues.

General Terms

Management, Documentation, Design, Economics, Human Factors,.

Keywords

Storyspace, hypertext, hypemedia, literature, fiction, education, design, implementation, support, history of computing, maps, links.

1. Storyspace

Storyspace was first publicly demonstrated at the first ACM hypertext workshop in November, 1987, by Michael Joyce, J. David Bolter, and John B. Smith. *Afternoon, a story*, Michael Joyce's classic hypertext fiction whose genesis was tightly bound to the original development of Storyspace, was published by Eastgate Systems in 1989, and Storyspace itself was published by Eastgate in 1991.

Nearly ten years later, Storyspace for Macintosh release 1.5— the last descendant of the original Storyspace implementation— was replaced by Storyspace 2, a fresh (but faithful) reimplementati on of the program. In the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee

, June 11-15, 2002, College Park, Maryland, USA.
Copyright 2002 ACM 1-58113-477-0/02/006...\$5.00.

years since 1991, Storyspace has undergone many revisions, enhancements, and one completely new implementation, Storyspace for Windows. It has been widely used for teaching hypertext writing for crafting hypertexts, and for studying published hypertexts; at times, Storyspace has seemed almost synonymous with literary hypertext.

While Storyspace has never been a notable commercial success, for over a decade it has served an active artistic, scholarly, and critical community. Hypertexts written with it continue to be read as eagerly and discussed as widely today as they were a decade ago. Such durability is exceptional in any software product; in shrink-wrapped, consumer software, Storyspace's longevity may well be without precedent¹.

Though aspects of Storyspace were discussed in [10] [23] [22], [7], and [8], no design paper for Storyspace has been presented. As Storyspace moves into its second decade, it seems a suitable moment to look back on the experiences of designing, modifying, and using Storyspace, and supporting Storyspace writers.

2. The Environment, Circa 1987

Storyspace has always been intended to run well on modest hardware, but the definition of "modest" computational resources has changed substantially since 1987. Comparing the target consumer environment for Storyspace 1 and Storyspace 2, we find that the "typical" memory allotment has increased a thousand-fold (from 512K to 512M). CPU throughput has increased somewhat more. Standard hard disk storage has grown even faster, from 440K to 10G.

3. What was novel in 1987

Storyspace hypertexts consist of nodes, or writing spaces, that are connected by directed links. The text of a writing space appears in its own window (a text window). Following a link causes a new text window to appear (optionally closing the previous window).

¹ Note that it is not merely the name or the brand that has proven durable. In most ways, the experience of reading early Storyspace hypertexts today is unchanged from the original — even though today's *afternoon* or *Victory Garden* happens to share neither code, nor hardware, nor operating system with the original.

Presentation of hypertext nodes in separate windows remained, in 1987, a somewhat controversial choice. Peter Brown's *Guide* [11] argued cogently that viewing multiple nodes together in context would prevent disorientation. Ted Nelson, had long advocated continuous, transclusive displays [38]. Conversely, HyperCard [2] and KMS [1] displayed each node in the entire screen frame, deliberately suppressing the complexity of multiple-window interfaces in favor of a simpler, more immersive style. The original Storyspace *reading* environment, on the other hand, used only a single text window (sometimes augmented with a map window and tool palettes), thereby avoiding the collage of distinct writing spaces that was seen in NoteCards [33] and Intermedia [46].

Storyspace links are *directed*, distinguishing source from destination both in their internal representation and in their externally visible semantics. Bidirectional links were popular in 1987, on both rhetorical and systems engineering grounds, and lasting prejudice against unidirectional links led many to anticipate that the Web would prove impossible to maintain. Nonetheless, the asymmetry between link source and link destination has proved important in developing and sustaining narrative thrust and in driving the reader toward the conclusion of an argument.

Storyspace documents are typically contained in a single file rather than spread across multiple documents. Writing spaces are organized into a single, distinguished, hierarchy, as in KMS, although the interface takes care not to assign explicit semantics to this hierarchy². It was once feared that the visible presence of a distinguished hierarchy might lead writers to rely too heavily on hierarchical navigation while neglecting the use of links, but over-reliance on hierarchy seems not to have been common, even among student writers.

By contrast, many Web sites depend extensively or entirely on hierarchical navigation. The source of this difference might lie in the ease of creating links in Storyspace writing, the obscurity of Storyspace's facilities for hierarchical navigation, some other way in which Storyspace privileges links over hierarchy, or elsewhere entirely. The presence of a hierarchical backbone establishes in the reader's mind a plausible sequence for operations (e.g. printing) that iterate over the nodes in a hypertext, thus tending to reduce accidental astonishment while reinforcing the reader's model of Storyspace's internal operations.

Storyspace provides multiple views — outlines, charts, treemaps — but the preferred interface for most users has been the Storyspace map. Hypertext maps were considered vital in 1987, as the existence of the "Navigation Problem" had not yet come into dispute [28]. Conklin's influential

² Movement in the hierarchy is effected by unlabelled arrow buttons, and Storyspace documentation took care to avoid suggesting that containment meant or should mean specialization, is-a, etc.

review article led many to question whether a system without maps could be considered a hypertext system at all [12]. Generating lucid maps of large hypertexts presents a formidable challenge [4], but if writers are asked to build maps or supply related metadata as a separate activity, many may postpone the added work indefinitely [32].

Storyspace achieves a useful compromise by constructing a global map, but requiring users to perform the layout manually whenever writing spaces are added or moved.

4. What is novel in 2002

Though Storyspace is a very old hypertext system, some aspects of Storyspace remain novel.

