Deficiencies of the HTML Hypermedia Model

Brendan Sechter

2026-03-07

The HyperText Markup Language, hereafter HTML, is not the only hypermedia model that has ever been designed. It is not the richest, the most consistent, or the most theoretically complete. It is the one that shipped at scale. A generation of practitioners now treats HTML's feature set as the definition of what hypermedia can do, when in fact HTML implements a strict subset of the properties that hypermedia researchers and system designers had already established as desirable by the late 1980s. This article walks the feature axes that historical hypermedia systems addressed, shows where HTML omits or degrades each one, and argues that the omissions were not oversights of a mature model but the deliberate simplifications of a document-delivery format that was later pressed into service as an application platform.

The comparison references six historical systems that treated hypermedia seriously as a first-class model rather than as a byproduct of document markup. Vannevar Bush's Memex, described in As We May Think, is the conceptual ancestor. Douglas Engelbart's oN-Line System, hereafter NLS, and the 1968 demonstration built on the Augmenting Human Intellect framework, produced working versions of most of Bush's properties. Theodor Nelson's Xanadu project, articulated across A File Structure for the Complex, the Changing, and the Indeterminate, Literary Machines, and later restated in Xanalogical Structure, designed a model of transclusion, deep permanent addressing, and micropayment settlement that HTML has never approached. Apple's HyperCard, the Business TRON hypermedia model, hereafter BTRON hypermedia model, from Ken Sakamura's TRON project, and Apple's OpenDoc each addressed specific properties that HTML lacks. TRON is the abbreviation for The Real-time Operating system Nucleus, and BTRON is the business-computing subsystem of the TRON project that hosts the hypermedia model discussed here.

What Every Hypermedia Model Must Address

Every hypermedia design must decide how to handle a set of properties that hypermedia researchers had catalogued by the time of the Dexter Hypertext Reference Model. The properties are not optional in the sense that a system can silently drop them. A system that omits a property has still decided its position on that property, and the decision has consequences for what users can do.

Link directionality. Are links one-way from source to target, or two-way with both ends aware of the relationship?

Link typing. Are links opaque pointers, or do they carry a machine-readable semantic type that describes the relationship?

Sub-document addressability. Can a link target a specific span within a document, or only the document as a whole?

Transclusion. Can a document include a live view of a span from another document, such that the included content remains a first-class reference to the source rather than a copy?

Permanence. Do addresses continue to resolve as the underlying documents evolve, and if they change, are the changes tracked?

Versioning. Does the system record and expose the version history of a document, or does each retrieval return the current state without provenance?

Native composition. Can a document be assembled from parts contributed by different applications, edited in place using the correct tool for each part, and saved back as a single artifact?

Machine-readable structure. Can other programs reason about the document's structure and the semantics of its links without parsing rendered output or scraping visual conventions?

The eight axes are not exhaustive but they cover the material differences between HTML and the historical alternatives.

Bidirectional Links

HTML's links are one-way. A source document names a target by uniform resource locator, hereafter URL. The target has no automatic awareness that the source links to it. Building a reverse index requires an out-of-band crawl by a third party, and the reverse index goes stale immediately.

NLS, Xanadu, and BTRON all treated links as symmetric relationships. In NLS, documented in the primary implementation report by Engelbart and English on the SRI research center, following a link from one document to another produced a visible marker at the destination naming the source. In Xanadu the bidirectional property was foundational, since the addressing scheme placed both endpoints under the same document-space registry. In BTRON the operating system maintained a link database that any node in the system could query in either direction.

The engineering consequence is significant. On the web, the operation "show me every document that links to this one" is a whole-industry problem solved by search-engine companies, whose reverse indices are opaque and unavailable to the site owner or reader. On NLS, Xanadu, and BTRON, the same operation was a local query against a resolved data structure. Wiki systems and static-site generators sometimes reconstruct the property by maintaining a backlink cache, but the cache is a per-site retrofit rather than a system-wide guarantee. The bidirectional-agentic-workflow discipline treated in a prior article is one small-scale reconstruction of the same property inside a documentation system.

Any one-way hypermedia system that reconstructs the bidirectional property with an external crawler is stale by construction. Let λ denote the site-wide rate of new-link creation and T the crawler's revisit interval. The accumulated count of unindexed backlinks at any moment is bounded by the product.

$$N_{\text{stale}} \leq \lambda T$$

The bound is tight in the worst case and cannot be driven to zero by any finite crawl rate. The historical bidirectional systems avoided the bound entirely by maintaining the link database as a first-class primitive rather than reconstructing it after the fact.

