# $W^3$ + Structure = Knowledge

Giordano Beretta, Computer Peripherals Laboratory

**HPL–96–99***
June, 1996

World-Wide Web, www, web site, Internet, hypertext, publishing, HTML, HTTP, informal, opinionated

These days Internet and the World-Wide Web are the hottest topics in every walk of life. The Internet frenzy is like a rogue wave, you can either surf it and have the experience of your life-time, or stay on the beach to watch it and maybe get whacked by it.

It is easy to jump on the boat, but it is not clear where the long-term opportunities are. In an effort to see the big picture, we examine some desirable features and present some opportunities for interesting research. In addition to a methodology for structuring web sites, we propose a scheme for categorizing links.

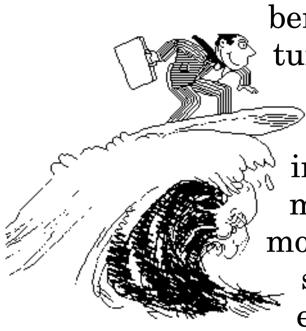
## Table of Contents

# 1 Introduction

Currently the Internet, the World-Wide Web (abbreviated as *web*, $W^3$ or *WWW*), browsers, applets, etc. are hot topics, not only in the technical community but in the society at large; this phenomenon is often referred to as the *Internet frenzy*. The technology is relatively affordable and is experiencing the same rapid acceptance as radio, automobiles, and compact disk recordings.

Like these technologies, the Internet is here to stay; it is not a fad like CB (citizen band) radio. The Internet has matured over thirty years,* during which some of the brightest minds in this century have contributed in creating a product of excellence. Because it is also a relevant technology, it will be woven in mankind's daily life as tightly and inseparably as running water and electricity.

The art of surfing is to spot good waves and then ride on them. From time to time there are big high amplitude and very long frequency waves called *rogue waves*. When such a wave hits the continental rim, one can swim to it and ride it out for the experience of one's lifetime, or one can stay on the beach, observing how the ridge of water curls over and breaks on the shore, getting whacked down by the wave. The Internet frenzy is a rogue wave and this is no time to stand on the beach.



Under the circumstances it is tempting to jump on the boat and reap quick financial benefits as the tide rises. It is not clear, however, where the opportunities for interesting long-term research will be when the technology is be established and commercial success depends on hard-core technology rather than on novelty and hype. Interesting solutions are those that scale well, and scalable solutions must be designed from the top down; bottom-up approaches are more successful when quick-fix solutions are sought. We examine some desirable features and outline some opportunities for interesting problems.

Events are happening in very rapid sequence and many time-scale comparisons have been suggested for such rapid growth rate (*e.g.*, one Internet year equals one dog year or one Internet year equals three minutes). The main reason why the Internet phenomenon is developing so rapidly is that most of the underlying technologies and the physical infrastructure are relatively old and therefore mature. Most of the work involved relates to engineering (*i.e.*, the application of well known principles and formulæ) rather than science (*i.e.*, the invention or discovery of new principles and technologies).

---

* Paul Saffo has observed that considering all major technologies introduced since the industrial revolution started, the expected time from invention to broad market acceptance for a new technology is thirty years. Statistically, the time is ripe for acceptance of the Internet by the masses.

We witness the coming together of new, powerful, and affordable technology systems; structural changes can occur very rapidly because all the components are already in place. Many people currently propelling the Internet frenzy have so much to catch up with that they have to move fast.[*] The important aspect is to recognize that this is a rogue wave and one has to act fast. Many surfers seek the best spot on the best wave, but only few have the mastery of identifying it at the right moment and to make the most of it. Experience is important and like the master surfer we have to learn from past frenzies of comparable magnitude.

Indeed, these are times similar to Venice in 1500, when Aldo Manuzio (*alias* Aldus Manutius, 1452-1516) joined a printing business and became a publisher. The adoption of Gutenberg's printing press, the brain drain of scientists from the collapsing Byzantine Empire bringing with them the Greek classics, an educated population that could read Greek, and a flourishing spice business whose profits allowed ordinary people to afford[†] books; all of these contributed to a vibrant, innovative environment.

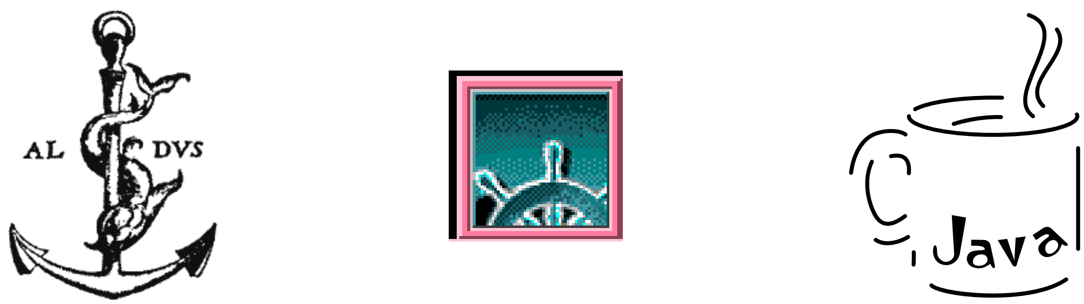



**Figure 1. Logos are intended to convey the image of a company. The Aldine publishing houses used the combination of a static symbol (the anchor) to symbolize slow and meticulous work with a dynamic symbol (the dolphin, a fast swimmer) to symbolize quick reaction to the market. Today, the Netscape site is based on a maritime theme, although nervous coffee themes are very fashionable at this writing.**

Manutius formulated a key idea that made him a main contributor to the Renaissance: he became a publisher instead of a printer.[‡] He searched for material and selected what he thought might have the largest readership. When he came across a classic text he thought might appeal to a wide audience, he had it translated from the Greek and published in Latin, and when he thought he might have a best-seller, he would even publish it in Italian, the common people's language.

---

* This is not a negative comment; rather, it reflects a different context. The early work was done by small commandos of scientists, in arcane programming languages, for experimental operating systems, and in research laboratories. Now the work is being performed by large armies of engineers, in common programming languages, for widely available operating systems, and in a production environment for the mass-market.

† The price of Manuzio's books was about a teacher's day's salary. Before, books could be afforded only by princes and wealthy monasteries. Compare this with the price/performance development of computers.

‡ The press was owned by an established printer, Andrea Torresano. Manuzio managed the printing shop, selected the texts to be published, made editorial decisions, and arranged for the marketing of the books.

To achieve quick financial gains, innovators have to move fast and conquer the challenging surf waves, laying their claims before their competitors, like the surfers compete for the best waves. As for $W^3$, there is no more time left for developing implementations; new ideas have to be implemented and distributed almost instantaneously, just as it happened in the financial industry when automated trading was deployed.

The lesson from Manutius is that although one can become rich by working hard operating a printing press, one can become wealthier by working smart and coming up with novel and bold technologies. The purpose of this report is to reflect on the current Internet frenzy and identify interesting problems that can produce opportunities for disruptive technological progress.

Two key properties of the Internet that have enabled the current explosive growth are that it is scalable and that it is based on open standards. To be successful, interesting problems must preserve both these properties. For commercial success, people often look for the *killer application*, like desktop publishing for the Macintosh and spreadsheets for the PC. Entrepreneurs now compete to be the first to discover the killer application for the Internet; various candidate applications are being investigated, such as commerce and vertical information markets.

On a more abstract level, there are broader ideas, like for example Netscape's list of "cool ideas" including superdistribution, ever-richer built-to-order, mobile languages, and nanocommerce* [10], or like Microsoft's idea of leveraging on the large installed Windows user base and the skills of their developers to launch a powerful suite of middle-ware development tools. Trying to go up by one more abstraction level, in this report we will write about what Phil Agre would probably call a story.

We will contend that the $W^3$ is about publishing. We have seen with Manuzio that disruptive technologies are non-obvious and require value judgements. Our value judgement as technologists is that we should help the average American citizen to transition from poster to publisher. To publish successfully one has to be able to articulate thoughts and ideas. We can accomplish this task by helping people to structure their information; in conjunction with the Internet technology and the existing $W^3$ infrastructure, this allows people to publish knowledge.

Mathematics has a well developed methodology to work with structures. In this report will only sketched some solutions based on graph theory. The next step will be to flesh out a strong and powerful system of axioms and then see which properties from the specific branch of graph theory are applicable. It appears that trees and in particular minimum spanning trees might play an important role.

* Nanocommerce refers to transactions worth between 1¢ and half millicent.

# 2 Problem description

*Information is no longer simply a strategic asset; it is a critical enabler of success*
*(Joel Birnbaum)*

## 2.1 The problem: information overload

In Appendix I at page 41, we mention some salient milestones of the Internet and $W^3$ starting with Memex. Vannevar Bush was not the first person confronted with information overload, which is not a consequence of the progress in science and technology in the first half of this century. Societies have bouts of rapid progress and prosperity every time a new means of open communication emerges. We computer technologists are very lucky to live in such a moment and to be among the few people competing from an advantageous position of strong relevant experience.

### *2.1.1 On the trend for open standards*

"Open" signifies it is owned by society at large and accessible by everyone. No individual or organization can exert control over it. A bout of rapid progress and prosperity is similar to a nonzero-sum game, while a stagnating society is like a zero-sum game. Control and secrecy are typical for zero-sum games, where winners always hide their strategies. Open environments are co-evolutionary; they are nonzero-sum games where winners announce their strategies in public so that other society members need to adapt to it.

$W^3$ is just such a new and open means of communication. Anybody with a computer and a modem can access it, and *de facto* the protocols are mostly in the public domain. For people with experience in the field, only a single action is required: act now! The $W^3$ phenomenon combined with the openness is such a sure bet with such a huge jackpot that it does not warrant much time for reflection. Two social forces are propelling the current trend to openness, one from the top and one from the bottom.

While in the 80s the computer industry captains at the top encouraged proprietary systems while bragging "I will win because I have the best technology," in the 90s the computer industry leaders have lowered their sights; strategic alliances have become the norm. This more pragmatic approach to business results in different approaches, where instead of trying to dominate the market with a new technology, companies will involve their competitors at an early stage. They seek for endorsement, and try to split the cake, laying claims only on the market slices that fall in the core competency domain of the particular company.

The force from the bottom has been shaped by concepts like the virtual corporation proposed in the Harvard Business Review and similar fora. The high-tech corporation of the 90s tends to have only a core of senior employees and all the practical work is delegated to contractors, hired only for a project's duration. Know-how spreads very fast from company to company as these modern journeymen wander from job to job like busy bees intermingling the genetic code of plants by pollination.

