

Lecture for Week 1

Read this lecture before attempting the assigned reading. Then refer back to it again as you delve into them. If you have never been exposed to Marshall McLuhan’s writings before, you will certainly find him to be the more difficult of the two authors, but soldier through. The seminal essay “The Medium is the Message” was in many ways the intellectual backbone for much radical thinking about technology and media of all kinds in the second half of the twentieth century.

The additional readings referred to and linked to throughout this lecture are not required—feel free to read them or browse through them if they pique your interest. They are just the sorts of things I might like to see mentioned in future threaded discussions. So don’t be afraid to click on the links—that’s what the lecture’s all about, after all....

Page numbers refer to the editions of the books on the syllabus.

Software documentation is a discipline at the intersection of high technology, workflow management, and creative writing. To navigate that intersection successfully, it is helpful to develop the ability to look at software from a perspective different from the typical end user’s, in a way that “make[s] us think twice about experiences that are second nature to us,” to use Steven Johnson’s words. *Interface Culture: How New Technology Transforms the Way We Create & Communicate* has been assigned as an attempt to get you to begin thinking about software differently from the way computer scientists usually do.



Fig. 1: [Steven Johnson](#)

“Your job as a writer is to work with [a] program interface and your knowledge of the user to produce descriptions, procedures, overviews, explanations, and all the content of [a] document. That content shares your wisdom and experience with the program in such a way that your users can follow easily and productively” (Barker 199).

Johnson’s book examines various aspects of software interfaces in counterintuitive, curious, and imaginative ways. It also provides some useful history about the development of the personal computer. As you read it, think about how each topic might relate to issues of software documentation, but don’t push the text farther than it will go. Ultimately, it is a book about interfaces themselves. Enjoy it as an example of good

writing about software; think of it as a way to get you thinking. Don't be afraid to stop and look up references to any unfamiliar authors, architects, composers, or artists. And take note of the many successful descriptions of software Johnson creates: he is a fine writer.

Here are four ideas to think about as you read the book:

1. Technology and culture, (or engineering and art), are not dichotomous. Software documentation is but one example of this.

“...[T]hink about the object world of technology as though it belonged to the world of culture, or as though those two worlds were united. For the truth is, they have been united all along.... But we have a habit—long cultivated—of imagining that as separate.... The opposition colors much of modern thought” (Johnson 1).

“We have reached a point where the various media evolve so rapidly that the inventors and the practitioners have blurred into one holistic unit, like a science lab hosting a creative-writing seminar. There are no artists working in the interface medium who are not, in one way or another, engineers as well. This has always been the case with culture and technology, of course; it's just that we used to pretend it was otherwise, by dutifully keeping the painters and the mechanics separate, on the college campuses, in the museum halls, on the bookshelves...” (Johnson 7).

(Johnson's idea, is, by the way, in some ways in direct opposition to the one put forth in the famous Cold War–influenced 1959 essay *Two Cultures* by the British novelist and physicist [C. P. Snow](#), who argued that artists and scientists were basically illiterates when it came to knowledge about the others' disciplines—and talked about the inherent dangers for society in an educational system that permitted such an outcome.)

This semester, on this virtual college campus, at least, we are going to try to reunite the two disciplines, as Johnson suggests, in the artist/engineer persona of software documentation writers: each of you.

2. Today's technological advances are not necessarily more radical than those of the past. They are just happening in more rapid succession. Because of that, their influences on the way we live, work, and even think, and on each other, have become easier to detect.

“Today it is the instant speed of electric information that, for the first time, permits easy recognition of the patterns and the formal contours of change and development” (McLuhan 353).



Fig. 2: [Marshall McLuhan](#)

Marshall McLuhan, a Canadian professor of English literature, was one of the first academics to teach about popular culture. In 1965, he became one of the world's first academic celebrities. It is difficult to comprehend today how famous this man's ideas made him, especially given that he was a rumpled, middle-aged academic not unlike the chairman of our own department. He coined the term "mass media" as it is now used (as well as "information age" and "global village"), and his brilliance and influence were, in his own lifetime, compared to those of Newton, Darwin, Freud, Einstein, and Pavlov. Although he died in 1980, before the heydays of desktop computing or the advent of the Internet, he managed, in his writings, to anticipate the invention of TV shopping networks, e-mail, and the Internet itself. He is very much the intellectual father of philosophers like Steven Johnson.

