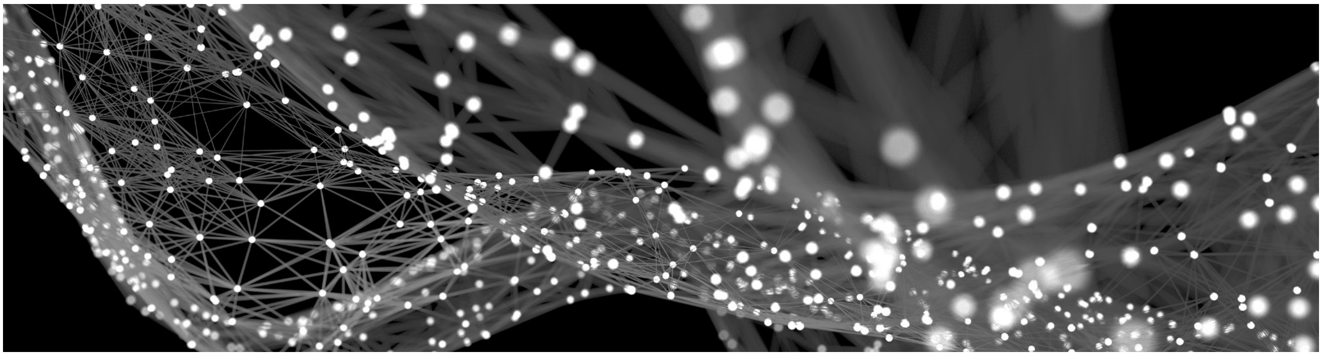
The SAGE Handbook of
Web History



Edited by

Niels Brügger and
Ian Milligan

SAGE reference
Los Angeles | London | New Delhi | Singapore | Washington DC | Melbourne

Contents

Hypertext before the Web – or, What the Web Could Have Been

Belinda Barnet

Historiography is always guided by specific tropes; it is 'infected by what it touches as the past' Demeulenaere (2003). Much has been written about the history of hypertext over the last 20 years, and I've contributed to that literature; in the process I've been infected by the vision behind the early systems from the 60s and 70s. This might be because I've got to know some of the inventors – I conducted 22 interviews with inventors for my book on pre-web systems *Memory Machines* (Barnet, 2013) – and drank the same Kool-Aid. The first system was built in counterculture-soaked 1960s California, though, so a bit of dreaming is appropriate, along with the occasional Yoga Workstation. This was an era when people had grand visions for their pre-web hypertext systems, when they believed that the solution to the world's problems might lie in finding a way to organize the mess of human knowledge: to represent its true interconnections.

Now for the important bit: this story stops before the Web. More accurately, it stops before hypertext became synonymous with the Web; as Ted Nelson[1], who coined the term 'hypertext' among other things, put it to me in 1999, 'People saw the Web and they thought, "Oh, that's hypertext, that's how it's meant to look"'. But hypertext is not the Web; the Web is one particular implementation of hypertext. In this chapter we will use Ted Nelson's definition of hypertext: branching and responding text read at a computer screen (Nelson, 1992). The Web is without a doubt the most successful and prevalent version of hypertext, but it is also an arguably limited one. Hypertext existed well before the Web – the systems here were imagined (and in some cases built and obsolete) well before Google and Facebook. Insofar as the Web is a hypertext system, it continues an already established line of evolution, and it is important to understand this pre-history to fully understand the Web.

The first hypertext systems were deep and richly connected, and in some respects more powerful than the Web. These early systems were not, however, connected to hundreds of millions of other users. You could not reach out through FRESS and read a page hosted in Thailand or Libya. The early systems worked on their own set of documents in their own unique environments. Although Nelson certainly envisioned that Xanadu would have the domestic penetration of the Web, and NLS had nifty collaborative tools and chalk-passing protocols, none of the early 'built' systems we look at either briefly or in depth in this chapter – NLS, HES and FRESS – was designed to accommodate literally billions of users. That's something only the Web can do.

As Jay Bolter put it in our interview,

> What the World Wide Web did was two things. One is that it compromised as it were on the 'vision' of hypertext. It said, 'This is the kind of linkage it's always going to be, it's always going to work in this way', [but] more importantly it said that the really interesting things happen when your links can cross from one computer to another… So global hypertext – which is what the Web is – turned out to be the way that you could really engage, well, ultimately hundreds of millions of users. (Bolter, 2011)

The goal of this chapter is to explore the visions of the early hypertext pioneers, starting over 60 years ago, and in the process, to broaden our conception of what hypertext could be. The chapter begins by exploring NLS, then Xanadu and on to HES.