The wandering journeymen do not owe loyalty to an employer. Consciously or unconsciously, they will confer a less proprietary style to their work, while corporate cultures are disappearing in engineering departments. Modern journeymen are watching their career and have to prove their skills again and again to remain employable; they have to stay at the edge of technology, cannot fall behind or stick out their neck and produce creative ideas that are too extreme.

As companies have shifted their investment focus from employees to shareholders, in the 90s the technical workforce's merit increases have shifted from rewarding individual performance to the collective investments results of the employee's mutual funds. This is most visible in industrial research laboratories. In the 80s scientists spent their leisure reading and writing under the publish or perish regime; in the 90s the researchers they spend their free time playing the stock market.

The researcher's and engineer's collective values have a strong impact on what companies deliver and on how product mixes evolve over time. All this reinforces the trend to open standards because today's competitor can be the next employer, and because know-how rapidly diffuses from company to company where is encourages similar practices and makes it easier to agree on standards.

As stated earlier, this openness is a golden opportunity because it lowers the bar; anybody can play and win. The two important decisions are to join the game early and to start in the best early positions for a long game.

### *2.1.2 Information overload is not a new problem*

In the matter of information overload, a giant step forward happened with the Enlightenment, when the Encyclopedia was being conceived in France. Denis Diderot an Encyclopedist, wrote, in 1755:

> *The number of books will grow continually, and one can predict that a time will come when it will be almost as difficult to learn anything from books as from the direct study of the whole universe. It will be almost as convenient to search for some bit of truth concealed in nature as it will be to find it hidden away in an immense multitude of bound volumes.*

It was not different even in the middle ages, when 12$^{\text{th}}$ century scientists where wondering how to cope with the amount of information stored in the Benedictine libraries. And even Julius Cæsar, the inventor of the newspaper, in 48 BC felt overwhelmed by the hundreds of thousands of books found in the Library at Alexandria, which had been started only shortly before in 307 BC by Demetrios Phalareus to collect copies of all books in existence. Drowning in data, followed by starving for information is a cyclically recurring phenomenon.

From an entrepreneurial point of view, the challenge is to interpret correctly and diagnose the current state of affairs, identify solutions, go out and make a killing in the market place. In this situation of a nonzero-sum game everyone can win, so the risk is very low.

## 2.2 There is no evolutionary solution

A solution could be gleaned from Aldo Manuzio's example. Instead of a printer—who is in a business with single digit growth rates—one should be a publisher—a business with a double digit growth rate. The $W^3$ equivalent is to become not a poster of data but an Internet publisher, or a provider of services, equipment and tools to the publishers.


**Figure 2. Publishing information is more valuable than posting data…**

As we have seen earlier, publishing requires mining for data, identifying markets, selecting and editing contents, packaging, brand management, distribution etc. The publication market might offer many opportunities to sell tools and equipment.

Unfortunately, the market for such tools and equipment is not clear. In fact, the model that traditional publishers will just toss their old tools and embrace the $W^3$ is probably incorrect. The traditional publishing trade follows a very strict *workflow* pattern and does it under considerable time and quality constraints.

Publishing on the $W^3$ is very different because the content is different: it is multimedia. $W^3$ publishing requires skills in graphics, animation, music, video, etc., and, most of all production skills to effectively coordinate and schedule a multi-talented team of people. It might well be that the $W^3$ publishers will form a community closer in culture to the glitter of Hollywood than to the sober albeit elegant tradition of printed media. $W^3$ did not become a popular pastime when CERN released the $W^3$ protocols and contents; it took off exponentially only when NCSA released the GUI[*] browser Mosaic (see "Appendix I: Fifty years in the making", p. 41).

---

* Graphical user interface

Although districts in San Francisco and New York where $W^3$ publishing is flourishing have names like Silicon Gulch and Silicon Alley, technologists themselves will not create the new paradigms—or disruptive technologies as they are called now. Technology is just an enabler; paradigm shifts have more to do with social values.

In Manuzio's time, one major problem was the sheer size of books. He came up with some technological solutions, like inventing the italic type style that can be easily read at a smaller size, and folding the paper form (*folio*) into 16 sheets to reduce the dimensions and make books portable. But with these techniques the books of the time where still too voluminous.

Manutius could have used a technical solution, like publishing each work in several volumes. Instead, he called upon a value judgement; in his time, the largest part of a book was taken by the annotations, which could be several times the number of words in the original text. It was believed that the value of a manuscript depended on the annotations, and on the number and quality of the commentators. Manutius decided that his readers would read the classics for their own intrinsic beauty and the comments would be of interest only to the scholarly. He published only the original text.

Publishing a book stripped of the annotations was not an obvious decision in Manuzio's time, and this is exactly the kind of disruption that a successful technology for the $W^3$ must enable. In 1500 Venice general education and wealth had reached a critical mass; Manutius recognized the potential of the new market and came up with the critical ideas and technologies to redirect books from an erudite audience to the general public. In the current Internet climate many key technologies have been developed, but the contents question has not yet been solved. We are lucky to be left with such a golden opportunity.

### 2.3 What is information?

As Christine Borgman points out [6], researchers are currently taking two opposite approaches to understand the Internet phenomenon and to think out ideas on how to increase the system's value. One is to view it as a set of technological problems with humans in the loop, and one is to view it as a set of social problems pertaining to technology.

The first model tries to devise technologies to solve problems such as scalable data bases that can handle billions of documents, servers than can process millions of transactions per second, natural language interfaces, indexing and search engines, fast communications channels, etc.

In the second model, researchers try to understand how the $W^3$ can foster new virtual communities and what services should be provided. They also worry about intellectual property like copyright and authentication issues.

Phil Agre of the University of California at San Diego has outlined three models for understanding the meaning of information enabling in the $W^3$ that are in widespread use, which he claims to be all wrong [1]:

**Table 1. Phil Agre's stories about information.**

| Model | Designers | Users | Information |
|---|---|---|---|
| information processing | gods | factory machines | processed material |
| masculine transcendentalism | prophets | caught up in an inevitable rapture | the fabric of heaven |
| information professionalism | professionals | individuals with information needs | homogeneous stuff to be stored |

We cannot solve the issue of creating new virtual communities, but we are in a good strategic position if we can identify the tools that might be needed, and provide them. One of the linchpins in the process of turning information into knowledge is the categorization of information, *i.e.*, the introduction of *structure*.

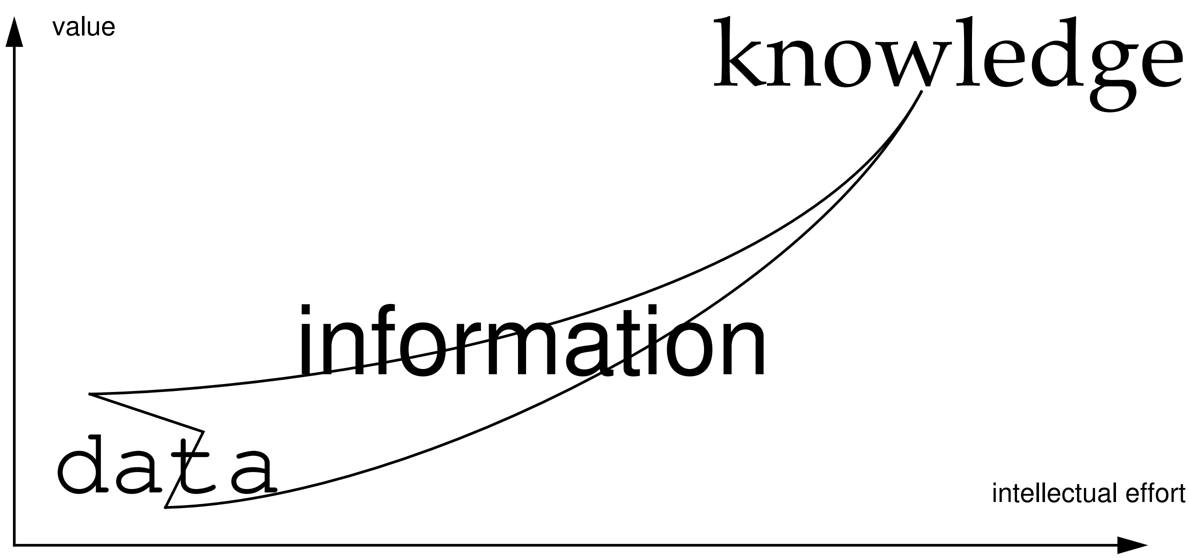



**Figure 3. …but knowledge is most valuable, because usually when we have a problem we do not know what to ask, what information is needed.**

This is not a simple task; for example keywording text can be learned, but indexing photographs is very difficult and expensive, especially if the iconography or meaning of a picture has to be established. As Christine Borgman points out, if we want to search for a picture of Dean Martin focussing on the actor's sex appeal, search engines will likely fail because the concept of sex appeal was not widely used in Dean Martin's time. Probably nobody would have used the "sex appeal" expression in a caption.

## 2.4 From chaos to order

Mathematics is the classical method to go from chaos to order. Mathematics is about structure (see Appendix II). The general method is to identify a structure in the set under investigation and then find all properties of this structure. This is even more powerful when an equivalent and even richer structure can be found. When the properties of the structure are identified, they are applied to the originally abstracted set to characterize this set in full. If the set is not tamed and controlled, the exercise is just purely æsthetical.

In terms of $W^3$, this means that for protocols like HTML and HTTP it is insufficient to just make them rich by adding numerous features and by making them consistent. The protocols must be such that they can radically increase value, where "radically" means that value must be increased by at least an order of magnitude.

At this early stage, protocol requirements must be stated boldly. If proposals are modest, the evolution of $W^3$ can get stuck "in a local maximum," *i.e.*, at a limited (or inadequate level). If parsimony is not deferred until maturity has been achieved, the system will be mediocre and may actually slow down the adoption of technology. For the best surfing experience it is important to catch the ridge of the wave; there is not much room for the timid, the risk takers will have the greatest satisfaction.

There is an exponential increase in value for the consumer when data is organized into information and information is digested into knowledge. The structure must be such that knowledge can be produced at the smallest cost. Furthermore, the structure must be universal, so it can be applied globally and uniformly to the technology. To paraphrase Xerox' recent slogan [2], the technology has to

- make $W^3$ publishing better
- make better $W^3$ publications
- allow to work better with $W^3$ publications.

# 3 Terminology and representation for $W^3$ links

## 3.1 Terminology

The elementary datum on the $W^3$ is an *object*. A structure is induced by the generic set of all names/addresses that are short strings that refer to objects. This set is called *uniform resource identifier* (URI). For brevity, we will call the elements in an URI *links*.*

There are two kinds of URIs, persistent and transient. A transient URI is called a *uniform resource locator* (URL) and consists of a protocol (*e.g.*, http), an optional user name/password pair separated by a colon and followed by an @ (at) sign, an Internet host name (*e.g.*, www.hpl.hp.com), a file path (*e.g.*, personal/Giordano_Beretta/Literature.html), and a paragraph name (*e.g.*, #542). For this example the syntax is as follows: *http://www.hpl.hp.com/personal/Giordano_Beretta/Literature.html#542*.

