DHQ: Digital Humanities Quarterly

2010
Volume 4 Number 1

Belinda Barnet <BBarnet_at_groupwise_dot_swin_dot_edu_dot_au>, Lecturer in Media at Swinburne University Melbourne, in association with Smart Services CRC.

Abstract

This article traces the development of two important hypertext systems in the history of computing, and the new paradigms they created: the Hypertext Editing System (HES) and the File Retrieval and Editing System (FRESS). HES was the world's first word processor to run on commercial equipment. It was also the first hypertext system that beginners could use, and pioneered many modern hypertext concepts for personal use. Although the idea of hypertext predates HES and FRESS, this article argues that these two systems were successful because they demonstrated hypertext to a sceptical public; they were both working prototypes.

Introduction: Demo or Die

How does one tell the story of a computer system? As Larry Owens observes in his essay on U.S. engineer Vannevar Bush, “However the historian chooses to answer [this question], utility must certainly play a role” [Owens 1991, 23]. In computer science and engineering, there is an emphasis on prototypes, software, things that are concrete.[1] This applies to engineering discourse as well; one of the main criteria applied to engineered technological solutions is that they work [Childress 1998], or more precisely, that they are *seen* to work. A good working prototype can send shockwaves throughout a research community, and it can inspire a host of new machines in quick succession. This “witnessing” is fundamental to the evolution of technical artefacts; when people see a machine in use, they can visualise the future of the object, connect it with what they are doing in their own work, and connect it with other machines or families of machines. An inventor is always and also a “combinatory” genius, selecting the best technical forms from a number of possibilities, and combining these into a new artefact. What has been witnessed, evaluated and shown to work constitutes the material for transfer. 1

This emphasis on prototypes and demonstrations is not just historically specific to mid-twentieth century engineering discourse; it is also evident in contemporary new media. 2

> Demonstrations have had an important, perhaps even central, place in new media innovation. In some centres of new media, the traditional knowledge-work dictum of “publish or perish” is replaced by “demo or die.” [Wardrip-Fruin & Montfort 2003, 231]

To put it simply: there is a difference between technical vision and technical artefact. As I have argued elsewhere [Barnet 2008], we can dream of time machines and of warp speed — but the reason an engineering prototype has a rapid and often “brutal” productive effect on its milieu is that it displaces technical and financial limits and demonstrates what is possible. Sometimes, it creates new paradigms.

In this article I will be tracing the development of two important prototype systems in the history of hypertext, and the new paradigms they created: the Hypertext Editing System (HES) and the File Retrieval and Editing System (FRESS). HES was the world's first word processor to run on commercial equipment. It was also the first hypertext system that beginners could use, and pioneered many modern hypertext concepts for personal use. FRESS was Van Dam’s second attempt at a hypertext system, and incorporated some ideas from another pioneering hypertext system from the 1960s, Douglas Engelbart’s oN-Line System or NLS. HES and FRESS were developed at Brown University in the late 1960s under the leadership of Andries van Dam, Professor of Computing Science at Brown. They were built at a time when computers, in large part, were still seen primarily as tools for mathematical calculations, and computer engineers had no business talking about text editing or interactivity. In an interview with the author, van Dam recalls that in the 1960s “the whole idea of a user in an interactive computer loop was still foreign to most people” [van Dam 1999]. The story of HES and FRESS is a remarkable one as much for its audacity as its historical influence, and the lack of detailed literature on its development is surprising given this influence. As working prototype systems, they had technical chutzpah. 3

The *idea* of hypertext predates HES and FRESS. The term was coined in 1962 by computer visionary Theodor Holm (Ted) Nelson to describe “nonlinear” text, although the concept of a technical system designed to facilitate nonlinear text can be traced back as far as 1910.[2] Nelson was a co-designer of HES, but was most famous for imagining a hypertext system in the mid-sixties called Xanadu. Xanadu, like its fictional namesake by Samuel Taylor Coleridge, was a vision in a dream, a computer filing system which 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. Nelson began writing about this idea in the sixties, and it has been “under development” for fifty years hence; he has not produced a working prototype or demo of Xanadu, although he did release some shells of code in 1999. To Nelson’s dismay, and unfairly, it has consequently been hailed as “the longest-running vaporware project in the history of computing” [Wolf 1995]. Due in part to this lack of a real-world prototype, Nelson has had difficulty attracting funding or having his work taken seriously. This vision of hypertext did, however, inspire van Dam. 4