Storyspace links may possess **guard fields** — Boolean expressions based on the reader's selection and previous trajectory. When the guard field predicate is based on the reader's selection, guard fields offer a simple generic link facility [35]. When based on the record of the reader's previous path through the hypertext, guard fields offer dynamic links whose behavior changes in the course of a reading [9]. In the latter role, guard fields proved invaluable for breaking cycles and helping to situate Cycle and Counterpoint at the heart of contemporary hypertext narrative [6, 21].

A fundamental hypertext design controversy, discussed as actively today as it was in the 1980s, is the use of external versus internal links. Storing links within the hypertext node, as in HTML, facilitates local editing but renders large-scale link consistency difficult to ensure. Storing links in separate external files, as in Intermedia [46] and the Open Hypertext Systems [41], can make consistency easier to achieve at the cost of complicating local edits. (See [13] for a superb review of this issue).

Storyspace achieves an interesting compromise between these approaches by **storing links internally but representing them externally**. Links are stored inside the monolithic Storyspace file; users don't see separate link files and need not be perplexed by subtleties arising from opening or closing the "wrong" link collection. But Storyspace links are represented separately and compactly, rather than being spread implicitly through the system; a Storyspace link includes

Figure 1. The Storyspace Link

Link	
	Source ID
	Source span
	Destination ID
	Destination span
	Path name
	Guard field

The source ID and destination ID identify nodes by an identifier that is guaranteed to be unique within a document, and intended to be globally unique: the node's

creation time, in milliseconds, augmented by the initials of the node's creator³. In consequence, links between distinct documents are entirely feasible, and links to published hypertexts retain their identities across different editions, platforms, and servers. This facility has not been used extensively, for Storyspace is not intended as a distributed hypertext medium, but has proven useful in pedagogical contexts [27].

Storyspace offers several views, but the **Storyspace map** is its most distinctive and most-used view. In the map, writing spaces appear as title rectangles (Figure 2)



Figure 2. A single writing space as it appears in a Storyspace map. This writing space contains several spaces, three of which are visible here.

Writing spaces may contain other writing spaces, and the hierarchical relationship between spaces may be changed by dragging one space into another, or by moving spaces in the map plane⁴. Every writing space appears exactly once in the containment hierarchy.

Links in the Storyspace map appear as arcs, with an arrow pointing to the destination. If the link has a path name, the path name appears at the link's midpoint. Early Storyspace versions draw links near links; later versions can be set to draw Bezier curves instead [37].

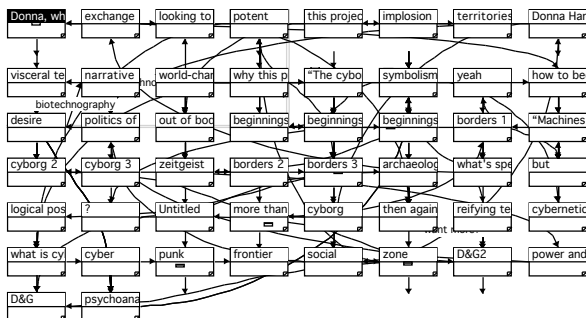


Figure 3. A dense link network, from *Cyborg: Engineering the Body Electric* [15]

³ The use of creator initials seems overly elaborate. In practice, however, a timestamp collision was actually observed between early editions of two published hypertexts(!)

⁴ The relative order among siblings is deduced by scanning the map left-to-right, top-to-bottom.

Dense link networks are often confusing. Storyspace helps elide unwanted information by drawing some links schematically. If a link's source and destination appear in the same level of the containment hierarchy, the link is drawn as an arc between them; otherwise, only a small stub of the inbound or outbound link appears. This information hiding helps keep maps useful by suppressing rarely-used detail.

The graphic representation of some link networks can nonetheless prove incomprehensible. This is sometimes what the author of the links intends. At other times, representational complexity merely breeds confusion; a common strategy for recovering from overly-complex link networks is to partition one complex map into several independent sections, each placed deeper in the hierarchy.

Storyspace offers privileged **default links**, links that are activated by pressing <return> or by clicking outside any text link. Default links frequently play an important role in shaping the reading experience, either by setting up a primary path from which departure is easy, or simply by changing the rhythm of reading, encouraging a more relaxed and less introspective approach without rendering the reader completely or permanently passive.

Finally, **Storyspace text link anchors** are boxed text, revealed when pressing a designated key, rather than typographically distinguished text as is common in current Web browsers. This design was hardly novel in 1987, for the problems of usability and unwanted emphasis that typographically distinguished links present were already understood. In avoiding these problems, now endemic in Web browsers and help systems, Storyspace appears more innovative today than was the case a decade ago.

Because text links are revealed by pressing a special key with the hand that doesn't hold the mouse, Storyspace encourages a **two-handed reading posture**. Changes in body position have always played a role in the perception of reading, and such matters as the shift from lectern to library table, the transition to silent reading, and the introduction of artificially-illuminated (and heated) reading spaces have all played subtle roles in the development of writing style [19, 24, 27]. It is conceivable that the utility of keeping one hand on the keyboard while using the mouse might tend to promote a different reading attitude than a one-handed posture, and this might be a fruitful area for empirical investigation.

5. Storyspace in Use: Maps

Because the Map View is the most capable Storyspace view, most writers use it extensively and its affordances, in turn, often shape hypertexts written in Storyspace. To explore some of the uses of the Storyspace map, we might examine an assortment of 28 published hypertexts (Appendix A).

The confusing densely-linked map in Figure 3 appears as a regular grid in the published hypertext, a format that facilitates scanning for titles and relative position in the

window. It is fairly easy, however, to reposition the writing spaces in the map to help clarify the link structure, albeit at the cost of using screen space far less efficiently (Figure 4). Note, too, the prominent use of cycles, interlocking sequences, and contours of adjacent and interlocking cycles. This map, the largest in *Cyborg*, is by no means unrepresentative.

