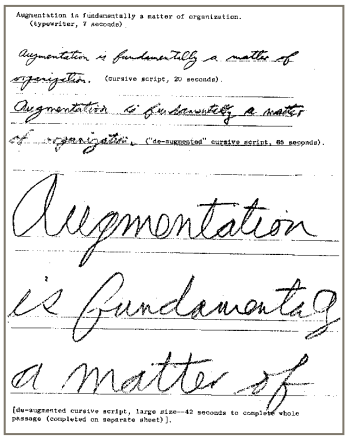
Tripping Up Memory Lane

Chris Nuzum, Co-Founder & CTO, Traction Software

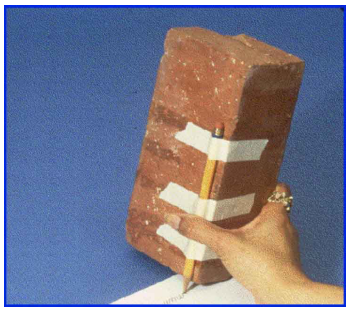
Hyperkult 2015

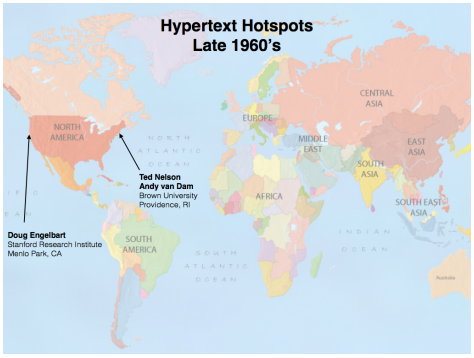
Hi. My name is Chris Nuzum. It's nice to be here with you all today here in Lüneburg, Germany, and to be part of the last Hyperkult conference.

I'm a hypertext practitioner. And that's the perspective I'm coming from in this talk. I'm fortunate to have spent most of my career in hypertext. This was no accident, but rather a by-product of my proximity to the brain trust that my mentor Andy van Dam cultivated at Brown University, where I studied Computer Science in the 80's.



Anyway, in May, I was back at Brown to celebrate the 50th anniversary of Computer Science—and Andy—at Brown. Andy led one of the two pioneering teams building hypertext systems in the 60's.



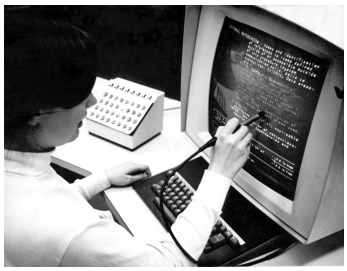


Hypertext Hotspots in the 1960's

West Coast—Doug Engelbart, De-augmentation, Mother of All Demos, circa 1968

The other was, of course, Doug Engelbart, whose work you all know well. While Doug was working on the 1968 Fall Joint Computer Conference "Mother of All Demos", Andy's students, many of whom were undergraduates, were building the Hypertext Editing System, HES, which,

With Andy van Dam, May 2015



East Coast—Andy van Dam & HESS, circa 1968

among other applications, was used by NASA for Apollo mission documentation.

As an undergraduate, my hypertext experience was limited. A few brief encounters with Hypercard, and a humanities class that used Intermedia, a hypertext system built at Brown's Institute for Research in Information Scholarship, as a teaching aid.

When I graduated, I started working at Electronic Book Technologies, with a great group of people, including Greg Lloyd, a veteran of Andy's HES system, and Steve DeRose, a veteran of its successor FRESS, the File Retrieval and Editing SyStem. At Electronic Book Technologies, we made arguably the first hypertext *publishing* system. It turned SGML files and stylesheets into electronic books, distributed mainly on CD-ROM. This was DynaText, in homage to the Dynabook, for those of you who know your history.

Electronic Book Technologies' DynaText, circa 1995

It was at Electronic Book Technologies that it dawned on me how much of the history of hypertext was relevant to what we were engineering, and how little of it I was familiar with. With Greg Lloyd as a guide, we stared a hypertext reading group around the time of Hyperkult 5. That was 1995, which was another significant 50th anniversary, that of the publication of Vannevar Bush's *As We May Think* in the Atlantic Monthly.