Any URI that is not a URL is called a *uniform resource name* (URN). URNs are supposed to be more persistent than URLs. Schemes for URNs are currently under development by the Internet Engineering Task Force (IETF).†

A similar construct is the *uniform resource citation* (URC). However, a URC is not a URI; it is a data structure of attribute/value pairs describing an object. The values are URIs and other information such as, for example, authorship, publisher, data type, etc. In publications by the World Wide Web Consortium (W$^3$C)‡ the URC is represented as "a mechanism of resource description, which can be seen as an instance of the general problem of knowledge representation."

Usually with HTTP entire files are transferred and the contents of such a file is called a *web page*. The pages in a directory and its subdirectories are called a *web site*. In the root directory of a site there is a designated file called the *index file* and its contents are called the *home page*. If an explicit index file is not present, HTTP creates an implicit index file consisting of a directory listing.

Following a sequence of links is called *browsing*, and a collection of browsing operations is called a *session*. A *browser* is a program that uses a protocol to follow links and display the pages indicated by the link. Browsers usually keep a *history* of the links executed in sequence.

In sum, the $W^3$ consists of a number of paragraphs and their identifiers, which are a quadruple set consisting of a protocol, an Internet host name, a file path, and a para-

---

* It is not completely clear to this author why URI refers to the set of all links instead of referring to a single link. In many documents the UR* terms are used to indicate a single link.

† http://www.ietf.cnri.reston.va.us/

‡ http://www.w3.org/hypertext/WWW/Consortium/

graph in a file. The contents of a file on the web is page and the pages are grouped into sites.

### 3.2 Graphs

In essence the $W^3$ consists of an arbitrary relationship among data objects. Directed and undirected graphs are natural models of such relationships (see [14] for more details). A *graph* $G = (V, E)$ consists of a set of *vertices* or *nodes* $V = \{v_1, v_2, \ldots\}$ and a set of *edges* or arcs $E = \{e_1, e_2, \ldots\}$. A pair of vertices corresponds to each edge; if edge $(v_1, v_2)$ corresponds to edge $e$, then $e$ is said to be *incident* on vertices $v_1$ and $v_2$, and $v_1$ is *adjacent* to $v_2$. When information is attached to the vertices and edges of a graph, it is called a *labeled graph*.

A graph is *directed* if the vertex pair $(v_1, v_2)$ associated with each edge $e$ is an ordered pair. Such a graph is called *digraph* for short. A weight can be associated with an edge of any graph, indicating for example the cost to go from a node to another, the time it takes, or the degree of difficulty to understand the contents of a node given another node. Fig. 4 shows an example of a weighted digraph.

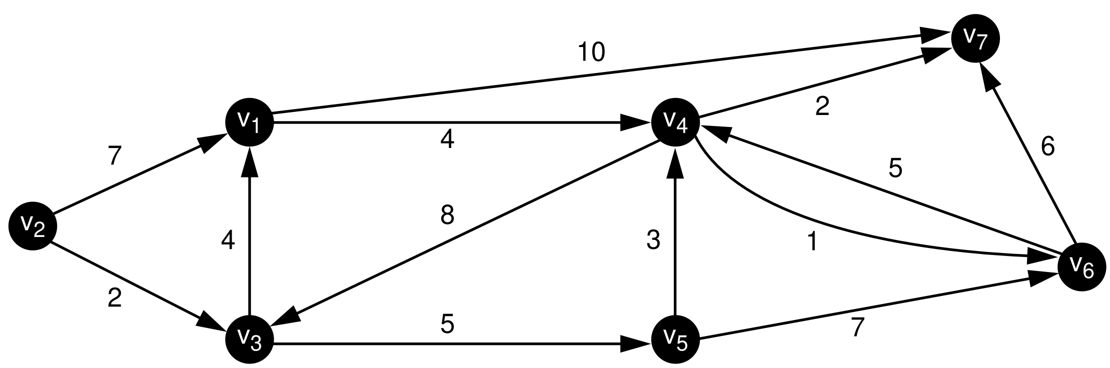



**Figure 4. A simple weighted digraph. Note that there are two arcs between nodes v4 and v6; they illustrate the frequent situation where the cost of going in one direction is different from the cost of going in the opposite direction. The absence of an ordered pair in the opposite direction can be interpreted as an infinite cost or impossibility.**

A *path* $(v_1, v_n)$ in a graph is a sequence of adjacent edges $(v_1, v_2, \ldots, v_n)$. In a digraph this path is said to be *directed from* $v_1$ *to* $v_n$; in an undirected graph this path is said to be *between* $v_1$ *and* $v_n$. The number of edges in a path is called the *length* of the path. The *distance* from vertex $v_1$ to vertex $v_n$ is defined as the length of the shortest path from $v_1$ to $v_n$. A path is *simple* if all vertices on the path, except the first and last, are distinct. A (simple) *cycle* or *circuit* is a (simple) path of length at least one that begins and ends at the same vertex. A graph that contains no cycle is called *acyclic graph*. A directed graph that contains no cycle is called *directed acyclic graph*, or *dag* for short.

A *subgraph* of a graph $G = (V, E)$ is a graph whose vertices and edges are in $G$. The subgraph of $G$ *induced* by $S \subseteq V$ is the subgraph of $G$ that results when the vertices in $V - S$ and all edges incident on them are removed from $G$.

An undirected graph is *connected* if there is at least one path between pairs of vertices in the graph. A digraph is connected if the undirected graph obtained by removing the edge directions is connected. A digraph is said to be *strongly connected* if for every pair of vertices $v_i$, $v_j$ there exists at least one directed path from $v_i$ to $v_j$ and at least one from $v_j$ to $v_i$.

A connected, undirected acyclic graph is called a *tree*, and a set of trees is called a *forest*. From the previous paragraph it follows that

> an undirected graph $G$ is a tree if and only if there is exactly one path between every pair of vertices in $G$.

Thus trees can be thought of as graphs that are minimally connected. A tree that is a subgraph of $G$ and contains every vertex of $G$ is called a *spanning tree* of $G$. If a weight (or cost) is associated with each edge, the spanning tree with minimum total edge weight is called a *minimum spanning tree* (MST). A typical application for MSTs is in the design of communications network and many interesting results have been published.

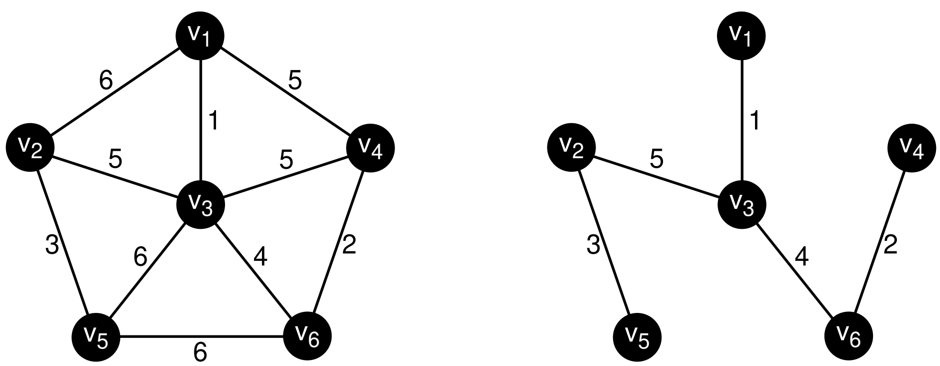



**Figure 5. A graph (left) and a spanning tree (right) for the same graph. The numbers on the edges represent their weights; the weight of a path is the sum of its edges' weight. An example of a weight is the cost to go from one edge to the other.**

Two graphs $G$ and $F$ are said to be *isomorphic*, $G \cong F$, if there is a correspondence between the vertices of $G$ and the vertices of $F$ that preserves the adjacency relationship (see Appendix II, page 45). Isomorphic graphs differ only in the labeling of the vertices.


HP

# 4 Sketches for solutions

We now discuss some technological artifacts that can be deployed to introduce structure in the $W^3$. The focus will be on aspects related to protocols, Internet printing, and web photography.

## 4.1 Intrinsic $W^3$ structure

By intrinsic structure we mean the basic structure that is minimally necessary to define the $W^3$. Each named paragraph in each file on an exported path on an Internet host is a vertex of a graph labelled by the URI. Each link, e.g., an occurrence of an URI element, is an edge from the vertex in which it is declared to the paragraph into which it points.

The links perform the “go to” function and there is no “come from” function, so it is clear that the $W^3$ is a digraph. However, the trail visited in a session can be viewed as an undirected graph when combined with the history trail, because the history trail can be used to backtrack each visited link.

There are two interesting subgraphs. One is the graph corresponding to a web site and one is the subgraph visited during a session. The goal for a publisher is to design a well-structured site graph, while the goal of a user creating a session is to succeed in obtaining the knowledge sought. One peculiar property of a sessions graph is that, because the session history is usually a implemented as a linear list, the browser software can efficiently remove cycles from the session graph creating an acyclic graph, although it is not clear that this behavior induces a good user model.

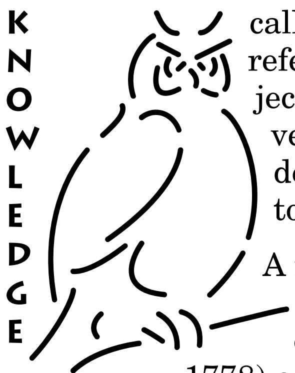
We are interested in identifying good methodologies to design site graphs. The session graph can be used to determine the quality of a site graph because usually a short trail is better than a long trail, having contributed to build knowledge more efficiently. A more accurate measure can be obtained by associating weights with the edges in the graph. An example for weights is the character position of the link in the HTML file, which is an indication of the amount of text to be read before the link can be exercised.

## 4.2 Maps on collections of $W^3$ pages

### *4.2.1 Example*

Suppose we want to create a site that can deliver our knowledge about natural history. It will have pages on the physiology of blood circulation, the nervous and endocrine systems, humans, orcas, condors, apples, spiders, etc. How can we organize the site?

Since we are creating a hypertext system and not a book, we will write a self-contained article on each subject. It is immediately clear how we can create hypertext links among the articles by making references active so users can go to them by clicking on the text. But this is a pedestrian use of hypertext; it is like creating the index for a book. The index is related to information. It is the table of contents that is related to knowledge, and devising a good table of contents is difficult.