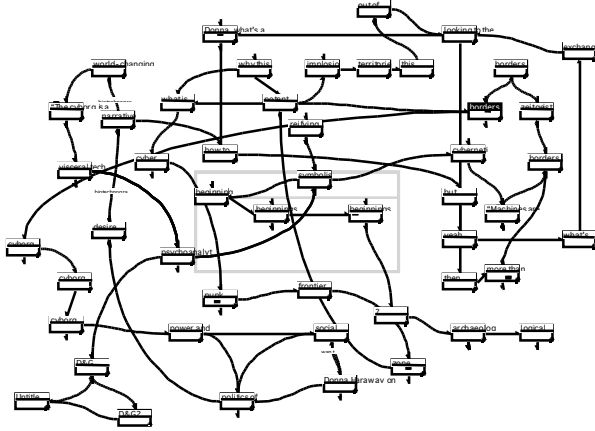


Figure 4. A map from *Cyborg*, rearranged for clarity

This map contains 50 nodes (of 567 in the hypertext) and 155 links. Of these links, 63 (41%) connect spaces within the map, 36 links enter the map from other parts of the hypertext, and 56 exit the map. These values are typical of the largest maps that appear in the 28 hypertexts examined.

	<i>Cyborg</i>	Mean of 28 hypertexts
Nodes in largest map	50	52
Internal links	63	68
Inbound	36	65
Outbound	56	56

Properties of the largest map in *Cyborg* compared to the data set mean

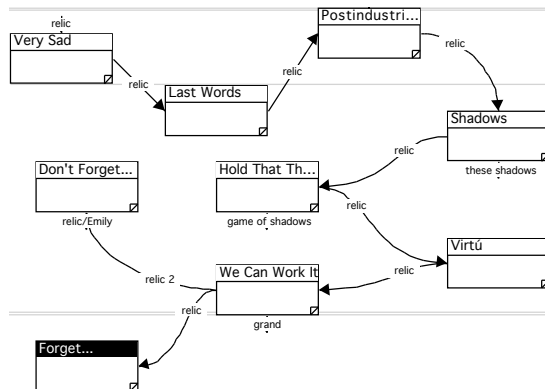


Figure 5. The largest map in *Victory Garden*

A key design goal for Storyspace was to promote fluent hypertext writing, rather than the creation of nearly-linear

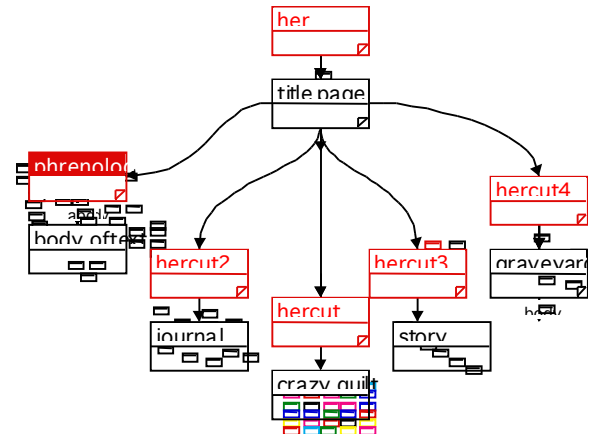


Figure 6. The top-level map in *Patchwork Girl* (Jackson 1996) may reflect both construction scaffolding and the overall hypertext structure

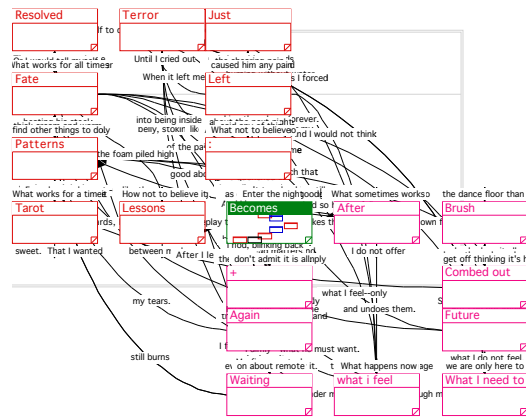


Figure 7. The story "Interlocked" from *Samplers: Nine Vicious Little Hypertexts* is inspired by a traditional quilt motif.

sequences and outlines. A qualitative review of the 28 documents examined here suggests a thoroughgoing embrace of complex hypertextual structure, an impression born out by their link density. A sequence of 50 nodes requires 49 links, so all these maps are much more densely linked than a simple sequence with occasional cross-references. Even considering only the links within the map, the patterns of linkage do not seem well approximated by a sequence.

Other maps do contain sequences, of course; the episodic structure of *Victory Garden* [36], for example, is reflected in its maps. Here, many of the maps reflect episode sequences; the dominant internal path follows the course of the episode while inbound and outbound links reflect connections between episodes.

The Storyspace reading environment allows authors to remove map views entirely from the readers' experience. Some authors take advantage of this to use the map views for organizational or other editorial purposes: for example, the top-level map in *Victory Garden* contains unlinked containers named Done I, Done II, Done III, etc. I presume these were used by the author to mark completion of various elements of the hypertext; as the map should always remain invisible to readers, the author has left his scaffolding in place.

Other writers intend readers to see the maps, and use visual characteristics of the map view to achieve a variety of ends (Figure 6). In *Sampers* [31], Deena Larsen sets out to explore some formal characteristics of hypertext prose by constructing nine short hypertext stories, each based on a traditional American quilting pattern (Figure 7). The pattern of linkage weaves through the simple abstract geometry, but the map (which is visible to the reader) shapes the reading experience.