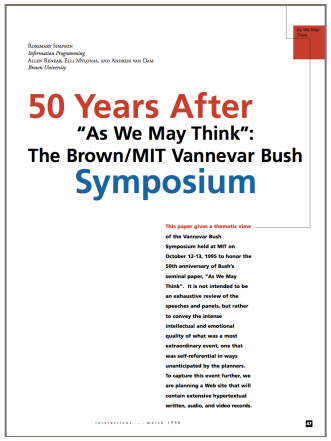

AS WE MAY THINK

A TOP U. S. SCIENTIST FORESEES A POSSIBLE FUTURE WORLD IN WHICH MAN-MADE MACHINES WILL START TO THINK

by VANNEVAR BUSH

The condensed, illustrated version of As We May Think was published in Life magazine, also in 1945

Our own little parallel *Hyperkult* took it upon ourselves to remind Andy of this, which he mentioned in the introduction to the Brown/MIT Vannevar Bush Symposium at MIT a few months later.


50 Years After
"As We May Think":
The Brown/MIT Vannevar Bush
Symposium

Andy and his crew at Brown did an incredible job making the Bush symposium a truly memorable event, featuring Doug Engelbart, Ted Nelson, Alan Kay, and Tim Berners-Lee, among many others. Douglas Adams gave a great after-dinner speech. It was during this event, as I walked along with Doug Engelbart on his way back to his hotel, describing my vision for bringing hypertext to the web to solve the problems of project

managers and other knowledge workers (a term we used a lot in the 90's), that Doug urged me to do something about it. Build it. Start a company. He felt that the problems were important, the ideas worth exploring, and the time right. And in his quiet, kind, reflective way—any of you who remember Doug know just what I mean—he stirred me to action.

A few years later, with an early version of Traction in hand, I visited Doug to walk him through our product and data model, with its many bows to NLS and his ideas.

With Doug Engelbart and an early version of Traction

So, what do I mean by "tripping up memory lane?". Let me start with a few quick words about the title of my talk. First, there's the idea of looking back, or reminiscing, or, as we say, taking a trip *down* memory lane. Lane, meaning street. *Eine Reise in die Vergangenheit, sozusagen.* And then there's "to trip up", or to cause one to make a mistake. And of course, tripping… well, many of you remember the 60's and 70's better than I do.

So, I'd like to talk about a few early hypertext ideas, and how we've used them at my company, Traction Software in our Traction TeamPage hypertext journaling system. This is really an homage to the ideas, sort of a celebration of how useful they are, and just maybe a suggestion that perhaps we haven't collectively leveraged many of them nearly enough.

So, what's this Traction TeamPage thing, and where and how does it fit in the history of hypertext?

If you think back to Doug's Journal, he wrote,

"(1) I felt it important in many dynamic operations to keep a log (sometimes termed a "journal") that chronicles events by means of a ***series of unchangeable entries*** *(for instance, to log significant events while evolving a plan, shaping up a project, trouble-shooting a large operation, or monitoring on-going operations). These entries would be* ***preserved in original form****, serving as the grist for later integration into more organized treatments. (2) I also wanted something that would serve essentially the same* ***recorded-dialogue*** *purpose as I perceived a* ***professional journal*** *(plus library) to do."*[1]

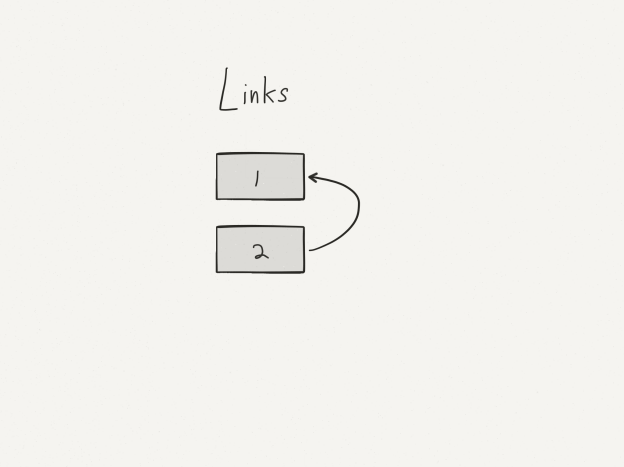
We built on Doug's double-entendre, adding a few more layers of entendre, (though still not in any risqué sense): that of a newspaper—think "Wall Street Journal", for keeping people up-to-date on the latest developments in their organization, using an actual newspaper visual metaphor

— and that of a financial journal, a bit like a double-entry accounting system, providing an audit trail of all transactions.