He saw electronic media as nonlinear, repetitive, discontinuous, intuitive, and proceeding by analogy rather than by sequential argument—that is, in direct contrast to the books and newspapers that had preceded it. His most famous dictums include, "*The medium is the message,*" "*We become what we behold,*" and "*We shape our tools, and thereafter our tools shape us.*"

"Beginning from the premise that "we become what we behold," that "we shape our tools, and thereafter our tools shape us," McLuhan examines the dictats of two technological revolutions that overthrew a settled political and aesthetic order: first, in the mid-fifteenth century, the invention of printing with movable type, which encouraged people to think in straight lines and to arrange their perceptions of the world in forms convenient to the visual order of the printed page; and second, since the late nineteenth century, the new applications of electricity (telegraph, telephone, television, computers, etc.), which taught people to rearrange their perceptions of the world in ways convenient to the protocols of cyberspace. Content follows form, and the insurgent technologies give rise to new structures of feeling and thought" (McLuhan xii).

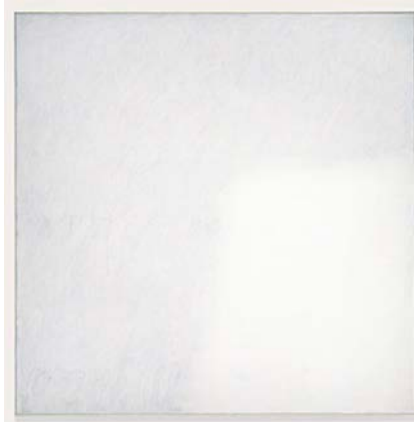
For example, the invention of movable type and the printed page voided a previous oral tradition and encouraged people to think in straight lines, to arrange their thinking and their perception of the world in forms convenient to the visual order and physical constraints of the printed page. Similarly, the invention of electricity and its applications to information (the telegraph, the telephone, television) taught people to rearrange their perceptions of the world in ways convenient to those protocols. In other words, *content*

always follows form, and new technology generates new feelings, thoughts and ideas. (How similar this is to the subtitle of Johnson's own book).



Figs. 3 & 4: Thomas Edison and Dr. Dre. Music had always existed, but being able to record it changed the way people thought about it and related to it.

These were pretty radical ideas in the 1960s, but in fact described something that had always been true. Before Thomas Edison invented a way to record sound (and figured out that it was good for something besides recording phone conversations), it was unusual for even the most accomplished musician to hear all of Beethoven's symphonies in his or her lifetime, because it involved being in proximity to a concert hall when one might be performed, and being able to afford to attend the concert. Today, it is commonplace for serious students of classical music to own several different recordings of each; to be able to call forth several more on demand via various Internet or cable TV channels; to fill in the gaps of their collections by the next day via outlets like Amazon.com; and even to have the scores for all nine works delivered at the same time. The instant access to this music has changed the nature of how musicians understand it, relate to it (it's no longer such a rare commodity), study it (you can listen to it over and over again if you choose—the same four bars if you're enamored of them), and, ultimately, how it influences the kind of music other composers write. A clearer but analogous example can be seen in the sampling done by rap musicians, who actually use recordings of other artists in their own compositions—a technique impossible prior to the invention of recording devices.



Figs. 5, 6 & 7: Paris Hilton (left): famous for being famous. Robert Ryman's *Surface Veil I*, 1970 (middle); oil and blue chalk on linen, 143 15/16 x 144 inches; Solomon R. Guggenheim Museum, New York: famous for being blank (notice any resemblance?). Jan Vermeer's *Girl with Pearl Earring*, c. 1665–1666 (right); oil on canvas, 44.5 x 39 cm; Royal Cabinet of Paintings Mauritshuis, The Hague: famous.