The easiest way is to sort the articles lexicographically, *i.e.*, alphabetically by subject. The result will be a site that is like an encyclopedia, a reference tool that quickly gives a succinct answer on a particular subject. Although there are links to other articles, this site would not be very effective at conveying knowledge, because the lexicographic order does not necessarily induce natural structure in, *e.g.*, natural history.

A more useful organization would be a classification based on symmetry of the body and metamerism (the repetition of body parts). This criterion was introduced by Carl von Linné (*alias* Linnæus, 1707-1778) and forms the basis for descriptive and systematic zoology and botany. This is essentially a tree where the leaves are called "species," and group beings that are very similar. Species that are similar are grouped into a "gender," similar genders are grouped into a "family," and so on with "order", "class," "type," etc.

An organization that may build more knowledge in the same set of articles is to follow the evolution of anatomy and physiology. The articles could be organized by the various systems (skeletal, muscular, tegumental, endocrine, etc.) and within each one can be structured by how it evolved to a more sophisticated and specialized structure as species mutated.

A completely different organization could be based on biotopes, that is, on how beings co-evolve in a habitat and how their existence is interrelated. It should now be clear, that—based on the same set of articles—many more organizations are possible.

The articles are vertices of a graph and the organization are the edges. For each set of vertices there is a number of possible graphs; although all graphs contain the same amount of information, they convey very different amounts of knowledge.

There is a second, superimposed graph that consists of the references equivalent to the index entries in a book and which we will call *index edges* to distinguish them from the *contents edges*.

We can now partition each article into a number of components based on the expertise of the audience; for instance an introduction for pupils, a summary for biologists, a general description for the educated reader, latest results for the scientist working in the field, etc. For each graph there could now be an embedded graph that points just at the portions of an article that are relevant to a given audience.

### 4.2.2 Generalization

We consider the maximal graph obtained by connecting all vertices to each other. By virtue of the axiom of specification in set theory, we can devise a condition that starting from this maximal graph deletes edges until a particular graph is obtained. Thus, a particular $W^3$ site is a *map* from the maximal graph to a subgraph. The maximal graph represents the *information* available at the site, while the subgraph represents the *knowledge* that has been compiled at the $W^3$ site.

The problem is to find maps that maximize the knowledge communicated to a particular reader, or, as expressed by Ho John Lee [11], the "identification and imposition of structure [that] turns information (facts) into meaning (interpretation and facts)." From a mathematical point of view (see page 45), this requires identifying a good system of axioms and then analyzing the structure of the model, which can be based on the graph construct.

Unfortunately, establishing the system of axioms entails dealing with such concepts as trust and credibility. This is a task beyond the scope of this report and for the remainder we will limit ourselves to outlining a methodology for identifying good maps, instead of a rigorous system of axioms.

### 4.2.3 More examples

Referring to Vannevar Bush's idea of associative indexing (see page 41), consider the set of vertices visited by a user browsing the maximal graph during a session. This browsing operation defines a new map of the collection, and as long as the user stays in the site the image graph is a subgraph of the previous graph, *i.e.*, the map is an endomorphism (see page 45).

We can also look at how the $W^3$ pages are actually stored in a global hierarchical file system. We obtain a new map, but this time its graph it is no longer necessarily an endomorphism. This map is very visible to the user in the location field and various menus and status fields, as it is shown in the hierarchical structure of the identifier for an URL.

If a search with a set of keywords is performed in the local site, the operation will be a new map, whose graph will probably be unrelated to the previously described graphs, depending on the contents of the query. Such a map is probably not interesting in itself, but can be the point of departure for a new rich map.

A publisher can create a new set of hyperlinks by connecting differently a collection of $W^3$ pages, thus creating an edited map. If the publisher is skilled, this can be a way to create added value to a collection of data. This map by itself is a marketable item, its value depending on the publisher's reputation and authority.

Thus, in a collection of $W^3$ pages there is more than just the hypertext structure induced by the reference to URLs. There is an arbitrary number of maps, and the hypertext is just an instance of such a map. A map can have interesting properties, such as producing a minimal graph, which could indicate that knowledge is represented in a compact way without redundancy.

The map can be implicit or explicit. In the last example, an explicit map would be a home page with links to the contents selected by the publisher—possibly annotated—much like the table of contents in a magazine. In an implicit map the user just indicates readiness for the next page, and the publisher delivers the next "byte" of information, either statically or dynamically, based on the user's profile or past trail.

We will now explore some methods for structuring web sites.

### 4.3 Ariadne's string

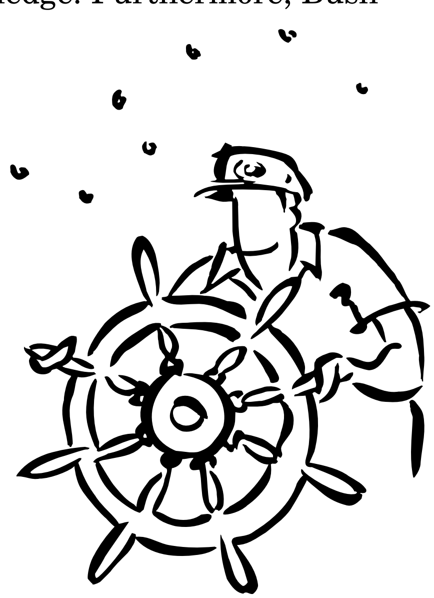
When we apply a criterion to filter a subset from a large set of data, we create information. If it is just presented it as is, the only advantage is that the quantity of data has been reduced and made more approachable for the consumer (or reader). However, it does not help the consumer to gain new insights, *i.e.*, it does not contain knowledge. To convey knowledge, the consumer must be presented with a *string of thought*. Gottfried Wilhelm Leibniz (1646–1716) wrote

> *The true method must provide us with a* filum Ariadnes*, that is to say a kind of sensitive and coarse means that guides the mind, in the same way as lines drawn in geometry and the type of operations that are prescribed to apprentices in Arithmetic. Without that our mind would not know how to go along a long path without straying.*

For Leibniz, the author finds a linear order in a set of information and creates knowledge in the form of a linear string. A different aspect of knowledge based on the same information requires a complete duplication or transcription of the information. Vannevar Bush's contribution though hypertext (see Section 6.1 on page 41) has been the separation of Ariadne's string from the underlying data, so that the data can be traversed in a number of strings for different aspects of knowledge. Furthermore, Bush generalized the linear string to a graph.

### 4.4 Avoiding to get lost

The lesson learned from early computer aided instruction (CAI) systems like Plato is that the user can easily get lost in a web of information. This produces frustration and the system is abandoned once the novelty effect has faded. Nievergelt and Weydert have identified the main questions that characterize the difficulties experienced by users of interactive systems [13]:



- Where am I?
- What can I do here?
- How did I get here?
- Where can I go, and how do I get there?

The goal is to devise structures that always lead to answer these questions. Nievergelt and Weydert have proposed a framework based on the concepts of site, mode, trail and trail editor that mirrors the questions above:

- Site: a neighborhood in the space of data, consisting of those data items to which the user has direct access at a given moment.
- Mode: a subset of the set of commands, consisting of those commands that are active at a given moment.
- Trail: a feasible time-sequence of pairs <current site, current mode>.
- Trail editor: supports the conventional help function (inspect and extrapolate the user's current trail) and conventional command macros (trail editing and reusing past trails).

Information about where a user is situated is valuable only if the structure of the navigable data is systematic. It turns out that users get easily lost if the structure is a generic graph; cycles make it most difficult to stay on course. A hierarchical structure like a tree has proven to be the best solution [4]. At each node there is always a parent node, there are sibling nodes representing alternate sites, and there is a root node from which exploration can always re-start. As we have seen in Section 3.2 on page 16, a tree has the desirable property that there is exactly one path between any two vertices, removing any navigational ambiguity.

A tree has a partial order. It can always be linearized by enumerating its nodes in one of a number of orders, such as depth-first, bottom-up, or level order [14]. During an enumeration an action can be executed at each node, such as displaying the neighborhood in a window or printing the contents of each node.

### *4.4.1 Canonical tree*

Ideally each collection of $W^3$ pages would be organized as a tree. This tree does not necessarily have to be implemented as a set of hypertext links in the collection. It is just another map and it can be stored independently of the $W^3$ pages, or without changing their contents.

There is currently no way to enforce a tree structure on a collection of $W^3$ pages, but ideally authoring tools would enforce, or at least encourage such a structure. Most documents have some sort of structure and this structure is hierarchical. For example, in this report there are a title, a number of sections, which are composed of subsections, and so on to paragraphs, sentences, words, and characters; the table of contents is a graph in this report, namely what we called earlier the contents edges.

If no such structure can be identified, one could always be created artificially, for example by generating the coding tree from the URLs or file names of the $W^3$ pages. More interesting is the opportunity to exploit the graph structure of the collection of $W^3$ pages and find semantic equivalents to properties like the minimum spanning tree (MST).

Returning briefly to the table of contents of this report as an example for a canonical tree, note that books usually have additional maps in the guise of indices, which we discussed on page 20. At this point of the discussion it is clear that we should be able to categorize links so that various maps on a collection can be clearly identified and separated.

### *4.4.2 Printing issues*

The existence of a tree structure greatly simplifies the printing task, because now the user would not only be able to print a $W^3$ page, but also to print a subtree by enumerating in depth-first order and printing the contents of each node as it is visited.

The problem is that the user no longer gets an idea of the size of a print job when a print command is issued. This is not only the case when a subtree is printed, but can occur even in the case of a single page. If only the current page is printed, the fact that web printing is not WYSIWYG* prevents the user from getting a feel of the size and scope of a print job, as soon as the page is larger than what fits in the user's browser window.

There are no issues with printing text and vector graphics, all problems have been solved for years with the use of style sheets [3]. Sampled images, instead, should be sent twice, at screen resolution (*e.g.*, 72 dpi or dots per inch) for rapid display and at a resolution matched to the print screen (*e.g.*, 200 ppi or pixel per inch) for adequate quality printing (see [5], Fig. 2 on page 15). Since images are two-dimensional, the time to transmit an image grows quadratically with height, not linearly. This makes it harder for humans to extrapolate data size from the screen display and the elevator in the scroll bar.

Still in the case of printing a single HTML page, if the page is long and the link is slow, the user may issue a print command before the whole page has been received to print it concurrently. In that case the user has no way to know how long the document print job is.

---

* What You See Is What You Get. HTML is about document structure, not layout, so formatting is adjusted dynamically to the current window size, which is usually not proportional to the printable area on a sheet of paper, especially if the user has a small display monitor.

If the user is allowed to print a subtree, it is clear there is no reliable way to gauge the size of a print job. It should be kept in mind, that the bottleneck is not necessarily a slow modem; even when user uses a cable modem connection or a T-3 line, the server or the Internet service provider's (ISP) gateway or even the Internet backbone can be congested.