Links

Anyway, since our subject is hypertext, let's begin with links. Doug Engelbart and Ted Nelson both thought and taught us a lot about links. Hyperlinks need to link *to*—and, ideally, *from*—places that can be identified. In a decentralized system like the web, where those endpoints can change independently over time, we all know what happens:

But 404's really weren't OK for our purposes. We needed a corpus that would be stable over time. Such an append-only journal, like Engelbart's "series of unchangeable entries" allows us to link to any item in the past.

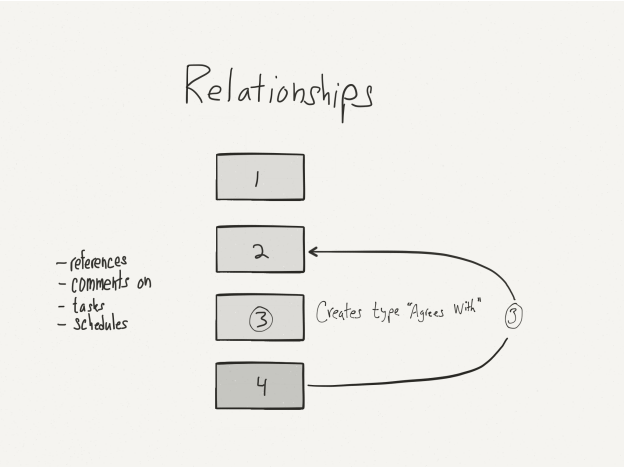


In this picture, the numbers refer both to time and to incrementing persistent identifiers for reach record.

Now we have a nice relationship, where 2 → 1, "two *links to* one". We can also say that "one *is linked to by* two". Naturally, in order to determine the incoming or outgoing links from any entry, we *could* just replay the whole journal, collecting links, but that would be boring. And slow. And wouldn't let us use our Computer Science skills.

So we build data structures and indexes that let us answer common questions really quickly. And the nice thing is that, since we have the journal, we can redesign and reimplement those data structures and indexes, and rebuild them from the journal content. Today people call this an *Event Sourcing* architecture. It has been very useful for us; we have built several generations and varieties of indexes, as well as written tools to transcode the journal into other representations.

What else do we represent with our journal model?



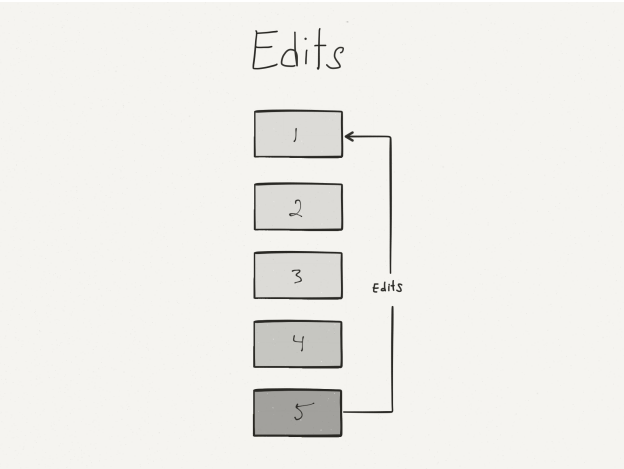
Semantic relationships. This works by creating a new relationship type—this is one of the operations the journal can express—and then applying that relationship type between two entities in the journal.

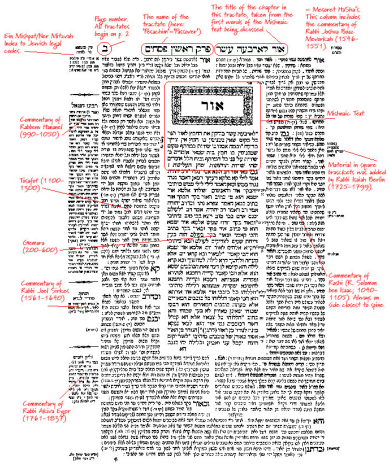
In our original implementation of this, we allowed any entry to express a *ternary* relationship between any two other entries, e.g. "6 could declare that 4 agrees with 2". It turned our that, as immensely powerful as that was, it required too-complicated an interface to explain *where* the relationship came from, so we changed it later to allow any entry to express a typed relationship between *itself* and another entry. So here, entry 4 is declaring that it agrees with entry 2. Still, a few customers really went crazy with semantic relationships in their journals.