Although Nelson was a co-designer of HES, there was a falling-out with the team during its design and implementation, and Nelson stated in an interview with the author that he was unhappy with the result [Nelson 1999]. As we will explore in this article, van Dam and his team were concerned less with grand visions and more with getting the system built and funded in the real world. HES was effectively the first hypertext system available on commercial equipment that novices could use. It introduced interactive reading and authoring to the humanities and the text link to lay users; it established a technical precedent for systems to follow. 5

In this article I will argue that HES and FRESS, like Doug Engelbart’s landmark system NLS, were influential on the history of hypertext and modern computing in large part because they *demonstrated* the concept of hypertext; they were both working systems. Where Ted Nelson, with his prescient ideas and presentations, had been campaigning for “nonlinear writing” housed on a computer for years without commercial success[3], van Dam and his team of 6 mostly undergraduate students actually built it. It was a difficult project precisely because the HES team had first to convince the world that manipulating text interactively on computers was not a crazy idea; convincing them that hypertext was useful was an even harder sell.

Computing in the 1960s

There was a lot going on in the computing industry in the late 1960s. Time-sharing had hit the industry[4], and the minicomputer had begun to establish its markets. Computing had begun to move into the wider arena of "data processing." Noting such changes, IBM began marketing to a full circle of customers instead of the traditional military and industrial customers. System/360, the mainframe on which HES was eventually built, was the first computer built by IBM aimed at businesses and corporations — a wider and more "domestic" market. There was a computing paradigm-shift going on. 7

But although businesses were adopting "data processing" for the first time, the world still saw computers as hardwired mathematics. "Text was not 'data'; it was something academics and journalists manipulated with pens or typewriters," van Dam recalls [van Dam 1999]. Processing power was limited, and universities meted it out like gold to the places it was needed most — in their opinion, physics and engineering departments. Computers were "for scientists and engineers to solve serious problems," van Dam recalls his then-VP for Finance saying, and he was told "if you want to crank out papers you can damn well use a typewriter" [van Dam 1999]. This vision of a human sitting at a computer screen and writing or navigating linked text was not obvious to the university. As Dr Douglas Engelbart, the inventor of the computer mouse, observed in an interview with the author, "The idea was wacky even in the seventies, when we had it working — real hypermedia, real groupware working" [Engelbart 1999]. 8

There was, however, an existing technology used for editing text on a computer; it was called a "line" or "context" editor [van Dam 1999]. This was designed for writing or editing computer programs, but it was often used covertly to create documents. Like Doug Engelbart, who unbeknownst to van Dam was working on the NLS/Augment system at the same time[5], van Dam saw the need to tailor computing technology to the humans that used it. In particular, he saw that an editing system which sufficed for computer programmers would not fulfil the needs of writers. Then he encountered Ted Nelson. 9

The Design of HES and FRESS

van Dam bumped into Nelson at the 1967 Spring Joint Computer Conference.[6] Passionate and eloquent, Nelson started telling him about hypertext. Nelson had been thinking about hypertext for several years and had tried to design a system at Vassar College in 1965 [Nelson 1999], but unfortunately like most of Nelson's ideas, it had not eventuated in a product of any kind. When van Dam met Nelson, "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" [van Dam 1999]. What Nelson did have was a vision of what hypertext might one day look like, and an infectious enthusiasm for the idea. Nelson envisioned a computer-based hypertext system where the user could take different trajectories through a network of information, reading and writing "non-sequentially" [Nelson 1999]. 10

> 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. I hadn't heard of Engelbart. I hadn't heard of Bush and Memex. So after meeting quite by accident at this computer conference, and talking about what we each were doing, we somehow got onto the topic. I had this wonderful graphics display...and I was working on various graphics applications at the time. He talked me into working on the world's first hypertext system and that sounded cool. [van Dam 1999]

van Dam didn't need much convincing. He had been thinking about computers and writing for some time now. So he gathered a team together at Brown and began work later that year, with the objective of trying out this hypertext concept.