Humans are much better at making value judgements than computers, therefore it would be a highly desirable HTTP feature if the server notified browser clients of the number of printed pages for each web page and subtree. Browsers could do it locally without changing the HTTP protocol, but this would require sending the data over the Internet—a huge waste of bandwidth resources when this is done by millions of users.

Scenarios indicating the value of the size hint:

- Some $W^3$ authors put a page size at links to large HTML pages; sometimes they forget to update the size when they change the contents, or make calculation mistakes. Computers are better at this kind of task than humans. Moreover, if the hint is part of the protocol, it would be a universal feature and browser clients might provide a better user interface if the information is encoded in a machine-friendly format.
- The user might have a fast 300 ppm (pages per minute) device set as the default printer and sitting at the root of a large subtree (e.g., a phone list or a large parts catalog). Before anybody notices, many reams of high quality paper can be wasted (this happens more frequently than one might suspect). People tend to waste less paper if a large print job generates a warning with the number of trees being consumed.
- Many people have a slow personal ink jet printer connected locally to their workstations. They have a low tolerance for long print jobs and often start printing before leaving work in the evening, leaving the printer running unattended for the night.

By mentioning on the previous page the concept of printing a subtree, we assumed the fact that printing is a disconnected task from browsing. Currently the $W^3$ protocols allow printing only in the form of producing a hardcopy of the currently displayed $W^3$ page. It is clear that this simple mechanism will be replaced just as fast as printing documents by sending a display bitmap to a printer was superseded by page description languages.

Professional publishers in particular need control of the layout and typographic display for their $W^3$ pages. The display includes such capabilities as using a rich variety of fonts. At this writing, the font issue is hotly debated because of the intellectual property issues involved and the size of the font data. Several caching schemes are being discussed to attack this problem.

If printing becomes an independent background task, the $W^3$ protocols can be extended to allow a negotiation of the transmitted document's reproduction parameters, in a similar fashion as this is accomplished in the facsimile transmission protocols. An independent printing task with capabilities negotiation can easily be generalized to incorporate *pro tempore* licensing or rental schemes for fonts, as well as clever downloading schemes that minimize the storage requirements for fonts in spoolers and printers.

### 4.5 Ability to mark & categorize links

Currently there are three types of nodes: global to $W^3$ (hierarchical identifier), local to the $W^3$ site (*i.e.*, same URL except for the last hierarchical identifier, so only the latter is specified), or local to the page (URL appended with an pound sign and an identifier called “name”).

This syntax reflects the location of a hypertext anchor: on a different host, in a different file, or at a different character position in the same file. While this is a necessary requirement to implement navigation at the physical level, it is an implementation detail irrelevant for both the publisher and the user.

We have seen earlier that is it useful to be able to distinguish the contents graph from the index graph of a site. In the following we will elaborate on this idea.

From a publisher’s and user’s perspective, it is desirable to be able to able to *mark* links and to categorize them. Marking is a boolean operation, either a link is marked or unmarked, with unmarked simply meaning “forget.” When a trail is saved or exported, unmarked links are simply ignored, sort of a garbage collection of uninteresting[*] links.

When a node is marked, then a facility should exist to *categorize* the nodes. Examples for categories could be a main thread through the information and cross-references. As we saw earlier in Section 4.2.1 on page 19, there can be many different “stories” around the information in a site, such as the different view of natural science. We would like to categorize the links so they can easily be discriminated and presented alternatively to the browser.

A suitable facility could be that used to designate classes in object based systems, such as the *symbol* syntax element in Lisp or the *atom* syntax element in Cedar. There is no predefined set of categories; a new category can be declared at any time and added to the existing set.

Categorization is hard and it is where the value added brought in by publishers would come into play. It could be supported by a simple mechanism consisting of a <user name, time stamp> pair, which for example financial consultant Jane Doe could publish after examining a Wall Street site and marking her picks. The trail of marked sites could then be published as “Jane Doe’s picks of the day.”

* Uninteresting is meant in the non-judgemental manner of not pertinent to the current map.

A node can be marked by two entities: the publisher or creator of the HTML document and the reader or browser of the document. There is a correspondence between maps and categories. For example, the canonical map is a category and the marks have values like root, sibling, child parent. Non-linear links are gotos and in a paper, for instance, would be cross-references that can be indices, glossaries, literature references, comments, etc.

Other examples of categories could be: note, detail, raw data, graph, literature reference, important, confidential, etc. It would be useful to have a *convention* for common atoms. This would ensure that Jane Doe's raw data is in the same category as John Doe's raw data. Each category groups the links belonging to a map.

### *4.5.1 User interface*

The user interface for marking is straightforward; all that is needed is a button in the menu bar. For the user, the operation is as easy as saving a bookmark in today's browsers.

Assigning a category could be implemented with a pop-up menu, where the user could select a category for the ones currently defined or enter a simple dialog to create a new category. This also solves the problem of not knowing the user's terminal bit depth, and works also for the chromatically challenged.

For displaying categories we propose the currently misused color coding facilities for URLs. Today HTML allows to use a variety of color pairs to display URLs, one color if a URL has been visited and one when it has not been visited. Today a page designer can assign the color arbitrarily, which is very confusing for the user and worse than using no color encoding at all.

The user of a browser should be the person to map colors and categories, not the page creator. The mapping would be similar to the "file kind" facility in the MacOS, but in reverse: a (variable) color is assigned for each (fixed) category instead of giving a name tag to each color code representing a file kind. There would be an unlimited number of categories, and a window with a table showing the color mappings would visually aid the reader in recognizing link categories.

### *4.5.2 Implementation details*

Atoms are straightforward to implement. They are represented externally as character strings (possibly using a two-byte character encoding, making it easier to read non-Latin plain text). The strings are stored in a symbol table resident in the browser and the hash function would not be defined, *i.e.*, the browser implementors can use a function of their choice (avoids hard-wiring a word size in a standard). This implies that symbol tables are always transmitted an character strings (if they are transmitted).

In addition to the character string representing the atom, each entry in the symbol table can also have a list of attributes for quantities like names, time stamps, etc. An attribute is a <name, value> pair and maybe each attribute value should allowed to be a regular expression.

An alternative to symbols would be to use universally unique identifiers in conjunction with a central *registry*, as this is done for example for RPC (remote procedure call) sockets or IP addresses. In applications where human judgement plays an important role, mnemonic identifiers based on conventions have proven very successful, such as for identifying object classes in software frameworks and operating systems. The only condition is that a given identifier must always be hashed into the same symbol table key.

Graphic characters are used as delimiters in URLs. The characters currently used are the slash for global URLs, no character for local URLs, the colon to separate the user name from the password, the @ (at) sign to separate the user name from the host name, and the # (number) sign to delimit locations local to a page. A $ (dollar) sign* could be used as a delimiter for the category in an URL (browsers just ignore tokens they do not recognize, providing backwards compatibility).

It is probably not necessary to do anything for this extension to HTTP, because the symbol tables are local to the browser. If it should be necessary or convenient to preload a symbol table, this can always be done easily through an applet. The disadvantage of this method is that the symbol table is finalized only when the entire HTML page has been received and parsed. If it is desirable that the GUI for the symbol table be available to the user immediately, the symbol table would have to be declared between the header and the body part of an HTML. File and authoring tools would have to insert such a declaration.

## 4.6 Obsolescence/perseverance of trails

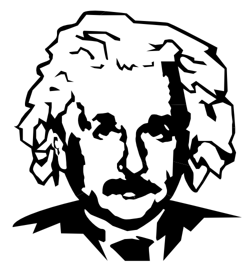
The operators of Digital Equipment's Alta Vista search service found that the average lifetime of an URL is only 45 days [6]; there is a problem of data obsolescence. How many dangling URLs do you have in your bookmark file? This ephemeral character of web pages is not the only problem; the URL itself is not a natural constant like π, and not even a permanent official key like a social security number or a vehicle identification number.

It is easy to imagine how a file path that can be altered by moving around a few icons on a desktop is a very ephemeral addressing scheme. It is less obvious, that the host

* The dollar sign is the delimiter for atoms in Cedar.

name in an URL is just as easily changed. In fact, a host name is just a key to a host table, where an IP address is stored. The entries in a host table can be changed at any time, for example for such mundane tasks as redirecting TCP/IP traffic to a physically closer mirror site. Moreover, the IP address is just a parameter set in a computer and can easily be transferred to a different machine.

In a sense, a dangling link is benign, because it triggers an error message. A link that is reassigned can be more annoying. Here is an example of such a link, where the link is a literature reference. For the event of the New York World's Fair, the physicist Albert Einstein was asked to write a short note on our times. The note was to be included in a time capsule to be buried at the Fair. As customary for scientists, Einstein sent a draft for the note to various friends asking them for comments. This draft* went into the history books, but the final version of the note might remain a lost secret until the time capsule will be opened.



This problem was very frequent before the widespread use the printing press because in the manual transcription of books the copyists drifted from the original text, which was often lost or unavailable. While the printing press made it possible to print large numbers of copies, reducing the probability of total losses, the digital technology with its capability to store a large amount of data in a small and inexpensive volume could make it possible to save for posterity not only the final result, but also its unequivocal genesis.

Pages in a trail should be cached because they constitute historical information leading to knowledge. If this is not a default (because of storage limitation problems on small computers), it should at least be made an option for the user. Watermarks can be used to mark pages that are not longer available or (different mark) have been updated. The same watermark facility can also be used to make users aware of *copyright issues*; this would require an HTML marker for copyright (or more general, legal) information.

There might be a legal issue with caching web pages. It is possible that if the cache is copy-protected or stored in a non-obvious place there is no violation of intellectual property rights. For example, it appears that currently when a file that is rendered by a separate plug-in, browsers such as Netscape write a temporary file in the /tmp directory. Temporary files usually have consecutive numbers and old files can easily be deleted automatically to keep the cache size more or less constant.

---

* »Unsere Zeit ist reich an erfinderischen Geistern, deren Erfindungen unser Leben beträchtlich erleichtern könnten. Wir überqueren vermittels maschineller Kraft die Meere und benutzen auch maschinelle Kraft, um die Menschheit von aller ermüdenden Muskelarbeit zu befreien. Wir haben fliegen gelernt und sind fähig, Mitteilungen und Neuigkeiten durch elektrische Wellen über die ganze Welt zu verbreiten. Die Produktion und Verteilung der Waren ist jedoch ganz und gar nicht organisiert, so daß jedermann in Furcht leben muß, aus dem ökonomischen Kreislauf ausgeschieden zu werden. Außerdem morden die Menschen, die in verschiedenen Ländern leben, einander in unregelmäßigen zeitlichen Abständen, so daß jeder, der über die Zukunft nachdenkt, in Furcht leben muß. Dies kommt von der Tatsache, daß Intelligenz und Charakter der Massen unvergleichlich niedriger sind als Intelligenz und Charakter der wenigen, die für die Gemeinschaft Wertvolles hervorbringen.«

However, an archiving facility that is explicitly visible to the user would be more valuable than a cache whose contents is hidden from the user. Hierarchical archiving solutions are readily available in the consumer market, although not yet widely used on the desktop. At least for legal reasons, hierarchical archiving of all data has been customary in the computer industry for decades.