We use this typed-link feature a lot. It's how we express a contextual relationships, like when an entry *comments on* another entry (or, in this case, part of another entry). Or when an entry *tasks* another entry, to turn it into a to-do. Or when it *schedules* another entry, to put it on the calendar. Naturally, those fast data structures come in handy when it's time to render an entry, since we are able to decorate it with anything that points to it. We tried to reproduce the talmudic style of contextual annotation, but at the time, it pushed our HTML skills beyond their limits.

Example of a link to a paragraph displayed in the context of that paragraph

There's another kind of semantic relationship that's native to the system. We said earlier that our journal was append-only, and—for the most part—immutable. What happens when we want to edit an entry?

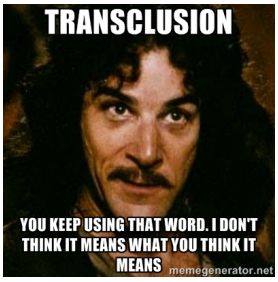




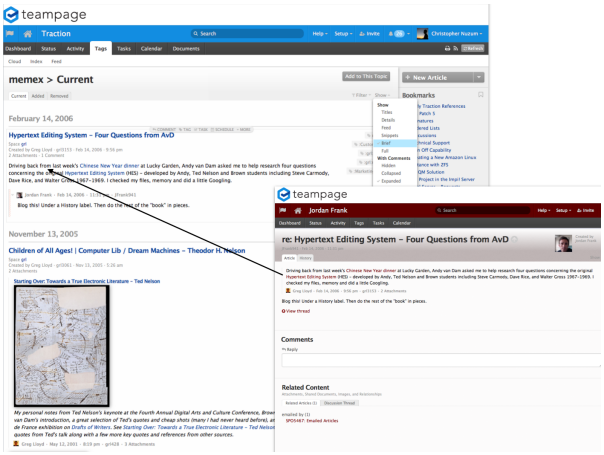
Talmudic annotations

Right, we create a new entry with an "edits" relationship.

The new entry links to the entry that it replaces, with an "edits" relationship, and when the original entry displays itself, instead of showing its own content, it *transcludes* the content of its most recent edit. See why we call it a hypertext journaling system?



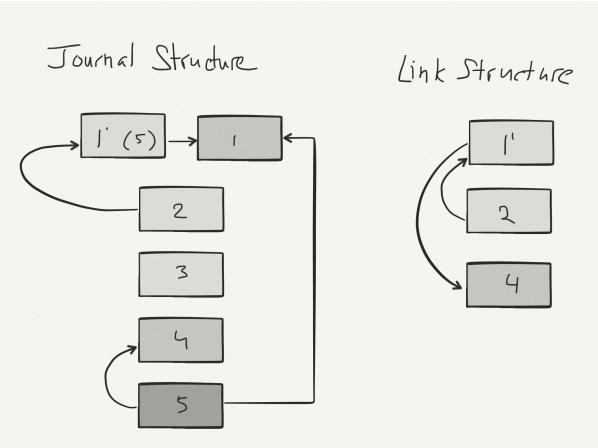
By the way, I like that I can say "transclude" here, and nobody bats an eyelash. We do a lot of transclusion. Any time you type an ID in text to make a cross-reference, we transclude the title of the cross-referenced article. We let you insert widgets that let you transclude any other content in the journal. And we support what we call "proxy entries", which are special entries that are journaled placeholders for external URIs. And when we display them, we can transclude the live content of the external URI. All without a single tumbler…



When viewed by itself, the comment on the right transcludes the context of the paragraph it comments on

Naturally, this behavior might not always be what you want, for example when you're editing, or looking explicitly at an intermediate version, so we followed NLS' example, and we apply a *viewspec* to every view. The viewspec controls many attributes of the view beyond whether the latest version should be transcluded. I'll talk about a few more later on.

But first, another note about links and edits. I mentioned earlier that entries could link to any entry that appears *earlier* in the journal. And now that you see how edits work, you can see how entries can also link to *later* entries.



In this example, entry 5 edits entry 1, effectively creating 1′. And since 5 *can* refer to anything that came before it, 1 (or, really, 1′, which transcludes 5) can now effectively refer to anything that comes *after* it.