Nelson went up there "at his own expense" [van Dam 1988, 87] to consult in its development, but found the experience quite frustrating. van Dam and his team set out to design a dual-purpose system for authoring, editing and printing text 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 sidelined the concept of hypertext in favour of print text editing. In an interview with the author, Nelson claimed what the HES team built was essentially a word processor with linking facilities; it was directed at the printed page [Nelson 1999]. Nelson was not happy with this "trivialisation" of his hypertext vision[7], and spent much time arguing for the elaboration of HES's hypertext facilities. For Nelson, paper was the enemy. 11

But the HES team was trying to convince the world that the whole concept of handling text on computers was not a waste of time and processing power. And the world knew text handling as a paper-based thing. "So not only were we selling hypertext, but at the same time document processing, interaction. Many people were still computing with cards," recalls van Dam [van Dam 1999]. As stated previously, even Doug Engelbart was having trouble convincing the world that humans and machines might work together interactively. There was not a funding body in the country that would support the idea of a completely non-sequential, online "web." Perhaps more importantly, van Dam already had a vision of what his writing system should do, a vision based on improving an existing human activity: creating and editing text for printed documents. 12

The HES team designed a system for the composition and manipulation of manuscripts, which could also be used as a reading machine to browse and query complex written materials online. They did not wish to inflict line numbers on the user, or to "make him program little changes in his data" [Carmody et al. 1967, 8] as line editors for computer programs did. Reading, writing and editing documents should be simple, and the tools should be suited to the task. Because they wished to emphasise the place of "ordinary" text editing as well as the more radical concept of non-sequential, online reading and writing, they called the project the Hypertext *Editing* System — a name meant to embrace this dual purpose. 13

It was a hard sell. van Dam recalls his chairman at the time saying, "Why don't you stop with all this hypertext nonsense, and do something serious?" to which van Dam replied, "Walter, I am doing something serious" [van Dam 1999]. At his own university, van Dam "had multiple reactions, from 'cute' to 'I don't get it' to my being in a pissing contest to be allowed to use the university's only mainframe computer for this research with the then-VP for Finance" [van Dam 1999]. After being shown a first version of the system, IBM, however, thought the project serious enough to provide funding through a research contract. This, recalls van Dam, put the project on much more legitimate ground and ensured that the undergraduates who had been programming HES as a bootleg graphics project were then paid for their efforts [van Dam 1988, 889]. 14

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 timesharing system operating in another partition). The user sat facing a 12" by 12" 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, by pointing at the text with a 15 light pen or via the keyboard [Carmody et al. 1967, 4]. The activities of the user were 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, moving, cutting, pasting, filing previous drafts and copying.

> Our philosophical position [was] essentially that the writer is engaged in very complicated pursuits, and that this work legitimately has a freewheeling character... therefore it became our intent to provide the user with unrestricted “spatial” options, and not to bother him with arbitrary concerns that have no meaning in terms of the work being performed. [Carmody et al. 1967, 8]

The HES team also did not wish to store text in numerical pages or divisions known to the user, except as she might deliberately divide her text, create links or number her 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 is in contrast to 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. These limitations meant that Engelbart could implement more efficiently [van Dam 1999]. HES was deliberately made to embody a freewheeling character, as non-structured as possible.

The system itself was composed of text “areas” which 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. Although the HES team used Ted Nelson's conception of a text link, Doug Engelbart was incorporating the same idea into NLS independently, unbeknownst to van Dam, who wishes he had known about 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.[8] Links were intended to be optional paths within a body of text — from one place to another. A link was intended to express a relationship between two ideas or points: an intuitive concept. 16

The HES team employed “human factors” techniques to design a system which guided and explained the user's every move without loss of effectiveness [Carmody et al. 1967, 14]. The system was much easier to use than NLS, perhaps due to the fact that it was created as much for writers as it was for engineers. Speaking from the year 2009, the editing features of HES seem quite obvious, almost trivial. At the time, however, it was difficult to convince people in a commercial environment that it was not an excessively complicated typewriter. Nobody could conceive of writing or editing text on a screen. As van Dam recalled in an interview with the author, 17