## 4.7 $W^3$ site categories

The following observations are based on this author's very limited $W^3$ surfing experience. More experienced people might have a better feel for the current status and future trends.

There appear to be three broad categories of $W^3$ sites:

1. Small publishers and individual posters
2. General publishers
3. Database / catalog

The first category, small publishers consists of sites with a few HTML pages that are updated periodically. They either advertise items that change slowly in time (example: http://www.hpl.hp.com/imaging/) or are personal pages of individuals (example: http://www.hpl.hp.com/personal/Giordano_Beretta/).

The second category, general publishers, covers companies that have a full-time staff devoted to the publication on the $W^3$. Publishing is their main business, and currently the main source of income is advertising. They collect as much demographic data on their surfers as they can. Example: http://www.hotwired.com/.

The third category consists of product catalogs or databases. We include here all searchable sites, such as news repositories and directory services. An example is http://www.hp.com:80/gsyinternet/products/products.html.

In the remainder of this section we will briefly comment on the issues in each of these categories.

### *4.7.1 Small publishers and posters*

A small poster is usually an individual with a home page and a few items she or he wishes to share. Like for small publishers, there will be only a few HTML pages. Because these people work on their pages only occasionally, any tools have to be very simple, so that their user interface can easily be re-learned at each use. Ideally all $W^3$ tools combined should have no more than half a dozen different functions [12].

Because of the smaller learning effort associated with it, this user category is the primary target for plug-ins to existing authoring tools. They are usually already familiar with these tools, and being able to export data as HTML files is of great help. Probably vanity is of some importance, thus simple tools like FrontPage, PageMill, or HomePage that allow to "dress up" a page might be welcome tools as long as they are simple.

However, as soon as the magic number of 7±2 HTML pages is reached [12], problems start to creep in. Dangling links appear here and there and the organization of the site can become obscure. This is because the tools do not scale. Tools like SiteMill, which check the consistency of links inside the site are very helpful, but they do not solve the organizational problem of giving a scalable structure to the site.

What is needed are tools that instead of operating on single HTML files operate on sets of files and encourage the adherence to a well structured methodology. A first step can be a tool more like a presentation software package. However, this approach may be too limiting because the structure is linear. We will discuss this issue in Section 4.8 on page 32.

### *4.7.2 General publishers*

Professional publications require and can afford a high degree of complexity. It is required because a publication has to attract readers, has to have a proprietary look or design for brand identity, and must be novel ("professional" means it cannot easily be produced by amateurs). Complexity is possible because a skilled staff is available and expensive tools can be purchased and quickly amortized.

Such publishers will always push the envelope of technology. They will concoct new design ideas that are very laborious to implement. Then in the next iteration the tool implementors will support the ideas in their next release of the tools, and the publishers will concoct new, more elaborate design ideas.

Professional publications are often organized graphically in the form of concurrent threads. HTML frames, which are too difficult for casual users to deploy successfully, are a good tool to present threads in parallel.

More than with printed media (like magazines, compact disks, CD-ROMs or video cassettes), publishers will have to rely on advertisements to create a stream of revenue. Also, because information on the $W^3$ is easily copied and redistributed, personalization of information is very important.

Already today most professional publishers require user registration. With the acquired very high quality demographic data, publishers can offer precisely targeted advertisements, for which they can charge higher prices. For the publisher it is very important to be able to follow users while they navigate a site because the analysis of the trail can be used both to customize the information presented to the user and to refine the user profile.

Because of this extensive customisation, it will probably not be meaningful for a publisher to author HTML pages. For a publisher instead it is more compelling to design just templates for pages or frames. Raw contents would be stored in small databases. The client software would monitor the user's activities and in combination with the user profile create on the fly HTML pages by merging the customized information, the raw contents, and using the templates for the visual appearance.

Thus for professional $W^3$ publication, custom client software that is easily modifiable is far more important than web page design tools. This might be a third life (after research and financial applications) for Smalltalk and its derivatives like NextStep, ET++, and Interviews.

#### *4.7.3 Database / catalog*

For this application group no static HTML files are necessary. The current method of using a plug-in to a database program for creating HTML files with the database contents is too inefficient to have a chance to survive.

The best way is to use the profile in the customer database to guide the user towards the information sought in the database. HTML files are generated on the fly using a set of templates. The opportunities are in creating readily re-usable server engines that can be updated in a very short time when a new strategy has been identified for better guiding the consumer to the information the vendor or the consumer are targeting (see Section 4.10.1 on page 36).

### 4.8 Scalable methodology for structuring sites

As an example, we will discuss a methodology for small sites and posters. For professional publishers and database applications the same principles hold, but the raw classes or a framework would be supplied to the client software, instead of publishing tools with a GUI. As the discussion will show, the generalization from applications to frameworks comes naturally in the $W^3$ case.

In Section 4.7.1 on page 30 we saw that authoring should be focussed on sets of HTML pages instead of single HTML files, and in Section 4.4 on page 22 we saw that a tree is the best structure to organize information. We will assume, that each site has associated with it a data structure describing the site's link structure. For example, in SiteMill such a structure is built each time a site is loaded. We call this structure the *site graph*.

### 4.8.1 Faster surfing

This is a proposal that requires changes in both HTTP and browsers. When we have a problem we do not know what to ask, what information is needed; consequently the $W^3$ is much more useful if it can be surfed faster. In essence, this is what search engines do, but they give only punctual information, no context and no knowledge, because they ignore structure.

This proposal consists in attaching a *property node* to each node in a site graph. This node is mostly graphical and

1. summarizes the node (this may be a picture)
2. contains links to the sibling nodes and the child nodes (this may be computed on the fly)

A $W^3$ browser would then have four modes, one for navigation, one for browsing, one for viewing, and one for printing:

1. *Navigate:* the site tree is displayed, each node is an active link; this mode gives a large context.
2. *Surf:* the current property node is displayed; this mode gives an at-a-glance view of the current location.
3. *View:* the current node is displayed; this it what browsers do now.
4. *Print:* gives the user all the information necessary for printing; the user can print the subtree if there is one or the current node; for each node the size in printed pages and an estimated print time is given for the selected printer.

### 4.8.2 Templates for tree nodes

The HTML paradigm* for organizing the information in a page is to use frames to organize the page in viewports. In our opinion it is very difficult to design with the frame paradigm, because the design has to be such that there is never scrolling inside a frame (nested scrolling is very confusing) and navigation sequences are not obvious (where does “go forward” take me? how do I go back to the previous frame set?).

A template paradigm is much simpler to understand, although for a well designed page more data has to be transmitted. Ideally the components of a template are imported by reference, because people like to redesign their pages very frequently and it is easier to change the contents of a file called “background.gif” than opening each file and changing the URL for the background.

* Currently still in flux.

Assuming the site is structured as a tree, a good design methodology could consist in storing actual information only in the leaf nodes. All other nodes would just have active icons to siblings and childs.

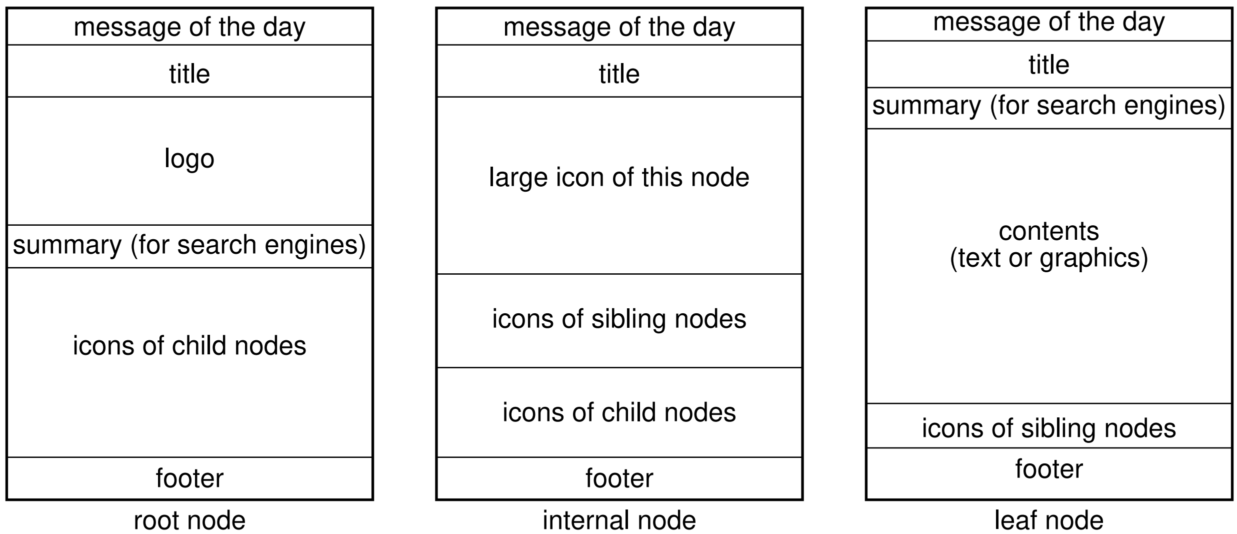



**Figure 6. Possible templates for $W^3$ pages. The title area would always also include links to the root node and to the parent node.**

It is immediately clear from Fig. 6 that all internal nodes can readily be generated by an authoring tool, sparing the author the tedious drag & drop and formatting tasks. The leaf nodes can be created with any tool, such as *FrameMaker, Word, Excel, File-Manager, PageMill, vi,* etc.

In reality a little more automation is possible. If a database package is used to keep track of the leaf nodes, only the contents part of a leaf node has to be created. Each database record would then include the information for the other parts (icon, label, etc.) as well as keywords that would be used to generate an ordered list of leaf nodes and the common parent node.

If an abstract is included for each leaf node, then the parent nodes of leaves could be formatted as tables with a row per leaf and two columns per row; one with an icon per file format in which the document is available (*e.g.*, PDF, Postscript, HTML, FrameMaker, etc.) and one column with the abstract itself.

## 4.9 $W^3$ applications

The common user model of $W^3$ is to view it as a tool to download stuff from the Internet. This model might come from the old custom of reading the NetNews and scouting for free software. The $W^3$, however, is about sharing information and therefore uploading and downloading must be viewed as symmetrical operations.