DOUG ENGELBART'S ON-LINE SYSTEM (NLS)

Dr Douglas Carl Engelbart, who died in 2013, was a softly spoken man. His voice was low yet persuasive, as though 'his words have been attenuated by layers of meditation', his friend Nilo Lindgren wrote in 1971 (Rheingold, 2000: 178). I struggled to hear him in our interview, being deaf myself, but that didn't matter; he had been describing the same vision in great detail to journalists, historians and engineers for over 60 years. Engelbart wanted to improve the model of the human, to 'boost our capacity to deal with complexity' as a species (Engelbart, 1999).

> Here's a human… He's got all these capabilities within his skin we can make use of, a lot of mental capabilities we know of, and some of it he's even conscious of. Those are marvellous machines there – motor machinery to actuate the outside world, and sensor and perceptual machinery to get the idea of what's going on… (Engelbart, 1998: 213)

For Engelbart, the most important element of the 'human system' was language; language is a powerful machine. Engelbart would seek to harness its nonlinear relationships with a computer system and externalize its 'networked' structure.

Engelbart still remembers reading about Vannevar Bush's Memex (hypothetical proto-hypertext system that Bush described in his 1945 *The Atlantic Monthly* article 'As We May Think'), and the moment when he became 'infected' with the idea of building a means to extend and navigate this great pool of human knowledge. He was a military radar technician out in the Philippines when he first picked up a reprint of Bush's article, in the summer of 1945, and wandered into a Red Cross library that was built up on stilts to read it. News of Hiroshima was devastatingly fresh; Engelbart was 20 years old and deep in thought (Barnet, 2013). He was wondering how he could 'maximize [his] contribution to humankind' as an engineer (Engelbart, 1999).

In this article Bush looked towards the postwar world as an engineer and predicted an exponential increase in human knowledge, especially scientific and technological knowledge. How are we to keep track of it all? How are we to prevent great ideas from being lost? Some ideas are like seeds. 'Or viruses. If they are in the air at the right time, they will infect exactly those people who are most susceptible to putting their lives in the idea's service' (Rheingold, 2000: 176). Although he didn't think about the article again for many years, the ideas in it infected Engelbart.

But five years after he had read 'As We May Think' in the Philippines, Engelbart claims, 'I formulated [my] goal on…human intellectual effectiveness' (Engelbart, 1962a: 236). These three 'flashes' were to become the framework he worked from for the rest of his career:

1 FLASH-1: The difficulty of mankind's problems was increasing at a greater rate than our ability to cope.
2 FLASH-2: Boosting mankind's ability to cope with complex, urgent problems would be an attractive candidate as an arena in which a young person might 'make the most difference'.
3 FLASH-3: Ahah – graphic vision surges forth of me sitting at a large CRT console, working in ways that are rapidly evolving in front of my eyes (beginning from memories of the radar-screen consoles I used to service). (Engelbart, 1988: 189)

This vision of a radar-screen console attached to a computer is important. Engelbart transferred this technology from the radars he was servicing in the Philippines to computers as he had learnt about them in engineering school (CRT phosphors came into common use around WWII). This image of a human sitting at a screen is the image from which all future work would depart. By the time Engelbart started work on NLS ten years later, progress had been made on 'presenting computer-stored information to the human… by which a cathode-ray-tube (of which the television picture tube is a familiar example) can be made to present symbols on their screens of quite good brightness, clarity, and with considerable freedom as to the form of the symbol' (Engelbart, 1962b). In 1951, however, Engelbart had to mentally extrapolate from radar screens. As he told me in 1999,

> I put together what I knew about computers and what I knew about radar circuitry etc. to picture working interactively, and it just grew from there' (Engelbart, 1999).

This was a radical idea in the 1950s. At that time, computers were large electronic devices stored in air-conditioned rooms, at many degrees of separation from the 'user'. They were attended to by technicians and fed their information out to these technicians on punch cards and printouts. The idea that a screen might be attached to a computer, and that humans might interact directly via this surface, was far left field.

The computer, screen and mouse would become Engelbart's parallel to Memex's microfilm storage desk, tablet display and stylus. With these technologies, Engelbart would 'update' an image of potentiality from a different era and bring it to digital computing. But his ideas did not garner the peer support Engelbart was seeking.

> After I'd given a talk at Stanford, [three angry guys] got me later outside at a table. They said, 'All you're talking about is information retrieval.' I said no. They said, 'YES, it is, we're professionals and we know, so we're telling you, you don't know enough so stay out of it, 'cause goddamit, you're bolloxing it all up. You're in engineering, not information retrieval.' (Engelbart, 1999)

Computers, in large part, were still seen as number crunchers, and computer engineers had no business talking about the human beings who used these machines. Fortunately, one of the few people who had the disciplinary background to be able to understand the new conceptual framework was moving through the ranks at the Advanced Research Projects Agency (ARPA). This man was J.C.R. Licklider, a psychologist from MIT. 'The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and the resulting partnership will think as no human brain has ever thought', wrote Licklider in a 1960 paper called 'Man-Computer Symbiosis' (Licklider, 1960: 131).