> Remember, we were doing hypertext at a time when there were no word-processing systems either. The HES is one of the very first word processors… I believe HES was the first document editor, specifically designed for documents, to run on commercial equipment, and NLS was the first document editor to run on a proprietary system, predating HES. [van Dam 1999]

One of Van Dam’s tasks was to convince humanities scholars and writers that this was a more efficient way of writing than using a pen or a typewriter. HES used a standard 32-key IBM function keyboard, but when they gave demos to writers in the late 1960s whose business was words, not engineering, they would “freak out” over all the buttons [van Dam 1999]. So van Dam learned to use progressive disclosure, not frightening humanities people with too many buttons. 18

> I made a plastic overlay which I essentially used to cover up all but five of the editing buttons: insert, delete, move, copy and jump. Then we would do an entire demo for half an hour or so with that … and then we would play peek-a-boo, strip off the first overlay, and lo and behold, there was another row of function keys. [van Dam 1988, 890]

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. 19

> The best I ever got was from people like Time and 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. This was a landmark presentation in the history of computing, and the audience, comprised of several thousand engineers and scientists, witnessed such innovations as the use of hypertext, the computer “mouse” and screen, and and telecollaboration on shared files via video conferencing for the first time. The unveiling of NLS is now known affectionately and with great respect as the Mother of All Demos, and was celebrated in December of 2008 at its 40th anniversary with almost all of the original team led by Doug Engelbart on stage, and Andries van Dam as the outsider commentator. 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 multi-user, multi-terminal cost-effective system. He went on to design the File Retrieval and Editing System (FRESS) at Brown with his team of hotshot undergraduates and one masters student. As van Dam observed in the Hypertext 1987 conference keynote address, 20

> ...my design goal was to steal or improve on the best ideas from Doug's NLS and put in some things we really liked from the Hypertext Editing System - a more freeform editing style, no limits to statement size, for example. [van Dam 1988]

The HES project was frozen as the team started work on the next-generation system. Meanwhile, IBM sold HES to the Apollo Mission Team at the Houston Manned Spacecraft Centre (unbeknownst to Nelson and van Dam and others who had worked on it at the time). van Dam now proudly recalls that it was used in NASA's Houston Manned Spacecraft Center for documentation on the Apollo space program [van Dam 1988]. As with most computing projects in the late 60s, government (and the military) had a keener eye for innovation than business. For what it was designed to achieve, HES performed perfectly. 21

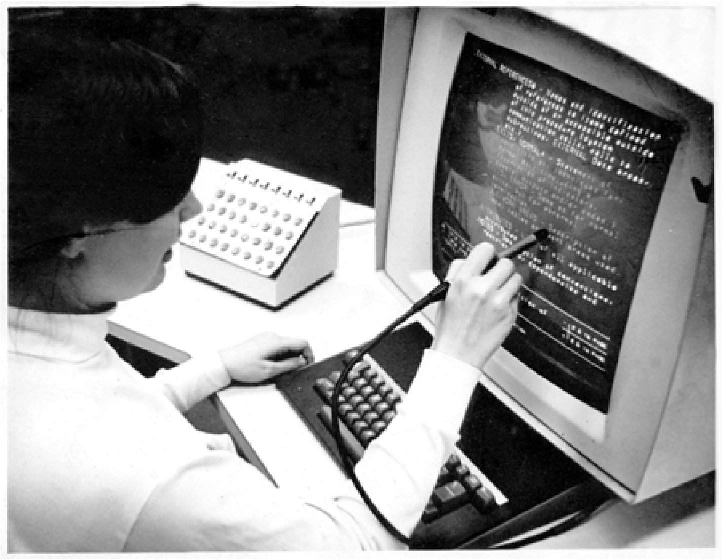

Figure 1. The Hypertext Editing System (HES) console at Brown University. Original photo by Greg Lloyd, 1969.