Typed Links

HTML links are semantically opaque. The `href` attribute names a target. The `rel` attribute allows a small vocabulary of relationship types drawn from the HTML specification and the microformats community, but the vocabulary is small, inconsistently used, and rarely reasoned about by user-facing software. The link `<a href="paper.pdf">the paper</a>` conveys

no information about whether the relationship is citation, refutation, elaboration, translation, or archival copy.

Xanadu specified typed links from the earliest published proposals. BTRON hypermedia links carried a type label drawn from an application-specified vocabulary. Even the modest efforts of the Resource Description Framework family, hereafter RDF, and JavaScript Object Notation Linked Data, hereafter JSON-LD, are late attempts to bolt onto HTML the machine-readable typing that historical systems built in at the model layer. The bolt-on strategy inherits the difficulties any bolt-on strategy inherits. Adoption is spotty, tooling is uneven, and the primary document format remains untyped.

The markdown-as-specification-language pattern treated in a prior article recovers a small amount of link typing by convention, using reference-style anchors with category prefixes such as `research_` and `related_post_`. This is a per-corpus convention, not a hypermedia model property.

Sub-Document Addressability

HTML supports the fragment identifier, which addresses an element by identifier within a document. Element identifiers are assigned by the document author, and the reader has no ability to address a span that the author did not decorate with an identifier.

Xanadu's tumbler addressing scheme addressed spans by absolute offset into the source document's stable character stream. Any span was addressable, regardless of authorial decoration. The Web Annotation Data Model published by the World Wide Web Consortium, hereafter W3C, addresses spans by selector expressions computed by the annotating client. Adoption of the annotation model has remained thin, and mainstream browsers do not implement it natively.

The consequence is that quoting practices on the web either duplicate content, breaking transclusion, or link to a whole document while relying on the reader to locate the referenced span visually.

Transclusion

HTML has no native transclusion. The `<iframe>` element inlines a foreign document as a rendered viewport but does not participate in the enclosing document's flow or styling. Server-side includes and JavaScript-driven content assembly can simulate transclusion, but the result is a copy at fetch time rather than a live reference to the source.

Xanadu's transclusion was foundational. A document could quote a span from another document, and the quoted span remained a first-class reference to the source, resolvable through the address system, tracked for provenance, and subject to whatever settlement or attribution the source had specified. Transclusion in Xanadu was not simulated by inlining, it was the primary mechanism for building composite documents.

The absence of transclusion in HTML forces the community to reinvent it in tooling. Static-site generators use include directives. Content-management systems use shortcodes. Web components approximate composition. Every reinvention preserves neither provenance nor live reference back to the source.

Native Document Composition

BTRON documents were compositions of typed parts, each edited by the application registered for its type. A document could contain a spreadsheet part, a diagram part, and a text

part, each editable in place using the correct tool. The document was a first-class container that recorded the types and locations of its parts.

Apple's OpenDoc pursued the same model on Macintosh. Microsoft's Object Linking and Embedding, hereafter OLE, is the surviving instance of the pattern in mainstream commercial software. The Portable Document Format, hereafter PDF, embeds foreign parts as opaque attachments.

HTML delegates the composition problem to server-side templating or client-side JavaScript component frameworks. Neither approach produces an artifact that can be edited by the correct in-place tool for each part. The document as a compositional container survives only inside document-editing platforms such as word processors and spreadsheet suites, none of which use HTML as their storage format.

Permanence and Versioning

HTML's addressing scheme is the URL, which resolves whatever the current server chooses to serve. A URL that resolved yesterday may return a different document today, an error tomorrow, and be reassigned by a subsequent site owner. The Internet Archive's Wayback Machine is a third-party retrofit that partially compensates for the absence of permanence at the model layer.