Licklider began financing projects that developed thought-amplifying technologies. ARPA support began in 1963, 'at varying levels – during 1965, about eighty thousand dollars' (Bardini, 2000: 23). But as Bardini points out (in his book and in a personal communication), it was Bob Taylor, initially working as a psychologist at NASA, who mustered the strongest support for the project. Initially he put in 85,000 dollars from NASA mid 1964 to mid 1965 (Bardini, 2000: 23). Then when he moved from NASA to ARPA in 1964 he told Engelbart that IPTO 'was prepared to contribute a million dollars initially to provide one of the new time-sharing systems, and about half a million dollars a year to support the augmentation research' (Rheingold, 2000: 86). That is the equivalent of around 14 million in today's dollars.

Around this time Engelbart asked a bright young SRI engineer named Bill English if he'd present a paper for him. English said yes, and joined Engelbart's project shortly thereafter. As English told me in 2011, 'I saw what he was doing and I was interested in it. So I joined him and that's how it all began' (English, 2011). He became Engelbart's chief engineer in 1964, and began work on some of the basic ideas we'll explore in the next section.

Bill Duvall joined SRI in 1966, but didn't start work on the NLS project until 1968. He asked to be moved onto Engelbart's team, but first he had to meet with the head of engineering at SRI.

> [He] was a very traditional engineer. I was a 24-year-old kid, or 23 years old, and he had this big corner office that had bookshelves to the ceiling and the little ladder that goes around, and a big desk. He sat on the other side of his desk, and I sat in a chair and he looked at me, and he said, 'You don't really think what they're doing up there is science, do you?' I think that reflected a lot of the official attitude towards what Doug was doing. (Duvall, 2011)

The freshly outfitted laboratory, the Augmentation Research Center (ARC), began its work in 1965. It started with a series of experiments focused on the way people select objects on a computer screen. In the context of screen-based interactivity, Engelbart and English's 'mouse' consistently beat other devices for fast, accurate selection in a series of controlled tests. Yet it took over 20 years to enter the commercial market, a time period that Engelbart considers strikingly long (Engelbart, 1988: 196).

At the heart of NLS was a basic philosophy that you should be able to link to anything, anywhere ('unless there was a specific reason not to do so' (Duvall, 2011)). This approach to information was entirely new – an approach that assumed from the outset that connectivity is important, that the relationship between and among ideas was just as important as the unit of information (or 'statement') itself. As Duvall remembers,

> The thing that I would say distinguished NLS from a lot of other development projects was that it was sort of the first – I'm not sure what the right word is – 'holistic' is almost a word that comes to mind – project that tried to use computers to deal with documents in a two-dimensional fashion rather than in a one-dimensional fashion. (Duvall, 2011)

Content insertion and navigation involved four basic commands: Insert, Delete, Move and Copy. The mouse served as a pointer to indicate where content was to be inserted or deleted in existing text. Most important, however, the Link function allowed cross-referencing to another statement – and the user could define cross-references at any level in the hierarchy. Links were character strings within a statement 'indicating a relationship' (Engelbart, 1999) to another statement, and they could be made in the same file or between different files.

The NLS design changed as it was created, evolving around the technical activities of the project team itself as they documented their efforts. In the process of designing the software, the team generated a number of technical reports, source code versions, communications, release notes, problems and associated solutions. These were linked together and tracked by date and version.

Consequently, between 1969 and 1971 (Bardini, 2000) NLS was changed to include an electronic filing arrangement that served as a linked archive of the development team's efforts. This eventually cross-referenced over 100,000 items (Engelbart, 1988). It was called the software Journal, the most explicit model of a hypertextual environment with embedded associative links to surface in the digital environment in the 1960s.

As mentioned previously, the entire NLS system was set up for perfect storage and recall. Everything was tracked and identifiable. Every object in the document was intrinsically addressable, and most important, these addresses never disappeared (Engelbart, 1999). They were permanent, attached to the object itself, which meant that they followed the object wherever it was stored, so links could be made at any stage to any object in the system. As Duvall remembers,

> [In NLS] we had it that every object in the document was intrinsically addressable, right from the word 'go'. It didn't matter what date a document's development was, you could give somebody a link right into anything, so you could actually have things that point right to a character or a word or something. (Duvall, 2011)

The NLS team called this a 'frozen state' addressing scheme (Engelbart, 1997), which is in contrast to the World Wide Web, where the finest level of intrinsic addressability is the URL (Universal Resource Locator, a character string that identifies an Internet resource, invented by Tim Berners-Lee in 1994[2]). Unlike the NLS object-specific address, the URL is simply a location on a server; it is not attached to the object itself. That said, NLS was working in its own little environment, on its own documents – not with billions of users.

By 1968 NLS had matured into a massive database and set of paths through this database, the first digital hypertext system. It was time to take NLS out of the Petri dish and set it to work in front of the engineering community. Engelbart took an immense risk and applied for a special session at the ACM/IEEE-CS Fall Joint Computer Conference in San Francisco in December 1968. 'The nice people at ARPA and NASA, who were funding us, effectively had to say "Don't tell me!", because if this had flopped, we would have gotten in trouble' (Engelbart, 1988: 203).