It's worth mentioning that we also support wiki-style forward links, to the names of entries —generally the title—that don't exist yet. More on that later, too.

Taxonomy

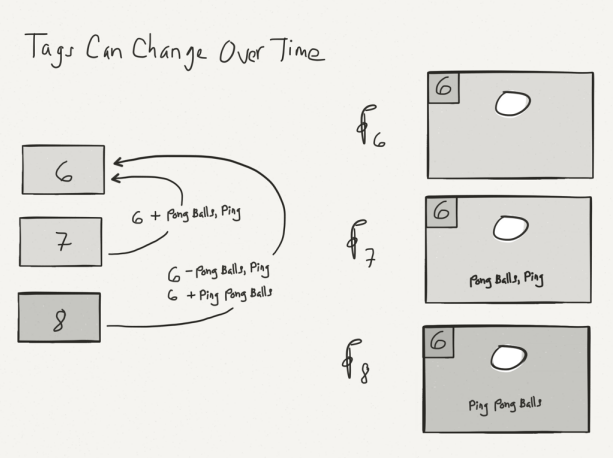
Now, if you're a good Xanadubie, you might be wondering about taxonomy. In Literary Machines 2/49, Ted Nelson quipped that, "The army designation of 'Pong Balls, Ping' has a certain universal character to it"—and made the argument that people should be able to define their own taxonomies over any given corpus.

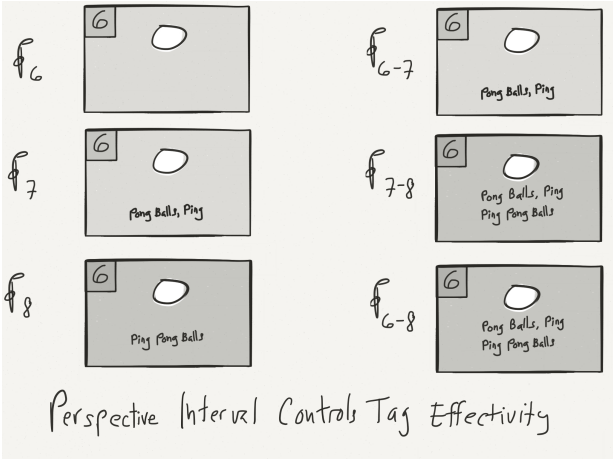
We use tags to represent taxonomy—sort of a *tags*onomy, if you will. And we allow people to have their own spaces, with their own tags, and to apply those tags to any content in any other spaces they can read. Via proxy entries, people can also tag anything they find on the Internet, or in any other URL-addressable system.

We also recognized that "tagsonomies" need to evolve over time, and we wanted to encourage people to change tags fearlessly, without worrying that they'd be losing information. And hey, it's a journaling system. It keeps a trail of everything you do—so why not?

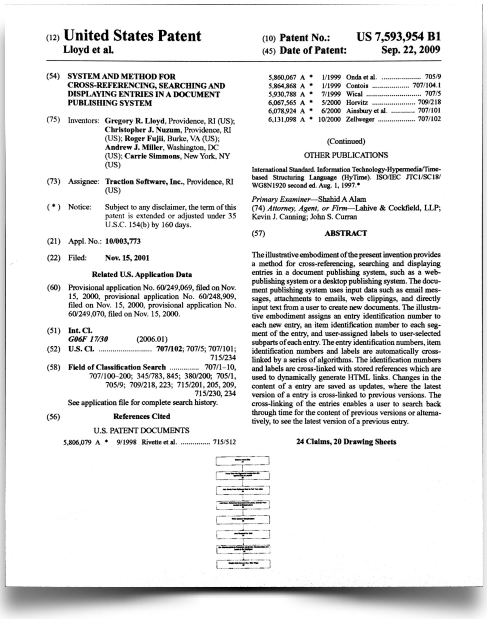
To really encourage fearless "tagsonometric" evolution (we were good Xanadubies—and not *just* for making up our own names for things!), we went a step further, and created the Perspective control, that allowed us to render tag history independent of content history.

Now, since it's a journal, if you just truncate the journal at a given point, you can see exactly what was current at the point of time where you cut it. Perspective lets you control the





tagsonomy independent of the content. In our ping-pong ball example, we can look at the effect of setting perspective to different points and ranges.



(12) **United States Patent**
Lloyd et al.