To be useful and long-lasting, uploading must be more than just posting vanity pages. Moving operations from the server to the client is not an answer. The problem is not how to implement Photoshop on the $W^3$; the problem is to find new paradigms and the best way to cast each of these paradigms.

For example, today many $W^3$ imaging applications apply a filter to an image defined by an URL. It would be more efficient to download the filter and apply it locally to the image, because the filter is encoded in fewer data.

To find a new paradigm for $W^3$ imaging, recall the old adage of quality *versus* quantity. Especially for the home user (or whoever pays for the equipment from her/his own pocket), the image quality is not yet there. So the opportunity for digital photography is not in quality but in quantity.

Considering that computers are good at number crunching and humans are good at categorization, an example of an application would be a photo categorizer based on HTML forms or on an applet, which would allow the user to "sort" the photographs into categories. A second tool could then be used to apply a filter or operation to the whole category.

Instead of then uploading the images to the server and applying the filters, the server could act as a broker, that in the sense of Bernardo Huberman's ecology of computing [16], subcontracts the operations to machines on the Internet which can do it at a low cost.

### 4.10 Limitations of other methods

The reason for the success of today's incarnation of $W^3$ are

1. the hypertext access paradigm encourages browsing
2. GUI[*] and WIMP[†] allow for simple but powerful user models
3. the flexible architecture allows for customized interactions

As mentioned earlier, $W^3$ has been around for a while and it was only the GUI that ignited the frenzy. Today $W^3$ is mostly hype, and when hype subsides, there will be a shakeout and only those who are able to deliver item 3 above will survive. The decline will happen when the $W^3$ transitions from the early adopters to the pragmatists—and Microsoft[‡] is working very hard to make it happen now.

---

* Graphical user interface

† Windows, icons, menus, pointing devices (or pull-down menus)

The key for success is to be able to provide knowledge (see Fig. 3 on page 13). Knowledge is highly personal, *i.e.*, it must be tailored to the recipient. The $W^3$ technology to achieve this is to generate HTML files and CGI interfaces on the fly. The common example is to generate the HTML dialect for the browser the surfer uses. Sites must be active.

It could be argued that active sites are not necessary, because all can be pre-determined, and surfers just click their way to the requested piece of information. This process, however, is what the value of information is about. The clicking sequence of the phone tree (press 1 if you are on a touch tone phone,…, press 1 for support,…) is not an acceptable *modus operandi*.

So, do active servers render Java or VisualBasic applets obsolete? No, because they are not related. Applets and forms should be used by the surfers to communicate to the server what they are seeking or what they have to offer.

As we saw earlier, if the users can identify themselves, the trail can be administered at the site by the server. Otherwise, administration must occur at the browser side. The browser solution has the advantage of privacy. However, only the site on the server knows when information changes and can be pro-active; this is surfer notification of important changes, *vs.* notification of checksum changes as it is done now.

Although for the reasons seen in Section 4.7.2 on page 31, publishers might want as much information as possible on the surfers visiting their site, in general there may still be a strong desire for privacy. A solution to this dilemma could be that at the first visit of a site the surfer requests a token. The site dispenses a unique identifier and the surfer's browser will always use this token at the given site. The surfers never have to reveal their identity. (The token can be given expiration dates both for the client and for the server.)

### *4.10.1 Example application*

Currently the $W^3$ Hewlett-Packard printer catalog is just an electronic version of the printed catalog. With the new methodology a customer first fills out a small profile, *e.g.*, customer identifier or user category (pupil, mom, accountant, engineer, etc.) and general parameters (all optional) like technology, price bracket, availability, etc. The customer is then presented with a web site built around the printers in which the customer is really interested. The format (technical data, examples, etc.) are tailored to the user profile.

---

‡ An interesting aspect of Microsoft's Internet strategy is that it is based on a sugar coating above proprietary technologies like ActiveX, DCOM, and VisualBasic (none of which scales well), while the current success of the Internet is centers on open standards and scalability. Nevertheless, Microsoft's might be a winning strategy because it ca lock in the consumers.

This method is more ecological and does not litter the Internet with unwanted information.

### 4.10.2 Search & directory engines

Usually, when we have a problem, we do not know what to ask, what information is needed. Therefore, search engines are only valuable as tools for building knowledge; they do not deliver knowledge. For example, under normal conditions, a search engine will not find Shakespeare's Hamlet phrase "to be or not to be" because it is composed entirely of stop words.

*W*$^3$ sites with information will have search engines. At sites where the information has been distilled into knowledge, search engines are obsolete.

### 4.10.3 Java applets

The idea behind Java applets is to ship behavior instead of data. It is a technological improvement and not a disruptive technology because it does not improve knowledge, it only improves its presentation (which for general publishers is nevertheless important because it allows the creation of attention-grabbing effects).

According to Jim Gosling [9], the main motivation for creating Java was to put behavior where the action is:

- put the user interface near the user
- do the computation near the data.

Consequently, Java is best suited to build applications for uploading information, not for publishing knowledge by synthesizing it on the fly.

An example for an application is *web photograph*y, where finishing services are offered over the Internet using a *W*$^3$ interface. The consumer can send via post mail exposed film cartridges or flash cards and the provider manages the pictures, or the consumer can use a *W*$^3$ browser interface to upload images or just URLs. Finishing services can be include image manipulation (application of imaging filters), photo album creation (database maintenance), and high quality printing & custom cropping. Possible applications for applets are:

- Image manipulation: the applet contains a crude public version of all filters. The consumer can try out the various filters on a low resolution version of the image. When the consumer has decided on the manipulations, the filter sequence is applied to the full resolution image on the server.
- Photo album: we saw on page 14 that it is hard to formulate the iconography of a picture. In a digital

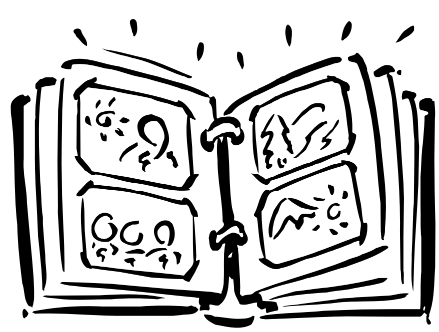


photo album, an applet could be deployed to create a WIMPy GUI to assist consumers to categorize, organize, keyword, or sort their pictures by moving icons or selecting key words from pop-up menus locally. The result will then be uploaded to the server where the database is maintained.

A word of caution should be said about the true platform independence of applets. Most operating systems have powerful collections of widgets and visual interface builders that allow the effortless creation of application front-ends. Many operating systems are platform-independent, so the generated application will run on many platforms. However, if the application relies upon an object library from that particular operating system, then the application will not run on a server on which the systems object library is not available. For example, if the applet is created with VisualWorks, it will not run on a machine unless Smalltalk is installed or an applet created with VisualBasic will not run on a Unix machine.

The fact that Java is a platform-independent language/abstract machine does not guarantee platform independence, because it allows to call system dependent libraries. Only applets that do not make such object calls are truly system independent.

## 5 Conclusions

*Tip the world over on its side and everything loose will land in Los Angeles (Frank Lloyd Wright)*

We are at the crest of the Internet frenzy wave and it is clear that to be a player in the market the wave has to be surfed now. It is less clear where there will be long term business opportunities when the frenzy is over and the market is driven by the pragmatists.

There is an important distinction between the European and Japanese business culture on one side and the American and Middle-Eastern culture on the other side. The first is relation-driven, while the second is transaction-driven. In relation-driven cultures a relation is established before a business is transacted; much time is spent on establishing the relation and cementing it legally to establish *bona fide*. In transaction-driven cultures the time is spent in finding the best deal, and once it has been found the transaction is very informal and quick.

Successful people in any walk of life often have a complementary approach. In relation-driven cultures, people who can find good deals and transact quickly are successful. In transaction-driven cultures the key to success is to have good relations (to be networked, "old boys network") that supply tips and insights which inject trust in transactions while limiting the search scope.

Here in America the pragmatist's tendency has been towards networking. In science, the Internet has been successful in allowing researchers to network. Pragmatists are now looking for the same effect: selling their ideas or wares in nanocommerce on the $W^3$, a neutral distribution channel open to everybody and not just to the "old boys." In short, what is desired is a shortcut to the American dream. Making the dream possible is a unique business opportunity.

The NetNews are a recent demonstration of what can happen when these aspirations are given free course; there is so much noise that it is impossible to make out any data. Publishing is very difficult. It requires hard work to structure ideas, it requires hard work to express ideas clearly, and there are few shortcuts. An informal poll suggests that senior research managers that have fancy web pages have assigned five to eight professionals to create them; good professional web publications employ a permanent staff of 12 to 30 people.

While the best short-term solutions are often created bottom-up by attacking a specific problem and using whatever tools are available, robust solutions that scale well require a clean design of the foundations and are best created top-down.

The $W^3$ is about publishing. We have seen that disruptive technologies are non-obvious and require value judgements. An example for a value judgement and ensuing new paradigm is to help the average American citizen to transition from poster to publisher. We could accomplish this task by helping people to structure their information; in conjunction with the Internet technology and the existing $W^3$ infrastructure this allows people to publish knowledge.

The opportunities are to create good structuring tools. As we have seen, this is predicated by a good set of axioms, which in our case is a methodology. The pages in a site should be organized in a tree, *i.e.*, there should be no cycles. There may be several trees on a site to cater for different types of users, each minimizing path length for efficient knowledge building.

For amateur publishers the tools will aid the structuring of static sites. Professional publishers will probably have sites that are computed on the fly, based on the user interaction. Publishers with a background in the entertainment business will probably have an edge over publishers from the printed media industry, because they have the skills to coordinate the work of a diverse team of creative workers.

Both advanced amateur and professional publishers will have high quality requirements on typographic style and image, when viewing online as well as when printing. Rendering data (fonts, bitmaps, etc.) and attributes (resolution, pixel depths, etc.) combined with the requirement of de-coupling printing from browsing suggest protocol extensions that go beyond the simple caching schemes used in today's browsers.

One of the most important creative processes in the compilation of knowledge is the categorization of information. We have proposed to extend the syntax for hypertext links to include a provision for a category identifier, such as a symbol. $W^3$ browsers could use the symbol to colorize active link text and make navigation easier.

# 6 Appendix I: Fifty years in the making

To provide some background, it is useful to show how the Internet frenzy is not a revolution, but inevitable outcome of 40 years of aging to maturity in a fertile environment. The intent is to show that we know how to swim and how to balance on a board; now we can just take the wave and surf it. One risk in reviews is to go back too far in time; to stay close to the immediate ingredients for the Internet frenzy, we only mention technologies that are directly part of the frenzy.

## 6.1 Memex and hypertext

At the end of WW II, in the July 1945 issue of The Atlantic Monthly magazine, Vannevar Bush reflected on the overwhelming flood of scientific information flowing into the learned communities [1].