HES, as a first prototype, naturally had its shortcomings. Part of the goal behind FRESS was to improve on these shortcomings. Firstly, HES was programmed specifically for the IBM /360 and the /2250 display; there was no device-independence. van Dam had seen the benefits of device-independence in Engelbart's demonstration; in NLS "the command line…worked on basically any device. So they had really engineered in good device independence from the beginning" [van Dam 1999]. FRESS was the first hypertext system to run on readily-available commercial hardware and OS: it was built for the IBM VM/370, an off-the-shelf time-sharing system, and could even run on terminals incapable of multi-window views. 22

Secondly, HES wasn't multi-user; it was specifically targeted towards the 2250, a machine for individual use [van Dam 1999]. The benefits of having a multi-user system were also obvious from Engelbart's demonstration of NLS [Ceruzzi 1998]. Primarily, a multi-user system allowed teams of people to work together more easily. In NLS, the tendency towards knowledge-exchange and collaborative work were more than just a system side-effect — they were a goal. van Dam decided to make FRESS multi-user. Although FRESS "didn't have the kinds of chalk-passing protocols that NLS had" [van Dam 1999] — in NLS, for instance, multiple users could work with a shared view of a single document-in-progress — it was designed from the outset to run on a timesharing system and to accommodate multiple displays of different types and capabilities from teletypewriters to glass teletypes (simple CRT screens) to multi-windowed powerful displays with pointing devices. 23

But the most popular new development for novice users in FRESS was not its capacity to accommodate multiple displays and users; it was the "undo" feature. FRESS pioneered undo for both word processing and hypertext, and arguably influenced the future design of word processing systems. Every edit to a file was saved in a shadow version of the data structure, and that allowed for both an "autosave" and an undo. Brown staff and students understood immediately the importance and usefulness of this feature [van Dam 1999]. As van Dam saw it, the ultimate system would have indefinite undo and redo.[9] 24

Importantly, FRESS supported arbitrary-length strings; it had no size limitations. According to van Dam, an important philosophical distinction between NLS and FRESS was that "Doug had these 4,000-character limits on his statements, and that was an anathema to us. It was an anathema to Ted, when he started out, to have limits on anything" [van Dam 1999]. The FRESS team believed that the system should not force you into an unnatural usage pattern, so software paging allowed for essentially unlimited document sizes while maintaining performance. Working in the FRESS environment, the user could not tell the difference between a two-page and a two-hundred page printed file, and could actually specify how all this would be viewed in the first place. FRESS borrowed many of the viewing specification ideas from NLS. 25

At the same time, hyperlinks in these files were addressable down to the character. The granularity was as fine as sand. As van Dam put it in our interview, "I don't want to go to a book, I don't want to go to a chapter, I want to go to the actual quote!" One of the reasons why the web does not have the same functionality as earlier hypertext systems such as FRESS or NLS is that a URL points to a document: unless the author pre-specifies anchors in the target page, there is no finer granularity than that. In NLS, a link took you to a statement. In FRESS it could take you to a character. So in this sense, van Dam points out, "I think we had a 'creamier' hypertext than NLS did," and they certainly had a creamier hypertext than the modern web. 26

Another aspect of FRESS which the web has not implemented is bidirectional linking. HES had unidirectional links, and the FRESS team decided that this needed to be changed. FRESS was the first hypertext system to provide bidirectional linking. It was also groundbreaking in that it provided different options to *visualise* this field of links and references, to see the global context for work: a map. 27

The outline functionality in FRESS was inspired by NLS (or as van Dam puts it, "was a straight rip-off" [van Dam 1999]). For readers who are not familiar with this terminology, an outline processor is a specialised word processing program for handling and displaying outlines. One of its most powerful features is the ability to suppress lower levels of detail and see the global context for work [Conklin 1987, 25]. The user can view just the top *n* levels of the structure —a map whose level of detail is defined by the user. This facilitates rapid movement between sections and makes editing and global navigation easier. But most importantly, this meant that in FRESS 28

> [We] had an ability to see the structure space, a visualisation of all the structure in the text, the outline structure and the cross-referencing structure. You could do structural rearrangements in that structure space in a quick overview mode and you would thereby induce those same edits in the text itself. [van Dam 1988, 891]