Others use the map as a symbol, or as a visual pun. In Timothy Taylor's "LBJ" (collected in [29]), the map initially appears to be a simple cluster but upon zooming out assumes a very different aspect.

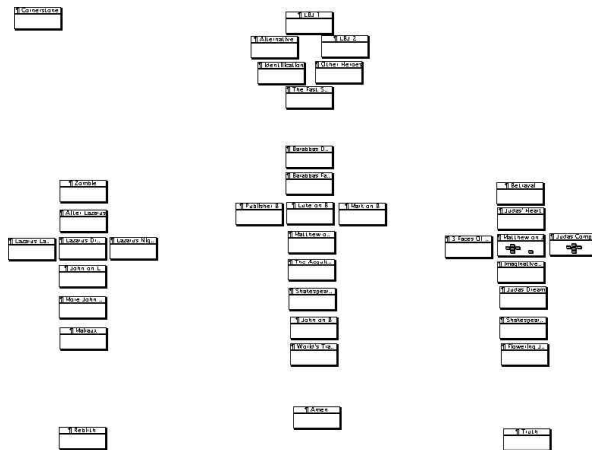


Figure 8. Timothy Taylor's "LBJ" (from [29])

Because the map view has limited facilities for manipulating a hypertext's hierarchy, we might expect habitual use of the map view to encourage broad, shallow structures. The mean depth of the document tree was 7.1 (median 6); the shallow documents are often too small to be very deep. The use of such depth is surprising in view of the obstacles the user interface imposes for building deep tree, although the deepest tree, *A Dream With Demons* [14] (depth 44) is a special case where the author nests spaces for narrative effect.

Notoriously, *after noon* [20] contains a single writing space that lacks any inbound links; since *after noon* is intended to be read in a Storyspace page reader that provides access to writing spaces only through links, this text is notionally unreadable:

"Man... never perceives anything
and only Jane Yellow leaves Douglas has read this screen.
That's not true. so have others.
"To be born again, first you have to die." *The Satanic Verses*

Nodes which have text but no links, known as *Jane's spaces*, might reflect an author's attempts to force readers to use map navigation, as well as authorial deletions, personal notes, messages to critics or collaborators, or mistakes. Jane's spaces are, at any rate, remarkably common; in our sample of 28 hypertexts, only twelve lacked at least one Jane's space.

6. Storyspace in use: Links

Beyond the Storyspace map, dynamic links and guard fields are perhaps Storyspace's most distinctive feature, and their utility in creating structure in large hypertexts, especially in hypertext narrative, has been widely discussed [6, 34, 44, 45]. It is therefore interesting to note that Storyspace writers use them sparingly. In the 28 hypertexts examined, 7 used guard fields ubiquitously (in more than 20% of all links). In eight hypertexts, guard fields are used occasionally, while in 13, the guard fields are completely absent. Guard fields rarely require more than one clause: the mean guard field in *after noon* has 1.63, but few other hypertexts approach this number. "Lust", a notoriously complex network, averages only 1.04 clauses per guard field.

Though guard fields may be used sparingly, links are not. On average, each writing space in the corpus has 3.5 outbound links. The most densely linked hypertext, *True North*, has an average of ten links per node, while only *after noon*, *We Descend*, *Patchwork Girl*, *Quibbling*, and *Genetics* average fewer than 2 links per node.

The size of the link networks in these documents is often formidable. "In Small & Large Pieces", a story of just

13,000 words, has 2,622 links. "Lust", with just 1,731 words, has 141 links.

7. Unloved Features

A number of Storyspace features have attracted surprisingly little interest, despite their prominent visibility and apparent ease of use. The Storyspace reading environment offers **navigation by typing** and conspicuous **yes/no** buttons, and when *after noon's* first page asked the reader

Do you want to hear about it?

the availability of the "no" button marked an important break with computer game convention [10] that situated the reader as hero-protagonist. Though this symbolic and effectual break has proved significant and fruitful [5], and though the features have remained available in each new version of Storyspace, they have seldom been used again.

Paths and link types in Storyspace have proven less popular than might be expected. In this case, the initial implementation, which substituted named paths for the more elaborate type mechanism of NoteCards [17], may well be at fault; by making link labels less semantically powerful (and so less onerous to create), Storyspace may have made them insufficiently expressive. A few hypertexts (notably Kolb's *Socrates in the Labyrinth*, for which see [26]) use named paths and the path browser for their intended purpose; others (*Samplers*, *A Dream With Demons*) discovered that they provide yet another site for inscription, a new writing space where authors could demonstrate that any potentially signifying element can and will become a site of inscription [29].

8. Support

Software debugging and technical support are usually considered quite distinct from research and design, and indeed are frequently treated as symptoms of managerial or technical failure. Support has rarely, if ever, been discussed in these Proceedings, but a decade of Storyspace support does reveal a number of useful lessons both for the design and implementation of hypertext systems and for understanding the way hypertext tools are used. As is usually true, support, enhancement, and maintenance have required the preponderance of Storyspace's development budget over its extended lifetime and have played a major role in the acceptance, and in the acceptance of the underlying technology in actual practice.

The Lit Crit Hotline. Even when a program's behavior is defined by clearly specified actions to be performed on designated input, software support can prove difficult, and the technical, logistical, and emotional challenges of live tech support are widely known. Supporting new writing tools and published hypertexts poses even greater challenges, because the performance of the software and the effectiveness of the text may seem to the reader to be inextricably entwined.