(10) **Patent No.: US 7,593,954 B1**
(45) **Date of Patent: Sep. 22, 2009**

(54) **SYSTEM AND METHOD FOR CROSS-REFERENCING, SEARCHING AND DISPLAYING ENTRIES IN A DOCUMENT PUBLISHING SYSTEM**

(75) Inventors: **Gregory R. Lloyd**, Providence, RI (US); **Christopher J. Nuzum**, Providence, RI (US); **Roger Fujii**, Burke, VA (US); **Andrew J. Miller**, Washington, DC (US); **Carrie Simmons**, New York, NY (US)

(73) Assignee: **Traction Software, Inc.**, Providence, RI (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 160 days.

(21) Appl. No.: **10/003,773**

(22) Filed: **Nov. 15, 2001**

Related U.S. Application Data

(60) Provisional application No. 60/249,069, filed on Nov. 15, 2000, provisional application No. 60/248,909, filed on Nov. 15, 2000, provisional application No. 60/249,070, filed on Nov. 15, 2000.

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/102**; 707/5; 707/101; 715/234

(58) **Field of Classification Search** 707/1–10, 707/100–200; 345/783, 845; 380/200; 705/1, 705/9; 709/218, 223; 715/201, 205, 200, 715/230, 234

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,806,079 A * 9/1998 Rivette et al. 715/512
5,860,067 A * 1/1999 Onda et al. 705/9
5,864,868 A * 1/1999 Contois 707/104.1
5,930,788 A * 7/1999 Wical 707/5
6,007,585 A * 5/2000 Horvitz 709/218
6,078,924 A * 6/2000 Ainsbury et al. 707/101
6,131,098 A * 10/2000 Zellweger 707/102

(Continued)

OTHER PUBLICATIONS

International Standard. Information Technology-Hypermedia/Time-based Structuring Language (HyTime). ISO/IEC JTC1/SC18/WG8N1920 second ed. Aug. 1, 1997.*

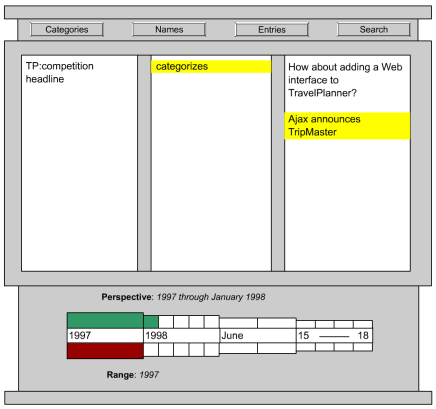
Primary Examiner—Shahid A Alam
(74) *Attorney, Agent, or Firm*—Lahive & Cockfield, LLP; Kevin J. Canning; John S. Curran

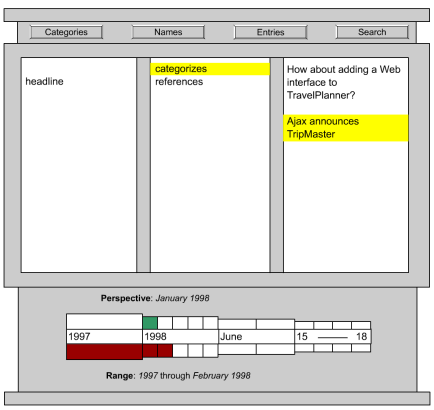
(57) **ABSTRACT**

The illustrative embodiment of the present invention provides a method for cross-referencing, searching and displaying entries in a document publishing system, such as a web-publishing system or a desktop publishing system. The document publishing system uses input data such as email messages, attachments to emails, web clippings, and directly input text from a user to create new documents. The illustrative embodiment assigns an entry identification number to each new entry, an item identification number to each segment of the entry, and user-assigned labels to user-selected subparts of each entry. The entry identification numbers, item identification numbers and labels are automatically cross-linked by a series of algorithms. The identification numbers and labels are cross-linked with stored references which are used to dynamically generate HTML links. Changes in the content of a entry are saved as updates, where the latest version of a entry is cross-linked to previous versions. The cross-linking of the entries enables a user to search back through time for the content of previous versions or alternatively, to see the latest version of a previous entry.

24 Claims, 20 Drawing Sheets

But on reflection, maybe we overreached a bit on this one. It turned out to be, well, a little difficult to explain to customers. We tried various ways to visualize the perspective control, including this Smalltalk browser-like apparatus we called the TrailBrowser.