29

In sum, unlike in contemporary hypertext systems like the web, FRESS provided a variety of coordinated views and conditional structure and view-specification mechanisms. The user had final control over how and how much of the document was being displayed — unlike with embedded markup like HTML. It also afforded separation of structure from formatting and hypertext semantics — also unlike HTML. And all of these features were designed to be easy to learn for novice users, based as they were on a multi-window function key and lightpen interface in addition to a command-line interface (in contrast to NLS, which had a more complex, and correspondingly more powerful, command language). FRESS actually displayed and handled complex documents better than non-hypertext "word processing" systems of the time. It was so intuitive and efficient that it "was used as a publishing system as well as a collaborative hypertext environment for teaching, research and development" [DeRose & van Dam 1999, 7]. It was also used at Nijmegen University and Philips Eindhoven in the Netherlands [DeRose & van Dam 1999, 19].

30 But this is not to say that FRESS was a run-away success at Brown, or that the project received financial support. In fact, quite the reverse; "computer scientists didn't think it was computer science, and humanists weren't paying for technologies" [van Dam 1999]. The most difficult group to deal with were the staff in charge of computing. In those days, computing time on the university mainframe was allocated via a virtual accounting system. You got n-thousands of dollars in your account (this was called virtual money or "funny money"), and when you ran out, your time was up. van Dam and the FRESS team, of course, were always arguing for more. One year the VP for Finance told the FRESS team that there would be no more money for this FRESS hypertext nonsense, as the computer was for serious physics and engineering work. van Dam had to use blackmail to keep the project alive.

> Under the Brown system, the computer is just as much a public utility as the library, and you can no more cut off people based on their field from the computer than you can cut them off from the library... only by threatening to go public and let the campus know that the engineers and physicists were treating the computer as their private fiefdom [did I] get the money. [van Dam 1999]

Luckily, the FRESS team kept going. Engelbart kept going. Ted Nelson kept writing and publishing and inspiring audiences of young hackers and bright-eyed alums. And slowly, very slowly, the computing community began to realise that text and hyperlinks on a computer might just work.

31 In 1976, the National Endowment for the Humanities supported a FRESS application for teaching English poetry. The FRESS team, and particularly van Dam, had wanted to use the system explicitly for teaching since its inception. In the NEH-sponsored course, students did all of their critical course reading and writing in the online FRESS docuverse. A poetry textbook was created in FRESS, to go along with a large collection of poems by Spenser, Tennyson, Blake and others [DeRose & van Dam 1999, 19]. Students' and teachers' comments were integrated into the web as the course proceeded, prompting further response and debate in the manner of contemporary teaching webs. And, incidentally, the term *web* was actually used in accordance with Vannevar Bush's understanding of the term as a cluster of trails [Bush 1945]. It was the first time hypertext had been used as a central teaching tool in the humanities, and arguably the first online scholarly community.

32 Intermedia, a distributed hypermedia system developed at Brown's Institute for Research in Information and Scholarship (IRIS) from 1985-1990, grew directly out of the work done by Van Dam's group. This was intended to be a networked, multiuser teaching tool.

Hypertext and the Early Internet

33 FRESS occurred on the cusp of the network era — 1969. HES had been neither networked nor multi-user. It was intended to be a personal device, for the use of writers and editors. But NLS and FRESS were pioneering networked multi-user hypertext systems. Although FRESS wasn't ideal in this sense, "You could still work together. It was multi-user" [van Dam 1999]. NLS and FRESS were both shifting across to a different technical phylum — from the personal tool to the public network. From 1968 on, research into hypertext systems focussed not on the creation of personal webs, but on shared and distributed systems. In fact, Doug Engelbart's NLS was one of the first nodes in the fabled ARPANET, which eventually spread its wires around the globe to become the Internet. However, as van Dam observed in an interview with the author,

> I don't think...that any of us truly envisioned that you could have what we have today, because the issues of interoperability just seemed far more insurmountable in those days. We didn't have networks with the kind of bandwidth that you do today. So, I certainly never foresaw the world-wide web. It surprised me. The fact that I can reach out and touch stuff in Ethiopia, as it were, is still a surprise to me. [van Dam 1999]

34 van Dam claims he didn't foresee the web; neither he nor Doug Engelbart could have imagined where their technologies would lead. But it is undeniable HES became the first working example of hypertext on commercial equipment, and in that sense a vision of potentiality for all future hypertext technologies. According to Ted Nelson, HES was actually the precedent to today's web browser [Nelson 1999].