The consequence is measurable. Empirical link-rot studies consistently observe that the fraction of broken outbound links from a published page grows toward unity on a multi-year timescale. The large-scale scholarly-corpus study by Klein and colleagues on reference rot measured the phenomenon across millions of citations and reported multi-year decay rates consistent with exponential decay. If $T_{1/2}$ denotes the observed half-life of a URL's continued resolvability and t denotes elapsed time since publication, the accumulated broken-link fraction follows the form.

$$f_{\text{broken}}(t) = 1 - 2^{-t/T_{1/2}}$$

Observed half-lives in the published literature cluster in the range of a few years, which places a ten-year-old page well past the fifty-percent broken-link threshold. The historical hypermedia models with permanent addressing produced $T_{1/2}$ approaching infinity by design, at the cost of the deployment infrastructure that a permanent-address guarantee requires.

Xanadu specified permanent addresses, whose resolution was guaranteed by the addressing scheme and whose contents were versioned in the document space. The Xanalogical Structure article restated the property clearly against the emerging web's failure to preserve it. Content-addressable storage systems, including the InterPlanetary File System, hereafter IPFS, approximate the property by hashing document contents, but adoption remains niche and the hashes address content rather than semantic identity.

Versioning of documents on the web is either absent, kept by the publisher as opaque server-side state, or reconstructed by third-party archival services. The reader has no in-band way to ask "what did this document say last week" without depending on infrastructure external to the hypermedia model.

Machine-Readable Structure

HTML mixes structural markup, presentational markup, and application behavior in the same document tree. Elements such as `<article>`, `<section>`, and `<nav>` carry structural meaning, but their use is optional and inconsistent. The primary consumers of the structure are search-engine crawlers, screen readers, and syndication tools, each of which uses a different subset

of the vocabulary and each of which retrofits interpretation over inconsistent authoring practice.

NLS treated documents as trees with explicit structural nodes. Xanadu treated documents as segments in a stable stream with explicit link and quotation records. BTRON treated documents as typed part collections with per-part structural conventions. In each system the structure was mandatory, not decorative.

The Resource Description Framework family of specifications and JSON-LD are the W3C-sanctioned retrofits. Their uptake in ordinary documents remains marginal. The Semantic Web project, articulated by Berners-Lee, Hendler, and Lassila as a program for machine-reasonable web content, was in essence an admission that HTML's structure layer was inadequate for machine reasoning and required a parallel graph representation to be bolted on.

Why HTML Displaced the Alternatives

The historical systems had richer models. HTML shipped. The reasons are worth stating in the same terms so that the shipping outcome is not confused with technical superiority.

HTML was radically simpler than any of the alternatives. A minimal HTML browser was implementable by a competent engineer in weeks. A minimal Xanadu implementation required a global addressing infrastructure and a settlement mechanism. A minimal BTRON implementation required an operating system built to support the hypermedia model as a first-class concern.

HTML was implementable over the existing internet stack without requiring any protocol beyond the Hypertext Transfer Protocol, hereafter HTTP. Xanadu, NLS, and BTRON each specified their own addressing and transport machinery, which required deployment infrastructure that no organization was in a position to provide at scale.

HTML permitted broken links, dangling references, and inconsistent addressing. The permissiveness was a defect against the historical model requirements. It was also the property that let the web grow without central coordination. A hypermedia model that required consistency would have required a central coordinator that no participant would have accepted.

HTML delegated typing, versioning, and composition to informal community conventions rather than baking them into the format. The delegation permitted rapid experimentation and permitted the eventual accretion of standards on top. It also permitted the eventual accretion of ad-hoc replacements for the missing features, none of which recover the historical model's properties in full.

Partial Recoveries

Several communities have reconstructed subsets of the historical properties on top of HTML. The reconstructions are informative because they show which properties remain load-bearing and which are optional in practice.

Wiki systems reconstruct bidirectional links inside a single wiki instance by maintaining a backlink table. The reconstruction does not extend across wiki instances or across the wider web.

WebMentions and Pingback reconstruct cross-site backlinks by voluntary notification. Adoption is thin outside the IndieWeb community.

Static-site generators reconstruct typed cross-references by convention and reconstruct transclusion by include directives at build time. The reconstructions are per-tool.

JSON-LD and schema.org vocabularies reconstruct machine-readable typing for specific commercial use cases such as recipe cards, event listings, and product markup. General-purpose semantic markup remains rare.

Content-addressable storage systems reconstruct permanent addressing by hashing content. The addressing property is preserved, the semantic identity property is not.

Web Components reconstruct native composition inside a single origin's JavaScript environment. Cross-origin composition remains a security surface rather than a supported model.

Reading through the reconstructions in aggregate reveals that the historical hypermedia research programme identified the properties correctly. The engineering community has spent thirty years reinventing subsets of the same properties on top of an inadequate substrate.

