Where's My Memex?:

Using Google and Web 2.0-tech to Bring Us Closer to a Truly Associative & Annotatable Internet

"All our steps in creating or absorbing material of the record proceed through one of the senses—the tactile when we touch keys, the oral when we speak or listen, the visual when we read. Is it not possible that some day the path may be established more directly?" - 'As We May Think,' Vannevar Bush, 1945

John Mahlman IV

DM7043 : Media Studies 2

May 15 : Spring 2008

Dr. Chris Leslie

1. Introduction

How can a user store something from the Internet that is needed at a later time? What are the primary methods? There have bookmarks, printing, and saving for offline use; these only save a *copy* of one state. One cannot tell if the site has been changed or how it has changed over time with these methods. Linking these saved pages is also impossible to do with current methods.

Websites are not static objects; they change with time, and sometimes completely move to other places. What happens when a page which was bookmarked three months ago moves to a different place on a server? The bookmark becomes useless, and now the user has to either find that information again, or lose it forever. This problem can be attributed to the hierarchical filing system (HFS), the primary filing system for computer systems and the Internet.

Despite recent and innovative technologies, the world is still using the HFS as it's primary means of data storage online. What this does is hinder our ability to quickly find, link, and annotate information for future reference. The human mind does not use a HFS to store its information; it uses associative links to store and recall information almost instantly. What is needed is a filing system that is more natural to the human mind and gives more freedom to the user; an associative filing system (AFS).

Internet technology has advanced so much that it is now possible to use it to create a truly annotatable-AFS that can replace the aging HFS. This paper discusses the problems with the current HFS and give examples that date as far back as the 1940s to show that there is and has been a desire to change from the current HFS to a more natural AFS for the Internet. It describes projects and tools that already allow simple AFS capabilities and allow the user to freely anno-

tate information found online. Finally, it gives present examples that prove that the technology is available to crate a working annotatable-AFS for the Internet.

2. Background

To understand why a change is needed with the HFS we need to look at what the real problem with the current system is--the "HFS problem." Once the HFS problem is made clear, one of the earliest written examples of an annotatable-AFS, Vannevar Bush's *Memex*, will be used to describe how an annotatable-AFS should work.

2.1 The "HFS Problem"

The HFS's that are used today all follow the same principle: files and folders are stored in folders that contain other files and folders. To retrieve the information stored in these folders, a user must work through each folder to find one file. Internet-wise, this problem becomes even more apparent if a page which has been bookmarked by someone happens to move. The link to that page is now obsolete and the user must find the new link to that page, a different link.

With a typical HFS, the user has the ability to store whatever files in whatever folders they please; however, if they want to store the same file in another folder (because it is relevant in some way) they have to *copy* the file to that folder. On the Internet, a user can create a hyper-link to another file. This link does not copy it references a file and links to it, solving the issue of multiple copies, but it does not address the issue of moving a file. Hyperlinks and aliases are all one-way links. As Theodor Holm Nelson, whose *Project Xanadu* we will visit later, writes:

> The Web...vastly but incorrectly [simplifies] these problems to a world of fragile ever-breaking one-way links, with no recognition of change or copyright, and no support for multiple versions or principled re-use. Fonts and glitz, rather than content connective structure, prevail (Nelson, 2000).

These one-way links are a plague on the Internet and can only be cured with a newer filing system which supports two-way linking of information. What these two-way links would do is prevent links to information from being lost if that information happens to change its location, whether it's by one directory or to an entirely new system. Two-way links allow a type of *communication* between files and folders; links could modify themselves to adapt to the changes accordingly. HFS does not support this.

Another problem with the HFS is that information is stored in repetitive, cumbersome trails. These trails can be very long and easily forgotten with deeply nested files and folders. These long links cause frustration to the user as well as waste time. These links are not stored in the system; therefore, it is impossible for a user to recall how the information was obtained.

Finally, typical HFS's do not give the user the ability to annotate information effectively. Annotation is typically done inside specific files this means in order to create an annotation about a specific file, that file must be edited by the user. Making these small changes in documents can cause problems for the user, but may also create unnecessary copies of the files. These copies add more confusion to the user and use valuable storage space on the system.

Creating a system that remedies each of these issues is not an easy task. The next section will describe the most famous, and possibly the ideal, example of a true annotatable-AFS "computer." Although this "computer" is not digital, the principles of its design solve every issue stated above; two-way linking, hierarchical trails, and duplicated information.