Similarly, McLuhan was the first to understand that media *about* traveling changed the entire *experience* of traveling, and that ultimately life would come to imitate the media itself. He said, “The postmodern imagination is a product of the mass media,” predicting phenomena like Paris Hilton, people who are famous merely because they are famous. In other words, at some point, the medium stops carrying the message, and simply *becomes* the message itself, whether that “message” is the use of recording devices to make “music,” the use of television to manufacture “fame,” or the reduction of painting to bare white canvases: an attempt by the artist to draw attention to the fact that a painting is merely painting, the exact opposite of what portrait painters during the Renaissance (like Vermeer) were striving for.

But Johnson wisely notes, “you can’t measure a medium’s influence without something to compare it with” (Johnson 4). “The automobile changed the way our cities developed.... The psychological introspection of the late-nineteenth-century novel paved the way for psychoanalysis.... The interface has already changed the way we use our computers, and it will continue to do so in the years to come. But it is also bound to change other realms of modern experience in more unlikely, unpredictable ways” (Johnson 25).

“McLuhan’s claim [was] that the printing press was the dominant force that transformed the hermeneutic Middle Ages into our scientific society.... The press didn’t do it just by making books more available, it did it by changing the thought patterns of those who learned to read” (Johnson 25). In other words, “the medium is the message,” McLuhan’s famous quote, is as true of computers and the Internet, which he did not live to see, as it was of the inventions like radio and television, about which he wrote about in detail.

“We’re used to communicating with our friends and family by sending them snapshots or sketches or tape mixes, but in the future we will reach out to those around us by sharing virtual environments” (Johnson 74). Environments, I might add, that will require documentation on how to use them.

“[U]sing a word processor changes how we write—not just because we’re relying on new tools to get the job done, but also because the computer fundamentally transforms the way we conjure up our sentences, the thought process that runs alongside the writing process” (Johnson 142). It affects the “sheer volume” of what we can generate, the “perishability” of what we create, and changes the relationship between a sentence “in its conceptual form and its physical translation onto the page or the screen” (Johnson 143).

We might even choose not to use certain words if Microsoft Word marks them with a red squiggle and we’re too lazy to look them up in the dictionary. In the end, Johnson writes, “[t]he computer...has changed the very substance of what I was writing, and in that sense, I suspect, it had an enormous effect on my thinking as well” (Johnson 145). This is precisely what Marshall McLuhan meant when he said that the medium is the message: that content follows form; that we may invent our tools, and afterwards, they, to a large extent, proscribe how we function. This is as true for technical writers as for anyone else.

“[N]ew technological forms—the television, the radio, the book—transform not only the balance of power between our senses but also our experience of other media. “Radio changed the form of the news story as much as it altered the film image in the talkies,” McLuhan argued. “TV causes drastic changes in radio programming, and in the form of the film or documentary novel” (Johnson 178). He would agree that computers have radically altered our experience of all forms of media that came before them, just as the Web altered our experience of computers.

3. Every act of writing and editing is a political act, and carries with it a value judgment.

“To imagine a language is to imagine a way of life.”—*Ludwig Wittgenstein*

“In our time, political speech and writing are largely the defence of the indefensible.”—*George Orwell*

“Different arrangements of words make different meanings, and different arrangements of meanings produce different effects.”—*Blaise Pascal*

“As a poet and writer, I deeply love and I deeply hate words. I love the infinite evidence and change and requirements and possibilities of language; every human use of words that is joyful, or honest or new, because experience is new.... But as a Black poet and writer, I hate words that cancel my name and my history and the freedom of my future: I hate the words that condemn and refuse the language of my people in America.”
—June Jordan

“Every word she writes is a lie, even ‘and’ and ‘the.’”
—Mary McCarthy about Lillian Hellman



Fig. 8: The sculpture depicting the crucifixion on the passion façade of La Sagrada Família, the cathedral designed by the great Catalán architect Antonin Gaudí in Barcelona. Although still under construction today, it relies on the traditional techniques of telling Biblical stories through sculpture that was used throughout the Middle Ages, when many people were illiterate, and such stories were learned orally and reinforced visually by imposing structures like this one.