A challenge peculiar to hypertext support is the difficulty of disambiguating a request for purely technical assistance from a request for help with rhetoric or literary interpretation.

Caller: I've been reading *after noon*. [pause] Or trying to. It just doesn't work.

Eastgate: Can you describe the problem exactly?

Remember, I can't see your computer, so lots of detail helps.

Caller: Well, I've tried reading several times. I click, I get to a new page, but after a while I always just seem to lose track of exactly what is happening or who is talking. **It just doesn't seem to work for me.** Am I doing it wrong?

The initial complaint suggested an installation problem – perhaps a defective disk – or a fundamental misunderstanding of the documentation. The actual "problem" was quite different (see [45] for a strikingly similar account). In diagnosing perceived problems with a hypertext or hypertext system, the boundary between software engineering and literary criticism can prove remarkably permeable.

Diagnosis and the Exam Week Bug: A challenging diagnostic problem sometimes arises from subtle interplay between technology and social practice. Some years ago, we identified a strange pattern in support calls that shared a number of common characteristics

- The callers had experienced file corruption causing them to lose valued work. Many had no effective backups.
- The callers worked in large networked labs, at a time when large networks were not yet common. Most were at large universities.
- The callers were facing very tight deadlines: the calls were invariably received at the end of a semester or the end of an academic year, often only hours before a critical project was due.

The key common element appeared to be the presence of a large network, leading to a fruitless search for problematic network support code. Repeated attempts to identify the problem failed, and the concentration of calls in narrow parts of the calendar year defied explanation and strained credulity. Why should a reliable system suddenly prove fallible in December and April?

In the end, a subtle memory issue in print driver support was located and corrected. The problem arose from the operational sequence Edit-Print-Edit-Save, but *not* from the sequence Edit-Save-Print-Edit-Save: saving the file before printing avoided the problem (because saving as a side-effect cleaned up the memory manager). Printing in Storyspace is a fairly unusual activity; when Storyspace users print at all, they often print finished documents (and

so do not edit after printing). Students facing deadlines were especially likely to print their work, find a mistake, fix the error, and print again – thus triggering the bug. The same students, inexperienced and sometimes fatigued users working under time stress, were most likely to omit backups, magnifying the consequences.

Storyspace Ate My Links: Just as unusual usage patterns can reveal unexpected infirmities in a system, habitual patterns may mask instabilities. Some time after the Exam Week Bug was discovered in Storyspace for Macintosh, a seemingly similar issue arose in the independent Windows implementation: a group of users, chiefly students, experienced inexplicable file truncation. Again, reports appeared only after extensive testing and widespread deployment, and although anecdotes suggested that the issue might be locally common it proved impossible to elicit a repeatable example. Because the immediate symptom involved file truncation, the file handling or serializing routines were suspect, but systematic unit testing, manual testing, and code inspection all failed to locate the source.

In the end, the problem was accidentally reproduced in the course of documenting an unrelated feature and proved to lie in management of link objects. Astonishingly, an error in the link manager was found that should have prevented link editing from functioning at all, yet this code has been deployed and used regularly for extended periods and was known to work reliably. As it happens, the defect was bypassed by optimizations that seemed to be special cases but accounted for nearly every link edit:

- If the link being edited happened to be the most recently-created link, the memory manager could be bypassed and the bug never arose. This turns out to be a very common case.
- If the link object after editing happened to be the same size, or smaller than, the original link object, the objects were swapped *in situ*, and the bug never arose. This, too, turns out to be common.

These optimizations were written because they were easy to write and test, and the optimizations were in fact correct and usually masked the faulty manager. Even where the defective link manager was used, frustratingly, the problems were often masked:

- If the link object after editing was larger than the original link object, the bug would only appear if the sum of the lengths of the guard field and path name were odd – about half the time, on average.
- Even if the bug were triggered, the lapse might turn out to be inconsequential. A deeper layer of memory management happened to hide its consequences of the for roughly 85% of all links. Worse, in small files (including all our unit test cases and many of our hand-testing scenarios), links *never* overlapped a memory page and the bug never appeared.

As a result, a procedure that could not and did not work turned out, in practice, to work so well that it made locating the error extremely difficult.

Of the support issues discussed here, this is the only example that should yield to the integrated testing discipline advocated by Agile Development [3]. In order to diagnose the problem, however, the link manager's unit test would need to work with a realistic document (since small data sets never spanned more than one memory page) and would need to have foreseen the even/odd length dependence. Real hypertext testing requires real hypertexts.

9. Unexpected Affordances, Surprising Applications

As might be expected, Storyspace has been used in a wide variety of settings, locations, and environments. To some degree, the distribution of users and applications may reflect inherent strengths of the program or of the underlying technology. In other cases, of course, accidents of timing, location, marketing, and procurement may play a dominant role. It is difficult to know, for example, whether the academic flavor of the Storyspace user population is primarily due to its inherent suitability to the task, to its publisher's marketing methods, to the availability of substantial academic discounts, or to other factors entirely.

A wide variety of interesting case studies of Storyspace use have appeared. Space permits brief mention of only a few examples. Landow [27, 30] has reported on instructional use of Storyspace hypertexts in the classroom over the course of many years. Pamela G. Taylor, in her doctoral study on using Storyspace for secondary school art courses, found that long-term portfolio building using Storyspace's hypertext linking promoted facets of personal psychological growth [43]. For example, a student discovered, over the course of two years of creating and interlinking her art, that a theme of physical abuse reflected problems in her life that she had the power to change. Here, too, it is difficult to distinguish the effects of the program from those of the instructor, the student, the circumstances, and the times.

Over the years, many unexpected applications of Storyspace have emerged, and Storyspace has prospered in unexpected niches.