The conference was set up using a video projector pointing at a huge, 20-foot screen that was linked to a host computer and piped back to the group at SRI via a temporary microwave antenna. The setup was very expensive, and although no special system capabilities were employed (NLS was run just as it was used back at the lab), the organizational and presentational machinery used almost all the remaining research funds for the year.

Engelbart sat up on the stage beneath the projection screen, a mouse in one hand and his other hand playing a special one-handed keyset. He manipulated the audience's attention by controlling their view of the information being explored; he drilled down through the data structure and presented it in multiple different views, each piece connected to the last by a link. The screen was divided into neat windows containing explanatory text or graphics about NLS, and also about the presentation itself in the hypertext Journal. It was dubbed 'The Mother of all Demos' by Andries van Dam. A video of this demo is now available on the Web.

This was the first public appearance of the mouse and the first public appearance of hypertext, screen splitting, computer-supported associative linking, computer conferencing and a mixed text/graphics interface. It proceeded without a hitch and received a standing ovation (Rheingold, 2000).

As a result of the NLS demo, many of the user interface technologies from NLS also migrated into computing over the next few years, the mouse in particular. Ted Nelson eventually incorporated the mouse into his Xanadu design (Barnet, 2013). NLS also allowed consideration of the modern windows-icon-menu-pointing-device (WIMP) interface.

Engelbart was working from a vision he had as a young engineer. This vision changed the world. Technical visions have no essence; there is no transcendent design, no Platonic form we are striving towards. There is, however, a recurrent dream – an elusive 'blessed break' – and this dream is an ancient one. It comes from long-standing cultural desires and anxieties about the ephemerality of human memory and knowledge.

The dream is to create a perfect archive for human knowledge, a machine that might 'extend, through replication, human mental experience' (Nyce and Kahn, 1991: 124) and capture the interconnected structure of knowledge itself. Most important, this would be a machine that we can control, whose workings are transparent to us and whose trails do not fade. Which brings us to the next system – my favorite system and recurring vision: the Magical Place of Literary Memory: Xanadu®.

THE MAGICAL PLACE OF LITERARY MEMORY: XANADU

> What I thought would be called Xanadu® is called the World Wide Web and works differently, but has the same penetration. (Nelson, 1999)

It was a vision in a dream. A computer filing system that would store and deliver the great body of human literature, in all its historical versions and with all its messy interconnections, acknowledging authorship, ownership, quotation and linkage. Like the Web, but much better: no links would ever be broken, no documents would ever be lost, and copyright and ownership would be scrupulously preserved. The Magical Place of Literary Memory: Xanadu. In this place, users would be able to mark and annotate any document, see and intercompare versions of documents side by side, follow visible hyperlinks from both ends ('two-way links') and reuse content pieces that stay connected to their original source document. There would be multiple ways to view all this on a computer screen, but the canonical view would be side-by-side parallel strips with visible connections. Just imagine. This vision – which is older than the Web, and aspects of it are older than *personal computing* – belongs to hypertext pioneer Theodor Holm Nelson, who dubbed the project Xanadu in October 1966.[3]

The name comes from the famous poem by Samuel Taylor Coleridge, 'Kubla Khan'. In his tale of the poem's origin, Coleridge claimed to have woken from a laudanum-laced reverie with 'two or three hundred' lines of poetry in his head. He had noted down but a few lines when he was interrupted by a visitor, and when he returned to his work later he found that the memories had blurred irretrievably. His mythical landscape, this vision of Xanadu, had passed away 'like the images on the surface of a stream into which a stone had been cast' (Coleridge, cited in Nelson, 1987: 142).

One of Nelson's most cherished memories is based on a vision of gently moving water. When he was about four or five he trailed his hand in the water as his grandfather rowed a boat. He studied the different 'places' in the water as they passed through his fingers, the 'places that at one instant were next to each other, then separated as my finger passed. They rejoined, but no longer in the same way' (Nelson, 2010: 35). These connections were infinite in number and in complexity, and they changed as the water moved. It was a religious experience, a vision that has stayed with him for 70 years. At that moment he started thinking about what he calls 'profuse connection', the interconnections that permeate life and thought: How can one manage all the changing relationships? How can one represent profuse connection? Xanadu was proposed as a vast digital network to house this corpus of ideas and evidential materials, facilitated by a special linking system.

The story of Xanadu is the greatest image of potentiality in the evolution of hypertext. Nelson invented a new vocabulary to describe his vision, much of which has become integrated into contemporary hypermedia theory and practice – for instance, the words 'hypertext' and 'hypermedia'. As he told me in 1999, 'I think I've put more words in the dictionary than Lewis Carroll. Every significant change in an idea means a new term'. Nelson came up with many significant changes, and consequently many new terms, some of which I discuss below. He also recruited or inspired some of the most visionary programmers and developers in the history of computing, many of whom went on to develop the first hypertext products (although this doesn't impress him: 'the problem with inspiring people is that they then try to credit you with things you don't like' (Nelson, 2010).