“In the days before Gutenberg, cathedrals were the great signifying machines of public life.... You could read the story of Christ in the impossibly detailed stone carvings.... All works of architecture imply a worldview, which means that all architecture is in some deeper sense political” (Johnson 44).

“Because the computer was by definition so malleable, capable of shape-shifting from one visual metaphor to another, it was theoretically possible for the interface to look like practically anything... (Johnson 45).

There are practical reasons why the desktop metaphor caught on, but political ones as well, influenced as much by capitalism as anything else (why *wouldn't* a corporation create a work environment that encourages computer users to use metaphors that relate to an office environment?) Similarly, the way you write, the language you choose to use,

and not to use, communicates something about you, your values, and your opinions about the software about which you are writing, as well as the task of software documentation itself. No piece of writing is value-free.

What has the way I have chosen to present this lesson to you told you about me so far? You can probably make some pretty good guesses about my politics, my educational background, my pedagogical philosophy, and my opinions about technology just from this one lesson, the language and images I have used, even from my grammar and punctuation.

4. The way things currently work is not necessary the best way.

If you took a class in biology in high school (and not since then) you probably have a fuzzy recollection of how scientists classify animals—vertebrates and invertebrates, fish amphibians, reptiles, birds, and mammals; insects, arachnids, crustacea, protozoa, and the like. Apparently, the Chinese, at a certain point in their history, had quite a different idea.



Fig. 9: “Those that are trained”

“These ambiguities, redundancies, and deficiencies recall those attributed by Dr. Franz Kuhn to a certain Chinese encyclopedia entitled *Celestial Emporium of Benevolent Knowledge*. On those remote pages it is written that animals are divided into (a) those that belong to the Emperor, (b) embalmed ones, (c) those that are trained, (d) suckling pigs, (e) mermaids, (f) fabulous ones, (g) stray dogs, (h) those that are included in this classification, (i) those that tremble as if they were mad, (j) innumerable ones, (k) those drawn with a very fine camel’s hair brush, (l) others, (m) those that have just broken a flower vase, (n) those that resemble flies from a distance.”—Jorge Luis Borges, *The Analytical Language of John Wilkins*

Who’s to say which system is more valid? More comprehensive? More “scientific”? In the end, it comes down to how you look at the world. Steven Johnson makes a similar point. “Since the desktop is organized around an illusion (space), why not organize it around another illusion: meaning? Perhaps the whole idea of one document existing in a single physical location... is just a hollow legacy carried over needlessly from the limitations of the real world” (Johnson 170–1).

Similarly, the way people work with software, and the way the software itself has been designed, is not always ideal. As a software documenter, it is your job to look at things from all angles, and imagine new ways things might be done. That is one of the reasons

I've had you read Steven Johnson and Marshall McLuhan—to get you to question the most basic ideas you've had up until now about interfaces, software, and workflow.

So, what *is* an interface, anyway?

According to Robert J. Johnson in his essay “When All Else Fails, Use the Instructions,” which you'll read later this semester, “The *user tasks* are the actions of use as perceived by the user. The *system actions*, on the other hand, are the actions that the computer system was programmed to accomplish by a system developer. The point at which these two aspects of technology meet is commonly called the *interface*” (Peeples 306).

Put another way, an interface is “software that shapes the interaction between user and computer. The interface serves as a kind of translator, mediating between the two parties, making one sensible to the other. In other words, the relationship governed by the interface is a semantic one, characterized by meaning and expressing rather than physical force” (Johnson 14). Extending that definition, a sign-language interpreter might also be seen as an interface of sorts: a semantic translator mediating between two parties who speak different languages—English, and American Sign Language. In this case, the interface, or translator, must speak both. A computer interface “speaks” both machine language and a graphical language accessible to end users.