35 So although there was much resistance to Van Dam's original project, it spawned and legitimised a new field of research: hypertext. There is still much work to be done in this arena, work which van Dam continued to do at Brown, and Engelbart at the Bootstrap Institute (now called The Doug Engelbart Institute)— the organisation he founded to continue evolving the relationship between humans and computers. Because "the things we decide now will continue to haunt us for decades. Decades" [van Dam 1999].

Acknowledgments

36 My thanks to Professor Andries van Dam, Dr Douglas Engelbart and Dr Ted Nelson for their time and interviews. Thanks also to Professor van Dam and Rosemary Michelle Simpson for their corrections and comments on this article.

Works Cited

Barnet 2005 Barnet, Belinda. *The Magical Place of Literary Memory: Xanadu*. Screening the Past. http://tlweb.latrobe.edu.au/humanities/screeningthepast/firstrelease/fr_18/BBfr18a.html.

Barnet 2006 Barnet, Belinda. "Engelbart's Theory of Technical Evolution". *Continuum: the Journal of Media and Cultural Studies* 20: 4 (2006), pp. 509-521. http://www.informaworld.com/smpp/content~content=a767889432~db=all~order=page.

Barnet 2008 Barnet, Belinda. "The Technical Evolution of Vannevar Bush's Memex". *Digital Humanities Quarterly* 2: 1 (2008). http://www.digitalhumanities.org/dhq/vol/002/1/000015.html.

Bush 1945 Bush, Vannevar. "As We May Think". *Atlantic Monthly* 176: 1 (1945), pp. 641-649.

Carmody et al. 1967 Carmody, Steven, Walter Gross, Theodor H. Nelson, David Rice and Andries van Dam. “A Hypertext Editing System for the /360”. In M. Faiman and J. Nievergelt, eds., *Pertinent Concepts in Computer Graphics*. Urbana: University of Illinois, 1967. pp. 291-330.

Ceruzzi 1998 Ceruzzi, Paul E. *A History of Modern Computing*. Cambridge, MA: MIT Press, 1998.

Childress 1998 Childress, Vincent. “Book Review: Engineering problem solving for mathematics, science, and technology education”. *Journal of Technology Education* 10: 1 (1998). http://scholar.lib.vt.edu/ejournals/JTE/v10n1/childress.html.

Conklin 1987 Conklin, Jeff. “Hypertext: An Introduction and Survey”. *IEEE Computer* September (1987), pp. 17-41.

DeRose & van Dam 1999 DeRose, Steven, and Andries van Dam. “Document Structure and Markup in the FRESS Hypertext System”. *Markup Languages: Theory and Practice* 1: 1 (1999), pp. 7-32.

Engelbart 1999 Engelbart, Douglas 1999, Interview with the author.

Nelson 1999 Nelson, Ted. Interview with the author.

Nyce & Kahn 1991 Nyce, James, and Paul Kahn, eds. *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. London: Academic Press, 1991.

Owens 1991 Owens, Larry. “Vannevar Bush and the Differential Analyzer: The Text and Context of an Early Computer”. In James Nyce and Paul Kahn, eds., *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. London: Academic Press, 1991. pp. 3-38.

van Dam 1988 van Dam, Andries. “Hypertext ’87 Keynote Address”. *Communications of the ACM* 31: 7 (July 1988), pp. 887-895.

van Dam 1999 van Dam, Andries. Interview with the author.

Wardrip-Fruin & Montfort 2003 Wardrip-Fruin, Noah, and Nick Montfort, eds. *The New Media Reader*. Cambridge: MIT Press, 2003.

Wolf 1995 Wolf, Gary. “The Curse of Xanadu”. *Wired* 3: 6 (June 1995). http://www.wired.com/wired/archive/3.06/xanadu.html.