Much has been written about Project Xanadu over the years (the ones I cite throughout this chapter are Landow, 1992; Rheingold, 2000). Nelson himself doesn't have the time to keep up with it all – and even when he does, as he put it to me in 2010, 'anything people write about me will be insufficiently praising, and so it's very hard to read it' (his comments on my own work were prolific). The remarkable thing about Xanadu is that, despite countless setbacks, it refuses to die. Its logo is, appropriately enough, the Eternal Flaming X. Paisley and Butler (cited in Smith, 1991: 262) have noted that 'scientists and technologists are guided by "images of potentiality" – the untested theories, unanswered questions and unbuilt devices that they view as their agenda for five years, ten years, and longer'. Often accused of hand waving and lucid dreaming, Nelson's Xanadu has nonetheless become the most important vision in the history of computing.

Nelson wears a lanyard across his neck and shoulder with three pens attached to it (when I first met him in 1999 it had sticky notes, at another point a stapler – the 'system is evolving' as he put it in a personal communication, 2012). He has been wearing it since 1997. The belt is filled with tools to connect things with, tools to deal with a world of paper. Like Bush, Nelson is painfully aware that ideas are easily lost in conventional indexing systems, that they are disconnected from each other, and that 'serious writing or research' demands connecting ideas together. Frustrated by the lack of a global, real-world system that might do this for him, and 'outraged' by the confines of paper (Nelson, 1998: 1–2), he feels the need to do this manually.

Nelson's struggle against the paradigm of paper led him to design an alternative. In 1960, at Harvard University, he took a computer course for the humanities, 'and my world exploded' (Nelson, 2010: 99). He understood immediately that computers were all-purpose machines that could be put in the service of information handling. Like Engelbart, he did not believe that computers were mathematical tools for engineers. He saw a solution to his problem of information handling, and he also saw a future where paper might be eliminated. 'The prison of paper, enforcing sequence and rectangularity, had been the enemy of authors and editors for thousands of years; now at last we could break free' (Nelson, 2010: 120).

One of the first ideas was based on his own 'terrible problem' keeping notes on file cards. The problem was that his cards really needed to be in several different places at once; new projects were built on earlier ideas, new documents were built on earlier ideas, so these items should be reusable by reference. Perhaps the computer might solve that problem; it could link them together. This idea would later prove important in the design of Xanadu.

> Many of my file cards belonged in several places at once – several different sequences or projects. Each card – call it now an entry or an item – should be stored only once. Then each project or sequence would be a list of those items. (Nelson, 2010: 103)

Imagine if the same item could appear in multiple places. You could connect each item to its original by an addressing method and retrieve them on a computer screen. Because this would be reuse by reference rather than by copying, you could trace each item back to its original source. This idea – that you should be able to see all the contexts of reuse, and that you should be able to trace items back to their original source – would 'drive my work to the edge of madness' (Nelson, 2010: 104). It would later become the kernel of Nelson's most innovative idea: transclusion. Transclusion, and also the ability to visually compare prior or alternative versions of the same document on-screen ('intercomparison') were integral to the design of Xanadu. Over the next 40 years, Nelson would hone these ideas and experiment with them in various incarnations.

In 1960 Nelson announced his term project: a writing system for the IBM 7090, the only computer at Harvard at the time, stored in a big, air-conditioned room at the Smithsonian Observatory. In the 1960s computers were 'possessed only by huge organizations to be used for corporate tasks or intricate scientific calculations' (Nelson, 1965: 135). The idea that expensive processing time might be wasted on writing, of all things, was deemed crazy by the engineering community. Nelson ignored this. He proposed a machine-language program to store documents in the computer, change them on-screen with various editorial operations and print them out. But this was no mere word processor (which in any case didn't exist at the time); Nelson envisioned the user would be able to compare alternative and prior versions of the same document on-screen.

The second part of Nelson's idea took shape in the early 1960s, when there was 'a lot of talk around Cambridge [Massachusetts, where Harvard is located] about Computer-Assisted Instruction, for which there was a lot of money' (Nelson, 1992: 1/26). It was not so much a design at this point, Nelson stressed in response to my request for paper, 'it was an idea that may have been on only one file card' (Nelson, 2010). At this time he conceived of what he called 'the thousand theories program', an explorable computer-assisted instruction program that would allow the user to study different theories and subjects by taking different trajectories through a network of information.

This led to another idea, which Nelson drafted as an academic paper while teaching sociology at Vassar College in 1965. He wants to stress at this point that there was no single eureka moment as 'the ever-changing designs had been swirling in my head for five years' (Nelson, 2012). The concept was clear enough, however, to put it down on paper. This revised design combined two key ideas: side-by-side intercomparison and the reuse of elements (transclusion). He thought about the architecture of the system and decided to have sequences of information that could be linked sideways. As with his first design, this would all occur on a computer screen, visually, in real time. He called this system 'Zippered Lists'.