This may seem fairly obvious, but the point (and the distinction) is important for technical writers. Other kinds of machines, such as cameras or printing presses, deal with representations as end products, but computers are completely symbolic systems.

“Those pulses of electricity are symbols that stand in for zeros and ones, which in turn represent simple mathematical instruction sets, which in turn represent words or images, spreadsheets or e-mail messages. The enormous power of the modern digital computer depends on this capacity for self-representation” (Johnson 15).

“[T]he modern graphic interface is defined by “direct manipulation.” The user makes things happen in an immediate, almost tactile way: instead of telling the computer to delete a file [the way you would in DOS], you drag it into the trash can. The underlying event is the same (the CPU follows instructions to erase a few sectors of the hard drive), but the *illusion* of the graphic interface is that you seem to be doing the work yourself” (Johnson 179). In other words, an interface can also be thought of as a set of illusions.

```
C:\>win

You are already running Windows.

| Press ALT+ENTER to switch this MS-DOS prompt between
  windowed and full-screen display.
| Type Exit and press Enter to quit this MS-DOS prompt and
  return to Windows.
| Press ALT+TAB to switch to Windows or another application.
```

Fig. 10: The decidedly non-illusory world of DOS: “...all those inscrutable commands and instruction sets emblazoned on green-phosphor monitors and etched into punch cards” (Johnson 150).

And what is a metaphor?

We all learned the dictionary definition of a metaphor back in grade school: a word or phrase that usually refers to one thing used to designate another, thereby making some sort of comparison. The famous example is from Shakespeare’s *As You Like It*: “All the world’s a stage,/and all the men and women merely players.” I prefer a slightly more modern (not to mention homegrown) example:

You’re the top
You’re the Coliseum,
You’re the top!
You’re the Louvre Museum,
You’re a melody from a symphony by Strauss,
You’re a Bendel bonnet,
A Shakespeare sonnet,
You’re Mickey Mouse.

You’re the Nile,
You’re the Tower of Pisa,
You’re the smile
On the Mona Lisa.

I’m a worthless check, a total wreck, a flop,
But if baby, I’m the bottom, you’re the top. —Cole Porter, lyricist and composer



Fig. 11: Cole Porter, an American master of metaphor.

Metaphors—tropes in which a word or phrase that usually refers to one specific thing is used to designate something else that bears some sort of relationship to it—are a very basic form of symbolism that form the cornerstone of poetic language (and have crept into everyday language as well: think of expressions like “a sea of troubles,” or even “it’s always darkest before the dawn.” Metaphors also form the “core idiom of the contemporary graphic interface” (Johnson 15).

“The widespread adoption of the GUI [graphical user interface] has dramatically changed the way in which humans and computers interact, and has greatly expanded computer literacy among people once alienated by the arcane syntax of the older “command line” interfaces.” (Johnson, 16) (Check out what command-line interfaces [looked like](#), and the nonsensical [alphabet soup](#) they forced their users to speak.)

Unfortunately, GUI elements can also form the core of what confuses readers about the documentation they are reading. Just as interface metaphors are used consistently (folders always look the same, and always behave the in the same way), they must always be described consistently and precisely in any piece of software documentation. In fact, you will be hearing a lot about the cardinal rules of **consistency** and **precision**—something interface designers and software documentation writers have in common—throughout the semester.

“What makes a metaphor powerful is the gap between the two poles of the equation. Metaphors create relationships between things that are not directly equivalent. Metaphors based on complete identity are not metaphors at all. In traditional interface design, a computer “window” bears a kind of superficial resemblance to a real-world window, but it’s the differences between the two that make the metaphor a successful one. We obviously can’t layer our kitchen windows atop one another, nor can we scroll through the vista they provide. There’s a necessary distance between the real and virtual window that makes the analogy useful to us” (Johnson 59).

Interestingly, “without exception, the leading examples of digital sociability didn’t require a spatial metaphor to make their communities happen. For the most part, the social fabric of cyberspace is still stitched together by the gossamer thread of text” (Johnson 71).