Storyspace, intended as a writing tool, has acquired a significant following as a tool for qualitative analysis in the social sciences and in film continuity. In both cases, its popularity arises from its implicit support for organizing without premature commitment – user interface affordances desirable (but perhaps not strictly essential) for the program's primary mission. Concrete visualization and graphic organizing tools help reassure investigators that proposed organizations are provisional, that assigning an artifact or an observation to a category may be quickly, invisibly, and seamlessly changed at a later date. Because links are never disrupted by reorganization, they provide a

useful way to retain connection in the face of massive change.

Easy creation of new writing spaces and facile reorganization also made Storyspace useful for a variety of note-making tasks, particularly in managing small workgroups. Project diaries and event planning notebooks took advantage of links to maintain connections in the face of frequent revision. Some writers have found the same facilities useful for plotting novels or planning videogame levels; here, the abstraction of compact writing spaces and the specialized writing environment also help writers work broadly and abstractly, without being tempted to engage details prematurely.

10. Headaches and compromises

Some aspects of the original Storyspace design were doubtless ill-advised. Other compromises, imposed by technological necessity, need no longer burden us. It may prove worthwhile, in passing, to take note of these, lest similar conditions arise again⁵.

Automatic Layout. The success of the Storyspace map is due, in part, to the grace with which it shifts the burden of choosing where to place each writing space onto the user. Locating a space is an inherent part of creating a new space or moving one from another document, and hence users are never tempted to postpone or neglect map maintenance (cf. [16]). Because choosing a place is integrated with creating a space, map-making is not perceived as a separate activity.

The position of a writing space in the hierarchical backbone is derived from the map position, with siblings ordered left-to-right, top-to-bottom. This dependence leads to two unfortunate effects. First, a small change in the map location of a writing space may lead to a large change in its outline position. Worse, if a writing space is moved in the hierarchy (e.g. by dragging it from out place to another in the chart view), its Map position must be derived from the outline position.

Early versions of Storyspace would automatically “clean up” the map positions of all siblings when one sibling was moved in an outline or chart, arranging all writing spaces in a neat rectangular grid. This behavior was intensely disliked by many writers – especially those who had constructed elaborate map layouts which Storyspace would

“clean up” into a rectangular grid without warning. Later, automatic cleanup was abandoned and the repositioned space was inserted in a consistent map location, even if this required numerous writing spaces to be crowded into a small area. This proved inelegant but less harmful than automatic cleanup.

Serialization. Storyspace 1 files were, in essence, flattened images of the memory representation of key Storyspace data structures. A 200-byte header provided a table of contents into the rest of the file. This was followed, in turn, by a heap of text strings, style vectors, and images. Next, the file contained a *struct* for each writing space, a *struct* for each link, a list of father-son pairs sufficient to reconstitute the document hierarchy, and a list of open windows. The resulting file format was very easy to write, requiring almost no memory overhead. This format was also easy to read, since the data buffers read from disk could be converted to live data with little more than a cast.

Speed was essential. When *Victory Garden* [36] was published in 1991, it took five minutes to load, a delay barely tolerable for reading and decidedly unpleasant for writing. This performance was achieved only at the cost of making Storyspace 1 files inherently brittle; if any part of the file were incorrect, the entire file was likely to prove unreadable. Storyspace 2 continues to support this format, but the underlying engine reads and writes XML files, incurring a substantial penalty in memory overhead and time efficiency but greatly simplifying the discovery and repair of file errors.

Uniform Accessors. Storyspace for Windows was written as a monolithic process in idiomatic C. The notion of using a separate storage manager (as in Intermedia [42]) was considered and rejected on the grounds of efficiency. Nevertheless, access to data structures in Storyspace for Windows was systematically constrained through access or functions, giving the code the flavor of a loosely-coupled system.

This ought to have facilitated testing, debugging, and ultimately to have enabled experiments that would replace Storyspace’s back-end with, for example, an open hypertext system link manager. In practice, none of this proved feasible, and the accessors imposed a continual complexity tax without offsetting benefit.

Memory Management. The original Storyspace for Macintosh relied on the application heap for dynamic memory allocation, and implemented its own fallback and reclamation procedures when memory ran short. Storyspace for Windows, faced with what was then an even less trustworthy operating system memory manager, implemented its own private allocation on heap. Considerable effort was devoted in both cases to avoid heap fragmentation and related pathologies.

This investment proved ill-advised. In principle, fragmentation might have been a problem, but in retrospect a naive approach would probably have proved

⁵ Such conditions do arise. Very early databases, for example, developed techniques for working in environments where bulk storage was tightly constrained. The same techniques proved useful in early personal computers, where available disk sizes were often smaller than RAM. Threaded interpreters, developed to take advantage of expensive (and sparse) core memory, proved invaluable in embedded systems where trimming memory footprints could have important ramifications for manufacturing and power budgets.

sufficient. The allocation blocks were either regular in size (writing space records), short lived (i/o buffers), or essentially permanent (texts inside writing spaces); it now seems likely that *any* allocation strategy might have succeeded about as well. Increases in available physical (and virtual) memory soon rendered elaborate heap management obsolete, but the added complexity of specialized allocation and recovery routines permanently burdened the code.

11. Storyspace 2 and Beyond

Storyspace 2 for Macintosh, introduced in 2001, is a careful recreation of Storyspace in a completely new computing environment. In planning Storyspace 2, great weight was placed on preserving reading semantics in existing hypertexts, so that well-known hypertexts would remain readily available to future audiences. It is important to observe that this did not prove particularly difficult or expensive; preserving our electronic texts indefinitely will not prove onerous, provided an audience continues to be interested in the works.