Zippered Lists permitted linking between documents: like the teeth in a zipper, items in one sequence could become part of another ('EXCEPT', Nelson wrote in response to this chapter, 'the two sides of the zipper don't have to be in the same order'). Versions of a document could be intercompared; an item could be an important heading in one sequence and a trivial point in another, and all items could be written or retrieved in a nonsequential fashion. Links could be made between large sections, small sections or single paragraphs. Writers could trace the evolution of an idea.

Crucially, the design also got him published. Nelson's first paper explaining the term 'hypertext' was presented at the ACM 20th National Conference in August 1965. It was not the first time that the word 'hypertext' had appeared in print, though. That was in an invitation to a talk Nelson gave at Vassar College, 'Computers, Creativity and the Nature of the Written Word', on 5 January 1965. (Actually, it's the word 'hypertexts'. A copy of this invitation appears in *Possiplex*.)

One of the first people who thought they might try to build *part* of Nelson's design was Andries van Dam (I stress 'part' here because van Dam had ideas of his own that he wanted to explore at the same time, such as print text editing). We explore this in the next section.

Van Dam would be the first person to attempt to build part of Nelson's vision.

SEEING AND MAKING CONNECTIONS: HES AND FRESS

We will now trace the development of two important hypertext systems built at Brown: the Hypertext Editing System (HES), co-designed by Ted Nelson and van Dam and developed by van Dam's students, and the File Retrieval and Editing System (FRESS), designed by van Dam and his students. Brown University has played a major role in the development of hypertext systems and humanities computing since 1967, due in no small part to van Dam's work in this area, and the Institute for Research in Information and Scholarship (IRIS) he helped establish there with William Shipp and Norman Meyrowitz in 1983.

Ted Nelson was co-designer of HES, and his ideas about hypertext inspired the HES project in the first place; van Dam credited Nelson for this contribution both in our interviews (1999; 2011) and in his public talks and published work. Nelson still feels, however, that he has been 'given no more credit than his [van Dam's] undergraduate students' (Nelson, 2012). Unfortunately, Nelson fell out with the HES team during its design and implementation in 1967, and he has stated in several interviews with the author that he was unhappy with the result. He is bitter about the experience, largely because he feels his vision was sidelined in favor of print text editing (Nelson, 1999; 2011). We explore this in more detail shortly.

Along with Engelbart's landmark NLS system, HES and FRESS constitute the first generation of hypertext systems. These are not dusty old antiques from the dawn of computing science, however; in many respects, FRESS by 1968 was more interactive than present-day HTML. As working prototype systems, HES and FRESS had technical chutzpah.

The Design of HES

Van Dam bumped into Nelson at the 1967 Spring Joint Computer Conference in Atlantic City. Passionate and eloquent, Nelson told van Dam about what he'd been doing since he left Swarthmore: hypertext. 'He had nothing to show for this idea, no prototypes or work in the sense that computer scientists talk about work – i.e. software, algorithms, things that are concrete', recalled van Dam (1999). What Nelson did have was a vision of what hypertext should look like, and an infectious enthusiasm for the idea. Nelson had in mind an entirely new genre for literature, and he had a new word to describe this vision: hypertext.

> Nelson's vision seduced me. I really loved his way of thinking about writing, editing and annotating as a scholarly activity, and putting tools together to support that… He talked me into working on the world's first hypertext system and that sounded cool. (van Dam, 1999)

Van Dam gathered a team together at Brown and began work later that year, with the objective of trying out this hypertext concept. He stressed in his communications to me that the idea was never to 'realize' Xanadu in its entirety. The intention was much smaller and more circumspect: to 'implement a *part* of his vision. We were not able to implement or even understand the breadth and scope of that [larger] vision' (van Dam, 2011). Nelson, however, was under the impression that his designs would be honored.

Nelson was initially very excited. He went up there 'at his own expense' (van Dam, 1988: 87) to consult in the development of HES, but found the experience frustrating. As observed previously, van Dam and his team wanted to explore the hypertext concept, but they also had their own plans for print text editing (which Nelson strenuously opposed). The team set out to design a dual-purpose system for authoring, editing and printing documents such as papers, proposals and course notes, which could also be used to browse and query written materials nonsequentially. From Nelson's perspective, this was a compromise on his design: it was simulating paper, and hypertext was a mere footnote to the system.

This sentiment is understandable; the reader knows by now that Nelson feels his vision was sidelined. But the HES team were trying to convince the world that the whole concept of handling text on computers was not a waste of time and processing power. The world knew text handling as a paper-based thing. 'Not only were we selling hypertext, but at the same time document processing, interaction. Many people were still computing with cards', recalls van Dam (1999).