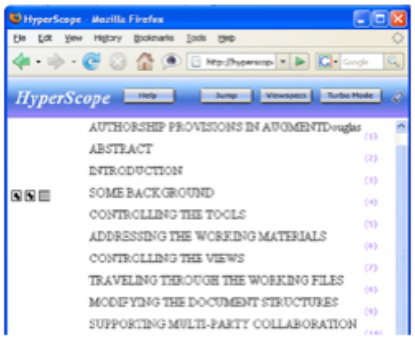


Maybe we were just too early with this feature, like many others. Or maybe we never found its killer app. Or maybe the past isn't as interesting to most people as we thought it would be.

In the current system, we settle for always showing the current perspective, and have removed the perspective control from the interface. However, it's worth mentioning that the tag history and audit trail are still huge benefits of using a journaling system… in addition to making ISO auditors really happy, they eliminate writing status reports saying what you've accomplished.

How so? One of our favorite views is "recently tagged done", which shows the journal entries to which the *done* tag was added. For *my* weekly status report, just look at the articles to which the *done* tag was added this week by *me*. Voila! And it's always up-to-date, and visible to whoever is interested.

There were a lot of other places where we leaned heavily on NLS, Xanadu and other systems, real or imagined.



4

A zoomed out view

5

Zoomed in on SOME BACKGROUND

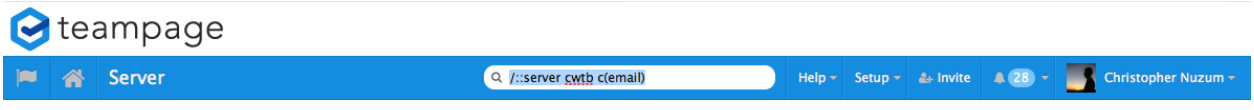
6

Normal view of SOME BACKGROUND

Elided Views

We loved NLS' feature that supported zooming in and out the amount of detail displayed. We built a similar control, which let you toggle between *expanded* and *elided* views. *Expanded* and *elided*. *What* were we thinking? Thank goodness we had the good sense to hire a designer. The renamed *brief* and *full* volume settings remain in the product today. See what I mean about tripping?

Another cue we took from NLS was the inclusion of a command-line interface (disguised as a search box) for people who know their way around the system, and want to get around really quickly. We named ours the *Rapid Selector*, in homage to the device Vannevar Bush was working on as the technical platform of the Memex. The command line supports all sorts of NLS-style controls. It



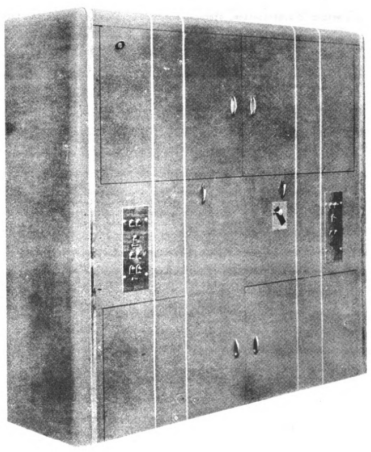
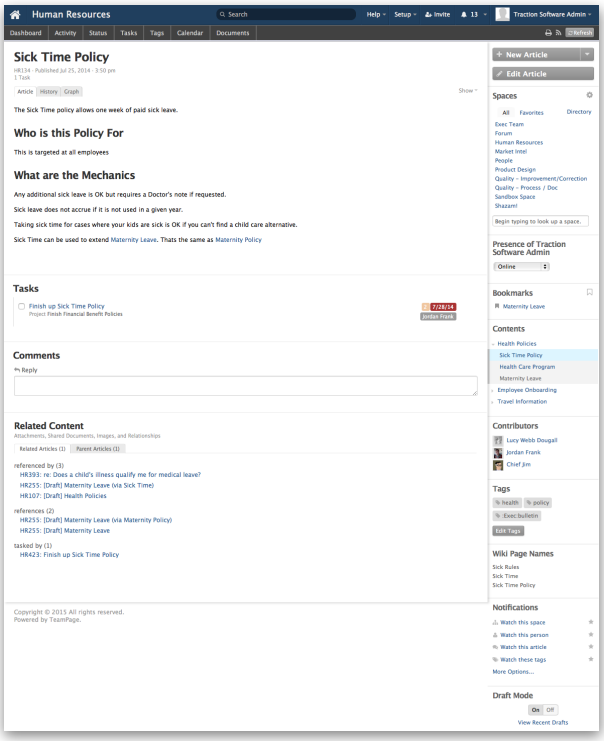


Figure 1. Microfilm Selector, Access Doors Closed As In Normal Operation

lets you change your viewspec. It lets you jump to—or, if you type the commands into an entry, *link to*—any view.