“As it turns out, the one domain that has successfully extended the original desktop metaphor into three dimensions is that of video games...” (Johnson 72).



Fig. 12: Keeping metaphor alive: A scene from Grand Theft Auto, a popular video game. ([Play](#) a video game over the Internet in real time.)

“The first time you encounter a scrolling window, there’s a sense of depth to it. The window appears to look out onto a dataspace that continues beyond the borders of the window itself.... But the illusion quickly wears off. The window starts to feel more two-dimensional, more like a piece of paper than a portal.... [T]he original metaphor has worn off, become literalized. This process is as old as language itself. The world of words is littered with these dead metaphors, poetry that has long since ossified into prose. We no longer think of our virtual windows as analogs of the real-world version. They’re a species unto themselves” (Johnson 87). This is one reason why software documentation is required to explain even the most basic elements of a user interface, and to use terminology consistently.

Where did the language of computing come from?

Well, we’ve borrowed from analogous forms that have come before it, of course. This is quite common. There is a fair amount of synesthesia in the language of the arts. We borrow words from painting to describe music, for example, and describe orchestral “colors,” as “dark” and “light,” for example. But as Johnson points out, computer metaphors went in a very specific direction.

“For the first time a machine [the computer] was imagined not as an attachment to our bodies, but as an environment, a space to be explored.... Most of today’s high-tech vocabulary derives from this initial breakthrough: cyberspace, surfing, navigating, webs, desktops, windows, dragging, dropping, point-and-clicking” (Johnson 24). These are all metaphors for navigating space.

A windows conundrum

Johnson asks why a windows-driven interface should be easier to use than a text-driven one.

“Spatial memory works only if the objects you’re trying to keep track of remain anchored in one place.... [But w]indows are more fluid, more portable.... [In fact,] windows rarely work in the service of spatial memory.... Even the most devoted windows advocate still thinks of his or her files in textual terms better suited to command-line interfaces such as DOS and Unix. To understand this, you need only pay attention to your thought processes when you’re fumbling around with your file-management software, looking for a stray document.... We pretend...that we’re remembering “where” we put the file, but what we’re really remembering is the name of the folder that contains it” (Johnson 77–8). For this reason, good software documentation shies away from discussions of physical locations of icons, especially in relation to one another: their locations are completely mutable. The locations of menus and other fixed interface items can be located with more assurance (*the File menu, located at the top left of the Microsoft Word menu bar, for example*).

Thinking about links

We have become so used to clicking on these handy devices that we rarely stop to think about how they function, or how they have changed the way we work and interact with our computers.

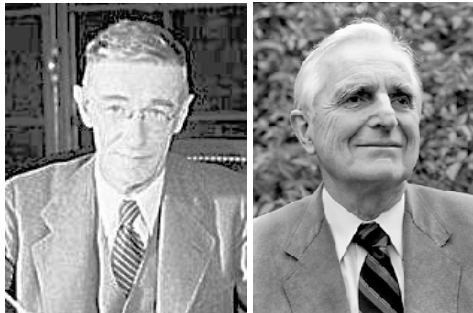
“[L]inks that join...various destinations are links of association, not randomness. A channel surfer hops back and forth between different channels because she’s bored. A Web surfer clicks on a link because she’s interested” (Johnson 109). (Notice how the word *surfing* has actually come to mean its opposite, in a way.)

“The imaginative crisis that faces us today...comes from having too much information at our fingertips, the near-impossible task of contemplating a colossal web of interconnected computers.... The modern interface is a kind of corrective to this multiplying energy, an attempt to subdue all that teeming complexity.... And on the World Wide Web, where this imaginative crisis is most sorely felt, it is the link that finally supplies that sense of coherence (Johnson 116). So links, like software documentation itself, are a way to make sense of the blizzard of information faced by any computer user.