Prior Art in the Comparative Literature

The comparative critique of HTML's model against its historical predecessors is not new. Frank Halasz's Reflections on NoteCards catalogued seven properties that hypermedia systems circa 1988 struggled with, all of which appear in modern form on the current web. Norman Meyrowitz's Missing Link keynote argued that the emerging community was already discarding hard-won properties. Ted Nelson's Xanalogical Structure restated the case in 1999, after the web had become the dominant deployment. The From Memex to Hypertext collection documents the historical thread from Bush through the systems that attempted to realize his vision.

None of the historical critiques stopped HTML's dominance. They do serve as inventories of properties that HTML omits and that reappear as engineering problems whenever a project tries to build serious knowledge infrastructure on top of it. The knowledge-graph pattern discussed in a prior article is one contemporary instance of the same recurring engineering problem, treated at the corpus level with the same conventions the LLM tooling community rediscovered independently.

Conclusion

HTML implements a subset of the properties that hypermedia systems circa 1990 had already established as necessary for serious document infrastructure. The omissions are load-bearing. Bidirectional links, typed links, sub-document addressability, transclusion, permanence, versioning, native document composition, and machine-readable structure are each addressed by historical systems and each either omitted or degraded by HTML.

The historical systems did not ship at web scale, for reasons of implementation complexity and infrastructure requirements rather than for reasons of model inadequacy. HTML shipped because it was simple enough to implement in weeks, tolerant enough of inconsistency to grow without coordination, and delivered over an existing protocol stack. The shipping outcome was decisive but the model comparison remains sharp. Every serious knowledge-infrastructure project on top of HTML rediscovers some subset of the historical properties as an engineering problem, and reconstructs a substrate-specific approximation of the property that the historical models specified as a system requirement.

Understanding HTML as a minimum-viable subset of an already-mature hypermedia research programme, rather than as the definition of what hypermedia can do, is a corrective that lets contemporary designers borrow from the historical vocabulary rather than reinvent it under new names.

References

- Berners-Lee, Tim, Weaving the Web, HarperOne, 1999
- Nelson, Theodor H., Computer Lib / Dream Machines, 1974
- Nelson, Theodor H., Literary Machines, 1981
- Nyce, James M. and Kahn, Paul, editors, From Memex to Hypertext, Vannevar Bush and the Mind's Machine, Academic Press, 1991
- Berners-Lee, Tim, Information Management, A Proposal, CERN, 1989
- Engelbart, Douglas C., Augmenting Human Intellect, A Conceptual Framework, SRI Research Report, 1962
- Sakamura, Ken, The Objectives of the TRON Project, 1987
- W3C, Web Annotation Data Model
- W3C, RDF 1.1 Concepts and Abstract Syntax
- WHATWG, HTML Living Standard
- Related Post, Bidirectional Agentic Workflow
- Related Post, LLM Knowledge Graphs
- Related Post, Markdown as a Specification Language for Agentic Workflows
- Berners-Lee, Tim, Hendler, James, and Lassila, Ora, The Semantic Web, Scientific American 284, 2001
- Bush, Vannevar, As We May Think, Atlantic Monthly 176, 1945
- Engelbart, Douglas C. and English, William K., A Research Center for Augmenting Human Intellect, AFIPS Fall Joint Computer Conference 33, 1968
- Halasz, Frank G., Reflections on NoteCards, Seven Issues for the Next Generation of Hypermedia Systems, Communications of the ACM 31, 1988
- Halasz, Frank G. and Schwartz, Mayer, The Dexter Hypertext Reference Model, Communications of the ACM 37, 1994
- Klein, Martin, Van de Sompel, Herbert, Sanderson, Robert, Shankar, Harihar, Balakireva, Lyudmila, Zhou, Ke, and Tobin, Richard, Scholarly Context Not Found, One in Five Articles Suffers from Reference Rot, PLoS ONE 9, 2014
- Meyrowitz, Norman, The Missing Link, Why We're All Doing Hypermedia Wrong, Hypertext 89 Proceedings, 1989
- Nelson, Theodor H., A File Structure for the Complex, the Changing, and the Indeterminate, ACM 20th National Conference, 1965
- Nelson, Theodor H., Xanalogical Structure, Needed Now More Than Ever, ACM Computing Surveys 31, 1999