2.2 Vannevar Bush's Memex

In 1945, Vannevar Bush published "As We May Think" in *Atlantic Monthly*[1]. Bush's primary reason for writing the essay was the growing amount of scientific information being produced as well as the inaccessibility of the information for research.

> There is a growing mountain of research. But there is increased evidence that we are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers—conclusions which he cannot find time to grasp, much less to remember, as they appear. Yet specialization becomes increasingly necessary for progress, and the effort to bridge between disciplines is correspondingly superficial.
> Professionally our methods of transmitting and reviewing the results of research are generations old and by now are totally inadequate for their purpose. (Bush, p. 37)

But, Bush describes that the real problem with our gathering of information is not a lack of devices for this specific use but the primary method in which we store our data:

> Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge from the system and re-enter on a new path (Bush, p. 43).

Because of these issues, Bush introduces an analog computer system which allows users to link information much like the human-mind (by way of association): the Memex[2].

The Memex is described as "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory" (Bush, p. 43). This Memex would essentially allow the user to store their entire life inside of "the piece of furniture at which he works" (Bush, p. 43). The Memex stored its information using microfilm with codes that are used to recall information as well as link the information one another. These links were done

---

[1] For the purpose of this paper I will use the 1996 reprint from *Interactions*.

[2] A portmanteau of memory and extender or indexer

using personalized association, which is the same way the human brain links information. These links are permanent until the user decides to remove them; unlike the human mind, these links will not fade with time.

The Memex was not only a tool to save space, but a tool to save incredible amounts of time. The Memex allows users to create links between information, thus much less time is spent drawing these links from memory, and much more time is spent discovering new information.

> ...at any time, when one of these items is in view, the other can be instantly recalled merely by tapping a button below the corresponding code space. Moreover, when numerous items have been thus joined together to form a trail, they can be reviewed in turn, rapidly or slowly, by deflecting a lever like that used for turning the pages of a book (Bush, p. 45).

This expedited recall of information is still sought after today.

3. Attempts at Annotation and AFS

After Bush's essay was published, people began to see the problem with the HFS used in computing. As early as the 1960s there were projects, papers, and working tools to help create a more humanistic filing system for information, even before the Internet. The following examples will attempt to show how different technologies have been used throughout the years to create a more annotatable-AFS and that even before Web 2.0 the technology was available for some sort of annotatable-AFS with computing.

3.1 Project Xanadu

In the 1960s and 1970s, American philosopher and sociologist Ted Nelson wrote of a software framework which became Project Xanadu. Project Xanadu was a new writing system which allowed multiple versions of a document to be stored, in what Nelson calls the *docuverse*, and allowed users to view all the changes made to the document which led to its current form.

The Xanadu writing system also utilized transclusion (a term coined by Nelson): the ability to include part of a document into another by reference.

> Transclusion is what quotation, copying and cross-referencing merely attempt: they are ways that people have had to *imitate* transclusion, which is the true abstract relationship that paper cannot show. Transclusions are not copies and they are not instances, but *the same thing knowably and visibly in more than once place*. While copies and cross-reference are workarounds in place of transclusion, aliases and caches are *forms* of transclusion (Nelson, 2000). [Emphasis in original]

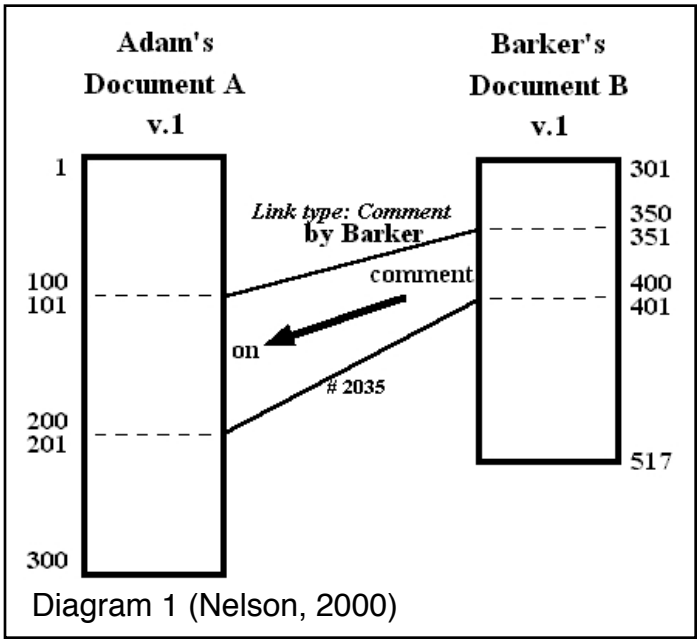


Diagram 1 (Nelson, 2000)