“Our traditional ways of organizing things ...are built around fixed stable identities: each document belongs to a specific category....” (Johnson 118). Think about the way we catalog library books, or classify insects, or elements in the periodic table, or even the way an encyclopedia is structured. Inspired in part by the writings of Vannevar Bush and his theoretical Memex machine, people began to think about digital information on the Web differently. “What made a nugget of information valuable...was not the overarching class or species that it belonged to, but rather the connections it had to other data.” (Johnson 118–9). This concept certainly has implications for the way online documentation, as well as the way documentation about websites, is structured.

You might have noticed a certain disdain for software documentation on the part of Mr. Johnson: “Users will learn over time to inhabit each new space as though they were developing sea legs. After a few acclimations, the initial sense of disorientation will seem less intimidating.... You can see this aptitude already in the generation of kids raised on video games. There’s a certain fearlessness they exhibit upon entering into a new information-space. Instead of reading the manual, they’ll learn the parameters in a more improvisational, hands-on fashion” (Johnson 228).

The behavior Johnson observes is true enough, and it’s been a long time since the first thing I did after loading a piece of software was look at the instructions. But I disagree with him on two levels. The majority of us are *not* the generation of kids who were raised on video games, and it will be a long time before that is true. And even *that* generation dusts off the manual when things in their brand new information-space get all jammed up, or they can’t figure out how to make the newest laser weapon they spent all that extra money for do what they want it to do. As intuitive as user interfaces are becoming, good documentation will never be irrelevant, although it must continue to adjust to the changing needs of its users.



Figs. 13 & 14: Vannevar Bush (left) and Douglas Engelbart

As you will read in Johnson’s chapter on links, the ideas of Vannevar Bush and Douglas Engelbart were the groundwork upon which the actual technology that become desktop computers was finally built. “As We May Think” is the [article](#) by Bush, originally published in the *Atlantic* in July 1945, which started it all. Douglas Engelbart’s [computer demonstration](#) at Stanford in 1968 started the revolution that would become desktop computing, debuting what would eventually become the mouse.

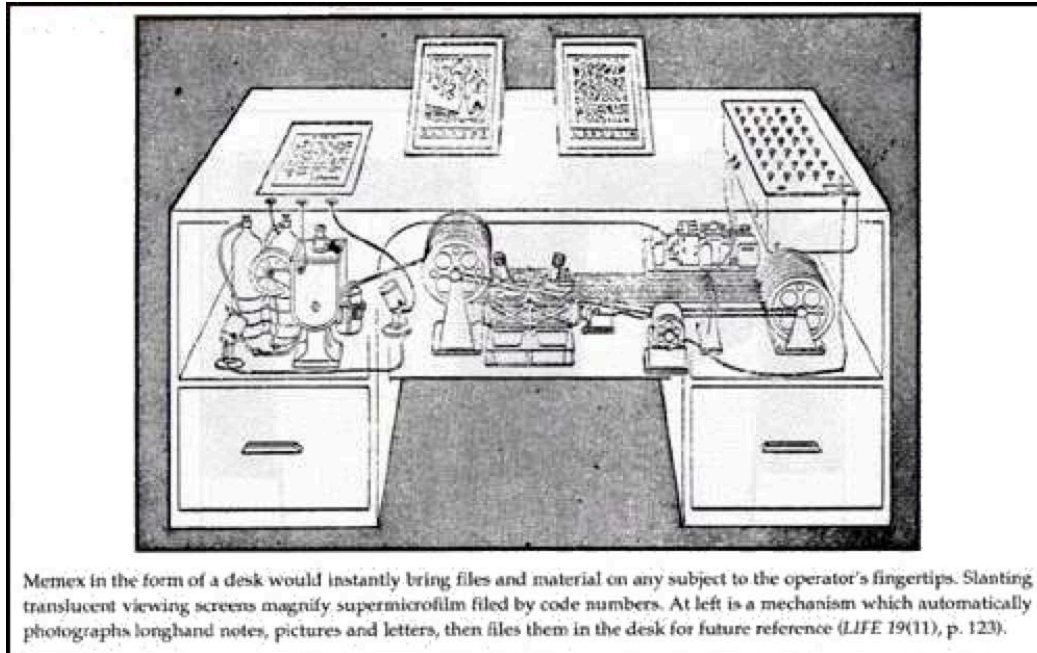


Fig. 15: A drawing of Vannevar Bush's Memex machine. "Memex wouldn't see the world as a librarian does, as an endless series of items to be filed away on the proper shelf. It would see the world the way a *poet* does: a world teeming with associations minglings, continuities" (Johnson 119). While I think Johnson gives librarians a bad rap here (they did, after all, invent the idea of subject catalog and cross references), Bush's idea of having a digital medium built around semantic links rather than physical locations has never been fully realized.