HES was set up on an IBM 360/50 with a 2250 display, and ran in a 128k partition of the operating system that controlled the 512k of main memory available (there was a complete times-haring system operating in another partition). The user sat facing a 12-inch-by-12-inch screen, browsing through portions of arbitrarily sized texts. Original text was entered directly via a keyboard, and the system itself was controlled by pressing function keys and by pointing at the text with a light pen or with the keyboard (Carmody et al., 1969: 4). The activities of the user corresponded directly to the operations normally performed upon text by writers and editors. The user was able to manipulate pieces of text as though they were physical items: correcting, cutting, pasting, copying, moving and filing drafts.

The HES team did not wish to store text in numerical pages or divisions known to users, except as they might deliberately divide text, create links or number headings. Rather than filing by page number or formal code name, HES stored text as arbitrary-length fragments or 'strings' and allowed for edits with arbitrary-length scope (for example, insert, delete, move, copy). This approach differed from NLS, which imposed a hierarchical tree structure of fixed-length lines or statements upon all content; Engelbart used 4,000-character limits on his statements to create a tighter, more controlled environment. HES was deliberately made to embody a freewheeling character, as non-structured as possible.

The system itself comprised text 'areas' that were of any length, expanding and contracting automatically to accommodate material. These areas were connected in two ways: by links and by branches. A link went from a point of departure in one area (signified by an asterisk) to an entrance point in another, or the same, area.

The HES team used Ted Nelson's concept of a hypertext link (though from Nelson's perspective they 'flattened' this by making the jumps one-way), as Doug Engelbart was incorporating the same idea into NLS independently, unbeknownst to van Dam, who wishes he was aware of this work. 'I hadn't heard of Engelbart. I hadn't heard of Bush and Memex. That came quite a bit later', van Dam recalls (1999). Links were intended to be optional paths within a body of text – from one place to another. They represented a relationship between two ideas or points: an intuitive concept. Branches were inserted at decision points to allow users to choose 'next places' to go.

In early 1968 HES did the rounds of a number of large customers for IBM equipment, for example, *Time/Life* and *The New York Times*. All these customers based their business on the printed word, but HES was too far out for them. Writing was not something you did at a computer screen. They had seen programs that set type, and maybe some programs for managing advertisements, but the concept of sitting in front of a computer and writing or navigating text was foreign to them.

> The best I ever got was from people like Time-Life and the New York Times who said this is terrific technology, but we're not going to get journalists typing on computer keyboards for the foreseeable future. (van Dam, 1999)

As we now know, however, in less than a decade journalists (and executives) would be typing on computer keyboards.

In late 1968 van Dam finally met Doug Engelbart and attended a demonstration of NLS at the Fall Joint Computer Conference. As we explored, this presentation was a landmark in the history of computing, and the audience, comprising several thousand engineers and scientists, witnessed innovations such as the use of hypertext, the computer 'mouse' and screen and telecollaboration on shared files via videoconferencing for the first time.

For van Dam this system set another, and entirely different, technical precedent.

The line or context editor was old technology. NLS was the prototype for creating, navigating and storing information behind a tube and for having a multiuser, multiterminal, cost-effective system. He went on to design the File Retrieval and Editing System (FRESS) at Brown with his team.

Nelson is adamant that the legacy of HES is modern word processing, and that it also led to today's web browser.

> The design of HES became the design of FRESS… then Intermedia, then imitated by Notecards and then by the World Wide Web. (Nelson, 2012)

Van Dam, for his part, thinks this is overreaching. 'I think Nelson, when he feels that the "bad example" that HES set had reverberations in the bad design of the Web or browsers, is giving our humble little effort an order of magnitude more credit than it deserves' (van Dam, 2011).

We do not have space here to go into van Dam's next system (FRESS): interested readers can find it in my book. I shall stop here, 40 years before the Web, and conclude.

CONCLUSIONS

I hope that this chapter has presented you with some earlier models of the hypertext concept, and in the process, demonstrated that every model has its benefits and its shortcomings. Hypertext is not the Web; the Web is but one particular implementation of hypertext. It's the best we've come up with insofar as it actually works, most of the time – and it has stayed the course for 22 years. It is not, however, the only way hypertext can be done – as the systems described in this chapter show.

We have also inherited a vision from these projects: a device that 'enables associative connections that attempt to partially reflect the 'intricate web of trails carried by the cells of the brain'' (Wardrip-Fruin, 2003: 35). More precisely, a tool for thought – a tool that might both organize and 'permanize' (to use Nelson's term again) the mass of deeply tangled data that surrounds us. For the world grows more and more complex every day, and the information we are expected to keep track of proliferates at every click. How are we to keep track of it all?