Storyspace 1 depended on a host of specialized data structures and indexes, all designed to speed performance. For Storyspace 2, on the other hand, performance was rarely an issue, and Storyspace 2 rests on a simple attribute-value store developed for a Tinderbox – a new hypertext platform otherwise unrelated to Storyspace. The common back-end might, in principle, provide structure services[39] although at present it merely provides a simple frame store. Though Storyspace 2 continues to rely on its legacy file format, XML serialization services provided by the back-end are already used for interprocess communication.

Improving a literary machine can be a precarious process. Just as it is important to preserve existing works, it is desirable to ameliorate unwanted defects and to take advantage of improvements in typography, image support, and hardware performance. Any change, however slight, might possibly change the reading experience; one critic has written at some length about the impact of minor user interface changes (such as replacing menu items with buttons) in *after noon*[18], while another dis course explores the history of that title's icon[25]! Deciding precisely what lies “inside” the hypertext and what is merely its cover, its wrapping paper, or its ephemeral situation, will require experience, patience, and discussion[40]; the codex underwent the same process [19, 24].

Having assured conservation of Storyspace hypertexts and of its writing environment, we turn now to planning Storyspace 3 in order to explore new directions in hypertext narrative. Of particular interest will be work to extend and enhance dynamic links. Just as the importance of dynamic links is clear, the limited expressiveness of guard fields, the formidable obstacle they present to new writers, and the difficulty of debugging and correcting the complex, distributed finite state machines they implicitly represent,

are all now abundantly evident. We look forward to new work to enhance current facilities while opening new vistas.

12. ACKNOWLEDGMENTS

Storyspace™ is a trademark of Eastgate Systems, Inc. The myriad invaluable contributions of Eastgate employees, authors, customers, and others over the years cannot be underestimated, and it is impractical to hope to enumerate them all here. Eric A. Cohen wrote the Storyspace manuals that have instructed generations of users. Eric Cohen, Colleen Humphries, Kathryn Cramer, Diane Greco, Charles Bennett, and Barbara Bean have all provided Storyspace tech support at various times. Portions of this paper have appeared previously in different form, are © Copyright 2002 by Eastgate Systems, Inc., and are used by permission. I am grateful to the National University of Singapore for a fellowship during the period in which this work was completed.

13. Appendix A

The 28 hypertexts considered include all 27 Storyspace hypertexts published by Eastgate between 1989 and 2001, and one (WOE) published by the journal *Writing On The Edge*.

A Dream with Demons	Edward Falco
afternoon, a story	Michael Joyce
Completing the Circle	Michael van Mantgem
Cyborg	Diane Greco
Dickens Web	George P. Landow
Figurski at Findhorn	Richard Holeton
Genetics	Richard Smyth
I Have Said Nothing	J. Yellowlees Douglas
In Memoriam Web	George P. Landow and Jon
In Small & Large Pieces	Kathryn Cramer
Lust	Mary-Kim Arnold
Mahasukha Halo	Richard Gess
Notes Toward Absolute	Tim McLaughlin
Patchwork Girl	Shelley Jackson
Quam Artem Exerc eas?	Giuliano Franco, M.D.
Quibbling	Carolyn Guyer
Samplers	Deena Larsen
Sea Island	Edward Falco
Socrates in the Labyrinth ⁶	David Kolb
True North	Stephanie Strickland
Turning In	Wes Chapman
Twilight, A Symphony	Michael Joyce
Unnatural Habitats	Kathy Mac
Unreal City	Christiane Paul
Victory Garden	Stuart Moulthrop
We Descend	Bill Bly
WOE	Michael Joyce
Writing at the Edge ⁶	George P. Landow, ed.

⁶ In published titles that include several notionally separate hypertexts, the primary (and largest) hypertext alone was examined here, unless otherwise indicated.