Another of Ted's important concepts from Literary Machines 2/25 is that links stay where you put them, even across edits. We went to great pains to get that right, in an HTML world. And he *was* absolutely right—it's essential. Otherwise you have hyperconfusion. Especially when these link targets are the anchors for comment threads and tasks.

In trying to leverage many ideas from a number of very different systems, we also ran into some unexpected complexity. For example, we support *Draft* and *Published* versions of entries. If we had stuck with only supporting ID-based links (like we talked about above), then displaying and resolving the link targets between entries within the same Draft or Published state, or across states, wouldn't have been so bad—we could have just used a viewspec to control the display of the content of the latest version in the target state. However, since we were also supporting *name-based* wiki linking, and since names could change in a later draft, link title transclusion—and also, more importantly, the link target—also had to change, depending on whether your viewspec said you were reading only Published content, or previewing Draft content. We definitely spent a certain amount of energy squaring circles, especially where our strong permissions model intersected ideas that were conceptualized for systems with relatively



Name-based links can resolve to different targets in Draft and Published views of the world. These are the Draft cross-references.

open permissions. But it's all worked out so far, and we—and our system—are the richer for it. In a metaphysical sense, anyway.

Parting Thoughts

Ultimately, in constructing any system, the set of features that you include—and that you don't include—is a question of curation and design. You try to do what's right for the system, drawing on the best ideas you can. You make sure that they all work together, and that they enhance each other synergistically. And if you know your literature, you'll appreciate that not all those ideas were yours first—and that there may be other profoundly relevant ideas out there, whose time has finally come. Leveraging those ideas and the thinking around them can save you lots of energy rediscovering them.

Of course, feature sets are also governed by timing and resources. Although he worked on related technologies, Bush never tried to build the Memex. His reason for proposing it in the first place was to inspire all the *other* scientists to continue their unprecedented degree of collaboration after the War, and that required a new and worthy challenge.

The challenge worked, and even though I'd argue that nobody has quite delivered on the promise of Bush's Memex vision, during the past *seventy(!)* years, we have accumulated a tremendous number of valuable ideas that trace their roots to it.

As we look forward to curating the feature sets of systems to come, let's be sure not to forget what we've learned, and to remember that just because we're bringing a "new" feature to a new platform, application, or device, that it doesn't mean that somebody hasn't already deeply thought through the ramifications. When I told Doug how much we got out of his papers, he was glad. He said that, "you'd think that nobody ever read them."

We do have giants whose shoulders we can stand on, as long as we have hyperkults who remain aware of them. Long live Hyperkult.

Thanks for listening. See you at coffee.

Chris Nuzum
Co-Founder & CTO
Traction Software
+1 (401) 528-1145
chris@tractionsoftware.com
http://tractionsoftware.com

Further Reading

- Douglas C. Engelbart, AUGMENTING HUMAN INTELLECT: A CONCEPTUAL FRAMEWORK, October 1962, http://www.dougengelbart.org/pubs/augment-3906.html
- Douglas C. Engelbart, The Journal, and Shared-Screen Telephoning, July 29, 1975, http://web.stanford.edu/dept/SUL/library/extra4/sloan/mousesite/Archive/Post68/augment-33076MarkUp.html
- Rosemary Simpson, Allen Renear, Elli Mylonas, and Andries van Dam, 50 Years After "As We May Think": The Brown/MIT Vannevar Bush Symposium, ACM Interactions, Volume 3 Issue 2, March 1996, Pages 47-67, http://dl.acm.org/citation.cfm?id=227187
- Ted Nelson, Literary Machines, 1981, https://en.wikipedia.org/wiki/Literary_Machines
- Andy van Dam, Ted Nelson, Hypertext Editing System, late 1960's, https://en.wikipedia.org/wiki/Hypertext_Editing_System