The problem Nelson, Engelbart and van Dam identified is just as urgent today. The Web has *not* solved this problem for us – arguably it has highlighted it, in razor-sharp text and 16.7 million colors. The Web is without doubt a world-changing and world-opening technology; as van Dam put it to me in 1999, 'the fact that I can reach out and touch stuff in Ethiopia, as it were, is still a surprise to me' (van Dam, 1999). Unlike any of the hypertext systems we have looked at here, the Web stretches between countries and engages literally billions of people. But there are things we could improve on.

Imagine a system whose trails did not fade. Imagine if documents and objects could be stored permanently, with their own unique address that never vanished, and retrieved at will. Imagine if any version of these documents could be visually intercompared side by side, like the teeth in a zipper – and the quotes or ideas in those documents could be traced back to their original source with a click. Imagine if we could separate the linking structure from the content, and that content could consequently be reused in a million different formats. Imagine if we could capture the deeply tangled structure of knowledge itself, but make it better, make it permanent.

Notes

1 Note that his full name is Theodor Holm Nelson.
2 An anchor tag is not an intrinsic address.
3 While working at Harcourt.

REFERENCES

Bardini, T. (2000) *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*. Stanford, CA: Stanford University Press.

Barnet, B. (2013) *Memory Machines: The Evolution of Hypertext*. London & New York, NY: Anthem Press.

Carmody, S., Gross, W., Nelson, T.H., Rice, D. and van Dam, A. (1969) 'A Hypertext Editing System for the /360'. In M. Faiman and J. Nievergelt (eds), *Pertinent Concepts in Computer Graphics*, 291–330. Urbana: University of Illinois Press.

Demeulenaere, A. (2003) 'An Uncanny Thinker: Michel de Certeau' in *Image & Narrative*, 5.

Engelbart, D. (1962a) 'Letter to Vannevar Bush and Program on Human Effectiveness'. In J. Nyce and P. Kahn (eds), *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*, 1991, 235–44. London: Academic Press.

Engelbart, D. (1962b) 'Augmenting Human Intellect: A Conceptual Framework'. *Report to the Director of Information Sciences,* Air Force Office of Scientific Research, Menlo Park, CA, Stanford Research Institute.

Engelbart, D. (1997) Interview with David Bennehum. Online: http://memex.org/meme3-01.html

Engelbart, D. (1998) 'The Augmented Knowledge Workshop'. In A. Goldberg (ed.), *A History of Personal Workstations*, 185–249. New York: ACM Press.

Landow, G. (1992) *Hypertext*. London: Johns Hopkins University Press.

Licklider, J.C.R. (1960) 'Man-Computer Symbioses'. In A. Goldberg (ed.), *A History of Personal Workstations*, 1988, 131–40. New York: ACM Press.

Nelson, T. (1965) 'A File Structure for the Complex, the Changing and the Indeterminate'. In *Proceedings of the ACM 20th National Conference*. New York: ACM Press.

Nelson, T. (1987) *Computer Lib/Dream Machines*. Redmond, WA: Microsoft Press.

Nelson, T. (1992) *Literary Machines*. Sausalito, CA: Mindful Press.

Nelson, T. (1998) *What's on My Mind*. Online: http://www.xanadu.net/xybrap.html

Nelson, T. (2010) Possiplex: Movies, Intellect, Creative Control, My Computer Life and the Fight for Civilization. Available at Lulu.com http://www.lulu.com/spotlight/tandm [March 2012].

Nyce, J. and Kahn, P. (eds) (1991) *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. London: Academic Press.

Rheingold, H. (2000) *Tools for Thought: The History and Future of Mind-Expanding Technology* (2nd Edition). Cambridge, MA: MIT Press.

Smith, L.C. (1991) 'Memex as an Image of Potentiality Revisited'. In J. Nyce and P. Kahn (eds), *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*, 261–86. London: Academic Press.

Tofts, D. (1998) *Memory Trade: A Prehistory of Cyberculture*. North Ryde, NSW: Interface Press.

Van Dam, A. (1988) 'Hypertext '87 Keynote Address' in *Communications of the ACM* Volume 31 Issue 7, July, 887–95.

Wardrip-Fruin, Noah (2003) 'Introduction: As We May Think'. In Noah Wardrip-Fruin and Nick Montfort (eds), 2003, 35.

Wardrip-Fruin, Noah and Montford, Nick (eds) (2003) *The New Media Reader*. Cambridge, MA: MIT Press

PERSONAL COMMUNICATIONS

Bolter, Jay (2011) Interview with the author.

Duvall, Bill (2011) Interview with the author (recorded via phone, Melbourne).

Engelbart, Doug (1999) Interview with the author, Baltimore.

English, Bill (2011) Interview with the author (recorded via phone, Melbourne).

Nelson, Ted (1999) Interview with the author at Kyoto University Japan.

Nelson, Ted (2010) Interview with the author at Swinburne University.

Nelson, Ted (2012) Comments on the manuscript for *Memory Machines*.

Van Dam, Andries (1999) Interview with the author at Brown University.

Van Dam, Andries (2011) Interview with the author via phone.