14. REFERENCES

1. Akscyn, R., McCracken, D. and Yoder, E., KMS: A Distributed Hypermedia Systems for Managing Knowledge in Organizations. in Hypertext '87, (Chapel Hill, NC, 1987), ACM.
2. Atkinson, B. HyperCard, Apple Computer Co., Cupertino CA, 1987.
3. Beck, K. Extreme Programming Explained: Embrace Change. Addison Wesley Longman, Reading MA, 1999.
4. Bernstein, M. "The Bookmark and the Compass: Orientation Tools for Hypertext Users". SIGOIS Journal, 9 (1988). 34-45.
5. Bernstein, M., Card Shark and Thespi: exotic tools for hypertext narrative. in Hypertext 2001: Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, (Århus, Denmark, 2001), ACM, 41-50.
6. Bernstein, M., Patterns of Hypertext. in Hypertext '98, (Pittsburgh, PA, 1998), ACM, 21-29.
7. Bernstein, M. "Storyspace: Hypertext and the Process of Writing". in Berk, E. and Devlin, J. eds. Hypertext/ Hypermedia Handbook, McGraw-Hill, New York, 1991, 529-533.
8. Bernstein, M., Bolter, J.D., Joyce, M. and Mylonas, E., Architectures for Volatile Hypertext. in Hypertext'91, (San Antonio, 1991), ACM, 243-260.
9. Bernstein, M. and Thorsen, L., Developing Dynamic Documents: Special Challenges for Technical Communicators. in 34th International Technical Communications Conference, (Denver, 1987).
10. Bolter, J.D. and Joyce, M., Hypertext and Creative Writing. in Hypertext '87, (Chapel Hill, 1987), ACM, 41-50.
11. Brown, P.J. "Do we need maps to navigate round hypertext documents?" Electronic Publishing —Organization, Dissemination and Design, 2 (2). 91-100.
12. Conklin, J. "Hypertext: An Introduction and Survey". IEEE Computer, 1987 (September). 17-41.
13. Davis, H.C., Referential Integrity of Links in Open Hypermedia Systems. in The Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, Hypertext 98, (Pittsburgh, PA, 1998), ACM, 207-216.
14. Falco, E. A Dream With Demons, Eastgate Systems, Inc., Watertown, Massachusetts, 1997.
15. Greco, D. Cyborg: Engineering The Body Electric, Eastgate Systems, Inc, Watertown, MA, 1994.
16. Halasz, F., McCracken, D., Meyrowitz, N. and Shneiderman, B., Confessions—What's Wrong With Our Systems. in Hypertext '89, (Pittsburgh, PA, 1989), ACM, 399-400.
17. Halasz, F.G., P.Moran, T. and Trigg, R.H., NoteCards in a Nutshell. in CHI+GI Proceedings, (1987), ACM, 45-52.
18. Harpold, T., Thickening: Reading in the Field of the GUI. in Digital Arts and Culture, (Providence, RI, 2001).
19. Johns, A. The Nature of the Book: Print and Knowledge In The Making. University of Chicago Press, Chicago, 1999.
20. Joyce, M. afternoon, a story, Eastgate Systems, Inc., Watertown, MA, 1990.
21. Joyce, M. "Nonce Upon Some Times: Rereading Hypertext Fiction". Modern Fiction Studies, 43 (3). 579-597.
22. Joyce, M. Siren Shapes: Exploratory and Constructive Hypertext Academic Computing, 1988, 11 ff.
23. Joyce, M., Storyspace as a hypertext system for writers and readers of varying ability. in Hypertext'91, (San Antonio, 1991), 381- 387.
24. Kilgour, F.G. The evolution of the book. Oxford University Press, New York, 1998.
25. Kirschenbaum, M. Lines for a Virtual t/o/o/pography English, University of Virginia, Charlottesville, 1997.
26. Kolb, D., Scholarly Hypertext: Self-Represented Complexity. in Hypertext 97, (Southampton, U.K., 1997), ACM, 29-37.
27. Landow, G.P. Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology, 2nd edn. Johns Hopkins Press, Baltimore, 1997.
28. Landow, G.P. "Popular Fallacies about Hypertext". in Mandl, D.J.J.a.H. ed. Designing Hypertext/Hypermedia for Learning, Springer-Verlag, Heidelberg, 1990.
29. Landow, G.P. Writing At The Edge, Eastgate Systems, Inc., Watertown, Massachusetts, 1995.
30. Landow, G.P. and Kahn, P. Where's the Hypertext? The Dickens Web as a System-Independent Hypertext ECHT'92, ACM Press, Milano, 1992, 149-160.
31. Larsen, D. Samplers: Nine Vicious Little Hypertexts, Eastgate Systems, Inc., Watertown Massachusetts, 1998.

32. Marshall, C., Halasz, F.G., Rogers, R.A. and Janssen, W.C. Aquanet: A Hypertext Tool to Hold Your Knowledge in Place Hypertext'91, San Antonio, 1991, 261-275.
33. Marshall, C.C. and Irish, P.M. Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext Intelligible for Readers Hypertext'89, Pittsburgh, 1989, 15-26.
34. Miles, A. "Cinematic Paradigms for Hypertext". Continuum: Journal of Media & Cultural Studies, 13 (2). 217-225.
35. Millard, D.M., Moreau, L., Davis, H.C. and Reich, S., FOHM: A Fundamental Open Hypertext Model for Investigating Interopability Between Hypertext Domains. in Hypertext 2000, (San Antonio, Texas, 2000), ACM, 93-102.
36. Moulthrop, S. Victory Garden, Eastgate Systems, Inc., Watertown, MA, 1991.
37. Nanard, J. and Nanard, M., Should link anchors be typed too? An experiment with MacWeb. in Hypertext '93, (Seattle, Washington, 1993), ACM.
38. Nelson, T. Literary Machines. Mindscape Press, 1987, 1982.
39. Nuernberg, P.J., Leggett, J.J. and Schneider, E.R. As We Should Have Thought Proc. of Hypertext'97, Southampton, UK, 1997, 98-101.
40. Rau, A., Towards the Recogniition of the Shell as a Integral Part of the Digital Text. in Hypertext 99, (Darmstadt, Germany, 1999), ACM, 119-120.
41. Reich, S., Wiil, U.K., Nürnberg, P., Davis, H., Grønbaek, K., Anderson, K., Millard, D.E. and Haake, J.M. "Adressing interoperability in open hypermedia: the design of the open hypermedia protocol". New Review of Hypermedia and Multimedia, 5. 207-246.
42. Smith, K.E. and Zdonik, S.B., Intermedia: A case Study of the Differences Between Relational and Object-Oriented Database Systems. in OOPSLA 87, (Orlando, FL, 1987), ACM SIGPLAN Notices, 452-165.
43. Taylor, P.G. "Hypertext-based art education: Implications for liberatory learning in high school." doctoral dissertation, Penn State University, 1999. See <http://www.eastgate.com/storyspace/art/Taylor1.html>
44. Tosca, S.P., A Pragmatics of Links. in Hypertext 2000 Proceedings, (San Antonio, Texas, 2000), ACM, 77-84.
45. Walker, J., Piecing together and tearing apart: finding the story in afternoon. in Hypertext '99, (Darmstadt, Germany, 1999), ACM, 111-118.
46. Yankelovich, N., Meyrowitz, N. and Dam, A.v. "Reading and Writing the electronic book". IEEE Computer (Oct. 1985).