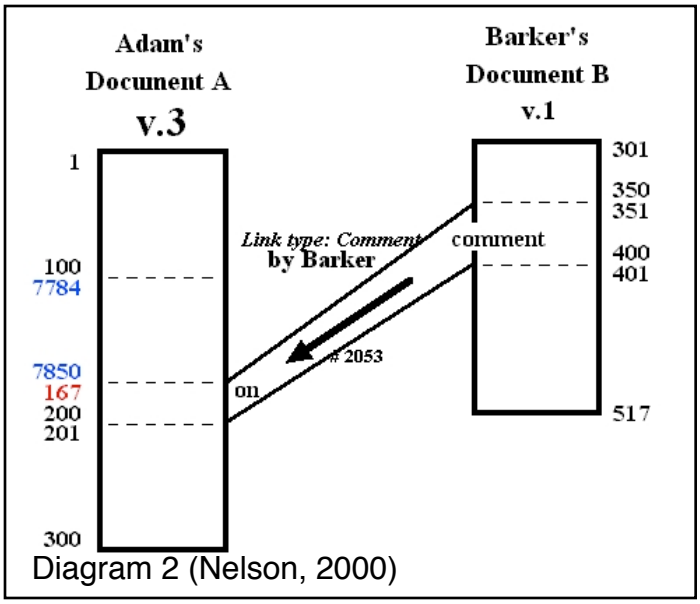


Diagram 2 (Nelson, 2000)

What makes transclusion so important is that it does not make copies of documents, it makes links. These links allow someone to go back and see where this information came from, as well as see what else links to it. This gathers more information in less time. Transclusion also allows users to keep the original data linked to something like a comment or another annotation.

Diagram 1 is a visual representation of one user, Barker, commenting on another user's document, Adam, by means of transclusion. The link in the middle shows that the link is a comment, but it also shows which part of Adam's document is

being referenced by Barker's document, in this case it is lines 101-201. Even if this document changes, these links will still be available; this is shown in Diagram 2.

Diagram 2 represents the same two documents as in diagram 1; however, Adam's Document A has been changed, either by Adam himself or by other commenters, to add lines 7784-7850. Although the file has changed, the links are still intact with the surviving characters of the new version.

Nelson's project has been largely considered a failure; nevertheless, there have been many other attempts to create Xanadu, and even today Wiki-technology is largely based on the Xanadu writing system.

3.2 The "Memex"

In 2000 a group of computer scientists created a "web companion" called "Memex."[3] Software Memex uses a combination of technologies from "hypertext data mining, browser plug-in and applet design, servlets and associated distributed database architecture, and user interfaces" (Chakrabarti, p. 670). What Software Memex allows a user to do is to ask the computer questions to recall information and links to information. These links are stored in the system using a variety of methods, and produces results based on the information stored. Software Memex uses bookmarks and browser history as a method of keeping trails.

> *Bookmarks **are** history.* Browsers make an unnecessary distinction between *history* and *bookmarks*. The prevalent view is that bookmarks are valuable links deliberately archived by the user, whereas history is the list of all visited pages, long and mostly useless, therefore suitable for purging now and then. However, the rapidly increasing volume-to-cost ratio of disks makes it unnecessary to discard *anything* from our personal browsing trails, if only the resulting data can be organized and analyzed usefully and effectively (features browsers do not provide).
> (Chakrabarti, p. 671) [Emphasis in original]

[3] For the purpose of this paper I will refer to this piece of software as Software Memex

Not only does it record and keep your browsing history forever, it learns folder structures using a complex "hypertext topic learning algorithm" (Chakrabarti, p. 671). This algorithm places a user's browsing experience in structured "topics" by placing new history entries in learned folders.

This is a large step forward from what systems do today; yet, it is still based on fairly old technology. The Software Memex is reliant on browsers and servers, and does not allow annotation at all. Bookmarking and browser history are all old technologies that only work for what they are designed to do, store a one-way link to outside information. If links change, the entire history is useless to the user. Although this is relying on HFS technology, it does show that there is technology available to lead us to the true AFS.