In conclusion

If you are taking this course, chances are you are no Luddite. You already have a healthy affinity for technology, and perhaps a knack for writing as well. Combining the two can be surprisingly challenging.

"Every age comes to terms with the latest technology by drawing upon imagery of older and more familiar things" (Johnson 16). "In today's society, the task of translation has migrated to the technicians" (Johnson 17). Among those technicians are technical writers like you. Whether or not you believe the medium of software interfaces is genuinely headed toward the "breadth and complexity of genuine art" (Johnson 18) as Johnson suggests, you will need to develop the ability to formulate a perspective on it, and draw upon a language through which you can then describe it consistently and accurately. "Some of this language will rise up *sui generis* out of the new technologies, but most of it will borrow extensively from preexisting traditions.... We need a new language to describe the new medium of the interface, but that doesn't mean we can't borrow some of our terminology from the forms that have come before it" (Johnson 18-9). And of course, you will have to learn to use that language grammatically.



Figs. 16 & 17: The machine that changed it all (left), and the man who invented it. A replica of Gutenberg's original printing press, which made the mass production of books possible, and moved society from an oral to a literary way of thinking.

A computer, Johnson notes, is a “symbolic system from the ground up,” “a machine that traffics in representations or signs rather than in the mechanical cause-and-effect of the cotton gin or the automobile” (Johnson 15). It has this in common with language, which is also a bottom-to-top symbolic system—words represent things and ideas, they aren't those things and ideas themselves.

But just because the GUI has made computers much more intuitive and user-friendly than they were in the days of flashing green C-prompts and hard-to-remember DOS commands does not mean that the visual metaphors in use today make computer documentation unnecessary. Quite the contrary.

Johnson points out that today's “parasite forms” of media “thrive in situations where the available information greatly exceeds our capacity to process it.... In these climates, all manner of metaforms appear: condensers, satirists, interpreters, samplers, translators.” (Johnson, 32) Software documentation is one such metaform that has always been around to help process information greatly exceeding people's capacity to understand it. It can function as a condenser, an interpreter, a sampler, even a translator or a filter, deciding for readers what is and is not important for them to concern themselves with at that moment (although probably never as a satirist, like the three examples below). Online software documentation has something else in common with these parasite forms Johnson discusses—its *flexibility*.



Figs. 18–20: The cast of *Reno 911* (top), *The Daily Show*'s Jon Stewart (left), and *South Park*'s the angst-ridden quartet fourth graders (all on Comedy Central): some of my favorite media parasites.

“In the early sixties McLuhan famously remarked that living with electric and mechanical technologies at the same time was “the peculiar drama of the twentieth century,” and Johnson noted that it is the interface that mediates this “strange new zone” created by the co-existence of the two, the place between McLuhan’s “medium” and “message.” It is the task of the writer of the software documentation, then, to describe that zone as cogently, accurately, precisely, and consistently as possible, while always being mindful of the needs of his or her reader.



Fig. 21: Sven Birkerts, a literary critic and writing teacher who discusses how the transition from print media to electronic media is “reweaving the entire social and cultural web”—and not necessarily for the better. Styled after Virginia Wolf’s *A Room of One’s Own*, Birkerts’ *Gutenberg Elegies* is a self-described “inquiry into the place of reading and sensibility in what is becoming an electronic culture.” Read an excerpt entitled [“Hypertext: Of Mouse and Man”](#), or more about [Birkerts and his books](#).