3.3 MyLifeBits

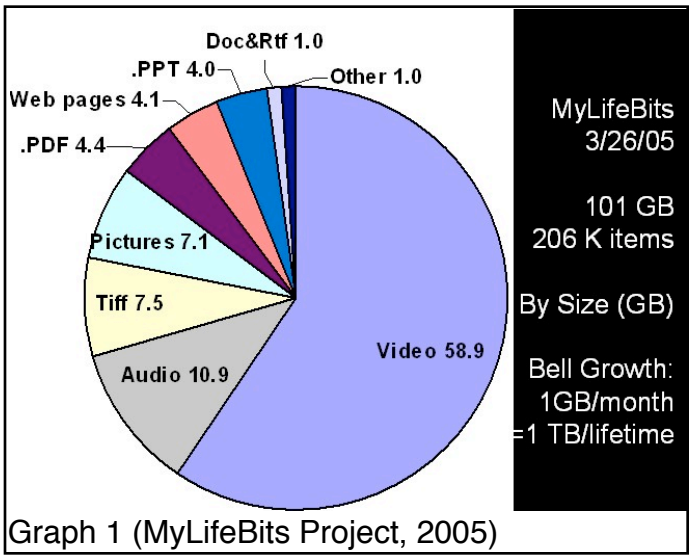
The MyLifeBits project started in 2001 when computer scientist Gordon Bell decided to go completely paperless[4] "doing away with a messy mountain of articles, books, cards, letters, memos, posters and photographs" (Bell and Gimmell, 2007). The MyLifeBits system uses a mixture of existing technologies--such as meta-tags, bookmarks, and databases; as well as advanced hardware technologies--such as a SenseCam; a camera developed by Microsoft Research that automatically takes pictures when its sensors indicate that the user might want a photograph.

> It is a system for storing all of one's digital media, including documents, images, sounds, and videos. It is built on four principles: (1) collections and search must **replace hierarchy** for organization (2) many visualizations should be supported (3) annotations are critical to non-text media and must be made easy, and (4) authoring should be via transclusion (Bell et al., 2002, p. 235). [Emphasis mine]

[4] Although the MyLifeBits project was officially started in 2001, Bell began digitizing his life in 1998

From this statement, the idea of MyLifeBits is very clear. As emphasized above, one of the goals is to "replace hierarchy" as the team found it difficult to work with[5]. The team realized the issue with HFS and switched to a database to store information as well as links and annotations; "Frustration with the file system led to testing the sustainability of databases for personal storage, and ultimately into research about next-generation storage systems" (Bell et al., 2006, p. 90). Using a database for personal storage is closing in on the idea of an AFS. Databases store data in a single place (the database) and allow linking of the information based on anything. Databases can store any amount of information and can store only one copy of the information[6].

Another problem faced with MyLifeBits was a matter of data storage. Images, videos, audio, documents, etc., all take up space on hard drives, what happens when those drives become full; moreover, how long does it take to fill the drives?



Graph 1 (MyLifeBits Project, 2005)

In 2007, just six years after the project had started, Bell had only amassed 150 gigabytes; this included 60 gigabytes of video, 25 gigabytes of images, 18 gigabytes of audio, and the remainder with e-mails, web pages, text documents, and presentations (PowerPoint documents). All of this information is stored across two computers, one of which is Bell's laptop (Bell and Gimmell, 2007). Graph 1 shows that the

[5] They even go as far a calling hierarchy a "straightjacket" (Bell et al., 2002, p. 236).

[6] SQL databases do allow duplicates but only if specified at creation; despite that, all databases require a 'primary key' (a unique identifier for each bit of information)

total storage in 2005 was 101 gigabytes, from 2005 to 2007 that number has only increased by about 49 gigabytes; 25 gigabytes each year. The volume-to-cost ratio of storage is so great today that it is practical for one to buy new storage if needed each year; even so, Bell's figures have shown that that may not even be necessary with the availability of such large drives. Filling these drives is another task that was found to be surprisingly difficult. According to Bell, a tera-byte drive[7] can hold more than 1 gigabyte per month for 80 years (2006, p. 90).

Although MyLifeBits is possibly the most successful project to fulfill the Memex idea proposed by Bush in 1945 to-date, the project is hindered by the fact that the system is, at the moment, hardware- and software-specific and fairly slow to search by today's standards.

After looking at these examples, one can get a good idea at what a true annotatable-AFS should include in it. Each project seems to tackle one or more of the issues we face today with current HFS technology; nevertheless, not one listed solves every issue with HFS. The next step is to use newer technologies that are more flexible, smarter, faster, and platform-agnostic to ac-complish the task.

3.4 Google

With the Internet, technology has advanced faster than ever before. Technology such as content management systems, personal websites, and web-crawling are still used today, but more advanced technologies are rapidly taking their places. Content management systems are becom-ing collaborative, open spaces (Wikis); personal websites are now blogs[8]; and web-crawling has been taken over by XML[9]. These new technologies are not only a method at viewing the web

---

[7] 1 Terabyte = 1000 gigabytes

[8] Blog software is a type of content management system, for now they will remain separate

[9] Extensible Markup Language: user defined document which encodes or serializes data.

faster and simpler, they are allowing users to do linking and annotation to the web. These technologies are considered Web 2.0 technology; that is, technology that uses "the web as a platform"(O'Reilly, 2005). What this means is that the Web is now like a computer system itself with applications and tools that are designed to run online as opposed to on the computer. What makes this idea of Web 2.0, the idea of using the web as a platform, so important to the creation of an annotatable-AFS?

The next sections will discuss how Google came from a simple annotation system to a platform of its own which offers annotation and referenced linking. It will also discuss how certain Web 2.0 technologies have changed how users can link and annotate information online, as well as share links with any number of people. Finally, it will show that with current Web 2.0 technology, it is possible to create the true, annotatable-AFS that Vannevar Bush envisioned.

Google was founded in 1998 by Stanford University PhD students Larry Page and Sergey Brin. According to Larry Page (2003), the original intention of Google was not searching the web, but a tool for rating annotations.

> It wasn't that we intended to build a search engine. We built a ranking system to deal with annotations. We wanted to annotate the web--build a system so that after you'd viewed a page you could click and see what smart comments other people had about it. But how do you decide who gets to annotate Yahoo? We needed to figure out how to choose which annotations people should look at, which meant that we needed to figure out which other sites contained comments we should classify as authoritative. Hence PageRank (qtd. In DeLong, 2003).

Google used a mixture of the two founders' interests; Page was interested in the link structure of the web while Brin was more interested in data mining as well as transclusion (DeLong, 2003). What Google essentially did was index the Web and rank it using the PageRank system.

> PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that we believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results.

> PageRank also considers the importance of each page that casts a vote, as votes from some pages are considered to have greater value, thus giving the linked page greater value...our technology uses the collective intelligence of the web to determine a page's importance. (Corporate Information - Technology Overview, 2008)

Not only does Google rank the pages, it uses "hypertext-matching analysis" to analyze the web pages content as opposed to its meta-tags. These Google technologies all allow faster indexing and searching online; billions of web pages can be searched in a matter of seconds. Technology this fast and this advanced has never been seen before, and this is likely just the beginning of what Google technology allows.

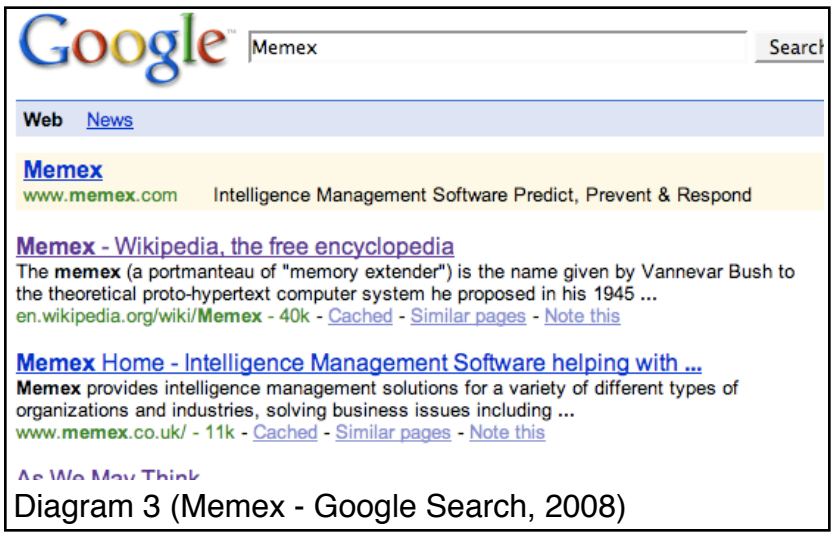


Diagram 3 (Memex - Google Search, 2008)

When searching on Google users are given a list of search results, these results will all contain the following options: "Cached," "Note This," and "Similar Pages" [Example shown in Diagram 3]. These Google features seem to have been created with the principles of the Memex, as well as Project Xanadu in mind.

Cached Pages shows a different version of the same document that has been stored in Google's cache. With this feature a user can view any changes to a page or document with a single click. This is strikingly similar to Nelson's *docuverse* in Project Xanadu.

Note This allows a user to make annotations on a website and store them in a notebook. This notebook is viewable online, and is saved online, not locally. Although these annotations

cannot readily be viewed by other users, this is a great step in giving annotation capability to web users.

Finally, *Similar Pages* will give the user a page that is similar to the one that is selected. This function was created with the idea to aid research; "If you are interested in researching a particular field, Similar Pages can help you find a large number of resources very quickly, without having to worry about selecting the right keywords" (Google Help: Search Features, 2008). This feature uses association to link pages as well as advanced indexing. Although the links are not personalized, this system is excellent proof that Web 2.0 technology is advanced enough to create an AFS. Personalization is the next step.

3.5 Del.icio,us

The Internet itself is a global network of computers, servers, and users all connected by digital communication methods, Web 2.0 took those communication methods and make them better. "Web 2.0...is not something new, but rather a fuller realization of the true potential of the web platform" (O'Reilly, 2005). Web 2.0 gave more power to it's users than the former "Web 1.0," by "harnessing collective intelligence" (O'Reilly, 2005). Not only are these technologies more advanced and intelligent, they can all be linked because they use the same technologies.

Hyperlinking has gone from creating a one-way link on a website to creating an organic trail of links to multiple sources. Links are created automatically by means of indexing algorithms and smart XML and RSS[10] feeds. A single website can link itself to an unlimited number of other sites by adding a single line of code. These links change when the information changes, and the links are usually stored in databases. These databases catalog information and allow us-

[10] RDF Site Summary or Really Simple Syndication: a feed from a website which gives a summary of updates

ers to link to it. SQL databases, PHP, and RSS all use transclusion to gather information; the information is simply read from a database and parsed elsewhere; it is not copied in any way. If the information changes, the references all change as well. Although databases are not a new technology, the means to get information from these databases are new.

Wikis use a mixture of all of these technologies to create a real-life AFS. Wiki's also use SQL databases to store information and links. These databases are called by the Wiki and information is parsed on the site as if it were written there when in-fact it is using transclusion. Not only do Wikis use transclusion to show information, they use association to link to other documents or websites. Wiki's also employ a document history for each document. When a user edits a document the database saves a copy of the document before and after it has been edited. Users can view the complete history of a document and see how it ended up in its current form. This is a direct version of Nelson's *docuverse.*

Diagram 4 (Popular tags on del.icio.us, 2008)

Many new Web 2.0 technologies use tagging systems as well. A tagging system allows a user to create keywords that relate to a bit of information which can be used to recall that information later. Tagging is an excellent example at associative linking; if a tag is called, every article that has that tag is called up.

Today, some websites use “tag clouds” as a form of navigation. Diagram 4 is an example of a tag cloud from the website *Del.icio.us*, a social bookmarking website. The words in the cloud are tags from users, the size of the word designates it’s popularity; so the tag ‘webdesign’ is much more popular than the tag ‘mp3.’ These tags are automatically taken from a database of tags from users and parsed in the manner shown. Each tag will take the user to a parsed list of bookmarks that use the specific tag. This is possibly one of the most innovate methods of associative linking online.

4. What is to be Done Next?

Taking in all of this information, the next step is to figure out what is to be done next. Looking back, it is obvious that Web 2.0 technology is far more advanced than any technologies that have been used in the past to make the Memex a reality; but, still it has yet to become reality. The technology to create the Memex is available now; the final step is to take technology from each example--from Xanadu to del.icio.us--and put it into one, complete system.

This system will not be an application on a computer, but a series of applications online. As stated above, to create a true, annotatable-AFS for the web it must be platform-agnostic. Web 2.0 technologies work on most any system, with almost any browser.

Information is no longer be stored in a strict hierarchy, but in more associative, natural databases. Today, these databases are already linkable, annotatable, editable, and employ transclusion right out of the box.

Coding has become more intuitive and now has the ability to parse any information it gets from a database in an easy to read fashion. The speed at which these systems work is unbe-

lievable, almost instant; billions of pages can be searched in a matter of seconds. The brain works the same way as a database; a user thinks of something (the information) and the brain (the database) searches through each record to find what that user may be looking for. This is what Bush wanted with the Memex, and this is what we have today.

5. **Conclusion**

Vannevar Bush wrote of the Memex in the 1940s because he realized that the system of storing and retrieving information was flawed. Over the past 60+ years several attempts have been made to remedy the HFS problem, but the technologies used only allowed these systems to get so far.

Nelson's Xanadu writing system was the first idea of hyperlinking; a form of association and transclusion. The Internet uses the principles of Xanadu, but only to a certain extent. To-date, the Wiki is the most advanced and near complete version of the Xanadu writing system; it employs document history, transclusion, and hyperlinking. The Xanadu was considered a failure mainly because of the lack of advanced technology.

The Software Memex was another step to allow users to annotate and create associative links. The web companion that was created kept a record of all online activity that the use did for as long as the storage lasted. The Software Memex failed in the annotation aspect of the Memex and was solely reliant on fairly old technology.

The MyLifeBits project is currently the best attempt to create a real annotatable-AFS. MyLifeBits is the Memex to the second-degree defined by logging all of a users life and allowing the user to store and view it very easily; MyLifeBits is *the* digital media Memex. Today, the

project is still active and; but, its downfall is its proprietary software and hardware as well as its speed.

The examples in this paper have each showed what is missing and what is needed to create the annotatable-AFS. Web 2.0 technology has solved much of the issues that plagued older systems. Web 2.0 technology is faster, more associative, more annotatable, easy to use, and platform-agnostic. Google has paved the way for indexing the web, annotating the web, and linking the web. Web 2.0 technologies are changing everyday; they are more advanced today than they were a year ago, and this trend seems to continue each year.

If we were to really create a Memex today, Web 2.0 would be the place to go for the technology. The Web 2.0 Memex would be the most advanced annotatable-AFS created, and it would only get better with time. All that is left to do is build it.

References

Bell, G., Gemmell, J., & Lueder, R. (2006). MyLifeBits: a personal database for everything. *Commun. ACM*, *49*(*1*), 88-95.

Bell, G., Gimmell, J., Lueder, R., Drucker, S., & Wong, C. (2002). MyLifeBits: fulfilling the Memex vision. *ACM Multimedia Conference '02*, December 1-6, 235-238. Retrieved Apr. 20, 2008, from http://research.microsoft.com/~jgemmell/pubs/MyLifeBitsMM02.pdf.

Bell, G., & Gimmell, J. (2007, Feb. 1). A Digital Life. *Scientific American*. Retrieved Apr. 20, 2008, from http://www.sciam.com/article.cfm?id=a-digital-life&colID=1.

Berners-Lee, T. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. San Francisco: Harper San Francisco.

Bush, V. (1996). As we may think. *interactions*, *3*(*2*), 35-46.

Chakrabarti, S., Srivastava, S., Subramanyam, M., & Tiwari, M. (2000). Using Memex to archive and mine community Web browsing experience. *Computer Networks*, *33*(*1-6*), 669-684. Retrieved Mar. 20, 2008, from the Science Direct database.

Corporate Information - Technology Overview. (n.d.). Retrieved May 4, 2008, from http://www.google.com/intl/en/corporate/tech.html.

Delong, B. (2003, Feb. 14). *Google and Larry Page: Archive Entry From Brad DeLong's Webjournal*. Retrieved May 4, 2008, from http://www.j-bradford-delong.net/movable_type/2003_archives/000032.html.

Google Help: Search Features. (2008). Retrieved May 14, 2008, from http://www.google.com/help/features.html#related.

Memex - Google Search. (n.d.). Retrieved May 14, 2008, from http://www.google.com/search?source=ig&hl=en&rlz=&q=Memex&btnG=Google+Search.

MyLifeBits Project - Microsoft BARC Media Presence Group. (n.d.). Retrieved Apr. 20, 2008, from http://research.microsoft.com/barc/mediapresence/MyLifeBits.aspx.

Nelson, T. (2000, May 23). *Xanalogical Structure*. Retrieved May 13, 2008, from http://xanadu.com.au/ted/XUsurvey/xuDation.html.

O'Reilly, T. (2005, Sep. 30). *O'Reilly -- What Is Web 2.0*. Retrieved May 14, 2008, from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Popular tags on del.icio.us. (n.d.). Retrieved May 14, 2008, from http://del.icio.us/tag/.