> *Professionally our methods of transmitting and reviewing the results of research are generations old and by now are totally inadequate for their purpose. If the aggregate time spent in writing scholarly works and in reading them could be evaluated, the ratio between these amounts of time might well be startling. Those who conscientiously attempt to keep abreast of current thought, even in restricted fields, by close and continuous reading might well shy away from an examination calculated to show how much of the previous month's efforts could be produced on call.*

Bush realized that the problem was not the amount of information, but the problem was finding information in a pile of data. Today's trendy phrase is "drowning in data—starving for information."

> *The difficulty seems to be, not so much that we publish unduly in view of the extent and variety of present-day interests, but rather that publication has been extended far beyond our present ability to make real use of the record. 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 recognized that the limitation of decimal classification systems is that they are linear. It must be noted that Bush does not say information should not be linear, he says that the organization of information should not be linear. When information is consumed, there is a chronology of events, which can be represented as a trail in the data. Bush states that a trail should not be hard-wired in the data by means of a classification scheme because there is no perfect order that is valid for all contexts, because humans reason by association (we will revisit this statement later).

> *The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. It has other characteristics, of course; trails that are not frequently followed are prone to fade, items are not*

> *fully permanent, memory is transitory. Yet the speed of action, the intricacy of trails, the detail of mental pictures, is awe-inspiring beyond all else in nature.*

Bush proposed a mechanical device called the *memex*. In software, Bush's invention is nothing other than what we call today *hypertext*.

> *A memex is 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. It affords an immediate step, however, to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the memex. The process of tying two items together is the important thing. It is exactly as though the physical items had been gathered together to form a new book. It is more than this, for any item can be joined into numerous trails.*

Despite Bush's memex proposal, hypertext failed until 1993 to become a meme.* Apple's attempt in the early Eighties to create a hypertext system (called HyperCard) has been a commercial failure, but for all these years hypertext has been an active research area and the technology has been employed successfully for specialized applications, such as instructional software, help systems, online manuals, etc. The lesson from HyperCard is that technological progress depends on marketing skills—early hypertext consumer applications lacked hype (*sic*).

## 6.2 Internet

It started in Fall 1969 under the name Arpanet [15]. The idea was to have a high-speed packet-switching network connecting research super-computers that would survive attacks like nuclear wars and sabotage. It was designed to be decentralized and blast-proof, *i.e.*, no central authority is in charge and the network would still work if part of it goes off-line because there is no root node. Due to this decentralization the Arpanet has scaled relatively well, except for a few hiccups.

Packet-switching is important to allow building a system that is assumed at all time to be unreliable. Each packet is individually addressed and each node just forwards packets not addressed to the node itself. The route of packets is irrelevant.

Until recently the Internet was used mostly for scientific communication in the form of sending e-mail and transferring files. The culturally homogeneous user community operated under an implicit code of ethics. Today, when the user community is very heterogeneous this anarchic trait of the Internet where nobody is in charge can be distressing, especially for executives who find themselves unable to find interlocutors or to shape strategic alliances that stand on a firm soil. Socially, the Internet is a new frontier—everybody is a stranger and newcomer, nobody can be trusted.

* A meme is a cultural artifact (idea, concept, behavior, practice, belief) that spreads (replicates) in a society. A similar term used in the marketing community is that of a megatrend.

## 6.3 TCP/IP

The original Network Control Protocol (NCP) was replaced early on by a more sophisticated standard called TCP/IP, which is still today's Internet standard. TCP, stands for *Transmission Control Protocol* and converts messages into streams of packets at the source, then reassembles them back into messages at the destination. IP, or *Internet Protocol*, handles the addressing, seeing to it that packets are routed across multiple nodes and even across multiple networks with multiple standards.

TCP/IP is quite efficient and has always been in the public domain, so efforts to propose proprietary alternative protocols have never been successful and TCP/IP is now the global protocol suite.

Two related important Internet protocols are *Telnet*, which gives a login shell on a remote machine, and *File Transfer Protocol* (FTP) which is used to transfer files from one machine to another, possibly anonymously, *i.e.*, without needing an account on the remote machine.

## 6.4 WAIS and Gopher

Until 1991 using the Internet required to explicitly transmit information. There were no other search software other than *whois* and Telnet with *ls* and *grep* to find information whose location on the Internet was unknown.

Gopher, developed at the University of Minnesota and named for the tunneling activity of the rodent and school mascot, employs a menu-driven front-end to read documents and download files stored on hosts. It requires a client running on the user's machine.

WAIS, or *Wide Area Information Server*, was developed by Apple, Thinking-Machines, and Dow Jones. It is an information retrieval system that allows a client to perform keyword searches simultaneously on multiple on-line databases.

## 6.5 World-Wide Web

CERN is an international laboratory in Geneva where physicists conduct high energy experiments. These experiments typically result in a very large amount of data that is then mined over a long period of time and at many institutions to search for particles and effects. In 1989, Tim Berners-Lee came up with the idea of using hypertext links to connect sets of data and sequences of interpretations of the data. This allowed physicists at various universities that where mining the same sets of data to coordinate they work.

The World-Wide Web ($W^3$, *WWW*) is a set of three specifications:

- URL, *Uniform Resource Locator* to locate information
- HTML, *Hypertext Markup Language*, to write simple documents
- HTTP, *Hypertext Transfer Protocol*, to transfer HTML files

The original version of the $W^3$ was very powerful, allowing physicists to easily trace the analysis of an experiment and to instantaneously find the articles with interpretations. However, it was also a dry and relatively boring system that would be used strictly for research purposes.

### 6.6 Mosaic

Early 1993, when the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign announced a graphical user interface (GUI) for $W^3$ and people downloaded it, the effect was stunning. Instead of a shell script the Mosaic tool had an interface where the user would navigate the $W^3$ by pointing and clicking on hypertext links. And the early site administrators had put color images (albeit often boring pictures of research facilities) on their first pages.

All the sudden, the X-Windows terminals radically changed their look, because in the midst of all those windows running control-key bound *vi* editors, there where windows that had the look and feel of HyperCard—and the Macintosh effect repeated itself after 9 years.

# 7 Appendix II: Structure in mathematics

The rigorous notion that mathematics is about structures is only very recent. If a date has to be set to it would be 1935, when the members of the Bourbaki seminar started working on the multi-volume treatise "The Elements of Mathematics," which is published under the *nom de plume* "Nicolas Bourbaki." This was after the so-called crisis of foundations [7], which lasted for about thirty years, was solved.

Before the crisis broke out mathematics was organized along a number of disciplines, like number theory, calculus, geometry, etc. New theories on ordered sets, lattices, topological spaces, groups, rings, fields, vector spaces, partial differential equations, etc. have made it possible to reorganize mathematics from the point of view of structure. Bourbaki was the first to exploit this new view of mathematics and the Elements consist of a meticulous re-write of mathematics where every detail is proved and there is absolutely no hand-waving.

The key notions are those of models and isomorphism. In simple words, a model is the "interpretation" of one mathematical theory by means of another. Isomorphism is about one-to-one correspondences between constructs with relations or operations. The characteristic of the modern notion of structure is that every structure carries within itself a notion of isomorphism, and that it is not necessary to give a special definition of it for each type of structure. For example, once the real numbers are "interpreted" in terms of whole numbers, complex numbers and Euclidean geometry are also, thanks to analytical geometry.

Bourbaki's Elements start with the development of the fundamental structures, which are the ordered structures, the algebraic structures, and the topological structures. The theory of ordered structures is the same as the theory of ordered sets. The theory of algebraic structures deals with mathematical constructs in which compositions are given. General topology deals with the study of topological structures.

The fundamental structures can be combined in multiple structures, which lead to the classical mathematical disciplines. For example, infinitesimal calculus can be built upon the fact that of all three fundamental structures can be combined on the set of real numbers: an algebraic structure (the real numbers are a field), an order structure (the real numbers form an order), and a topological structure (there is a notion of limes).

The *general structure theory* is based on the notions of sets, relations, and maps. Consider a set $S$ with an n-tuple $\sigma$ of relations on $S$. The ordered pair $(S, \sigma)$ is a *relational construct*. This notion generalizes the fundamental structures, because the algebraic operations can be interpreted as relations and a topology can also be represented as a relational structure.

The classification of constructs is performed with systems of axioms. A *system of axioms* represents the properties of constructs. A construct with the properties represented by a system of axioms $\mathcal{A}$ is called a *model* of $\mathcal{A}$ or an *$\mathcal{A}$-construct*. We say that a model $(S, \sigma)$ of $\mathcal{A}$ has a *concrete structure* $\sigma$ of type $\mathcal{A}$, or an *$\mathcal{A}$-structure* (*e.g.*, a lattice structure or a group structure). For brevity, in the following by "construct" we will always mean a "relational construct."

Mathematics consists in forming new constructs and studying the correspondences between constructs. One practical method is to find as many models as possible for a given system of axioms. The method to establish correspondences between constructs and to induce structures on sets is to use a *map*.

To this purpose we need to introduce the notion of homology. The *operand distribution* of a construct is the number of operands for each element in a relation. For example, consider the set $S = \Re$ of real numbers with the n-tuple of relations $\sigma = (+, 0, N, \cdot, 1, R, \leq)$, in which case the operand* distribution is [3, 0, 2, 3, 0, 2, 2]. Constructs with the same operand distribution are *homologous*. Since the operand distribution is determined by a system of axioms, models of the same system of axioms are always homologous.

Consider two homologous constructs $\mathcal{G} = (S, \sigma)$ and $\mathcal{G}^* = (S^*, \sigma^*)$. A map $h: S \rightarrow S^*$ is a *homomorphism* if $h$ preserves all relations, *i.e.*, when

$$\forall R_k^{i_k} \in \sigma, \forall x_1,\ldots,x_{i_k} \in S \ : \quad (x_1,\ldots,x_{i_k}) \in R_k^{i_k} \Rightarrow (h(x_1),\ldots,h(x_{i_k})) \in R^{*i_k}_{k} \tag{1}$$

A homomorphism of a construct $\mathcal{G}$ in $\mathcal{G}$ itself is called an *endomorphism* of $\mathcal{G}$. The strongest map is a map that not only preserves all relations, but that also is a one-to-one correspondence. A map from $S$ in (on) $S^*$ is an *isomorphism* from $\mathcal{G} = (S, \sigma)$ in (on) $\mathcal{G}^* = (S^*, \sigma^*)$ iff the map is injective (bijective) and the map and its inverse are homomorphisms.

Isomorphism is an equivalence relation on any set of constructs. We now consider a sufficiently large set of constructs. An *abstract structure* is an equivalence class of isomorphic constructs. An alternate definition is that if $\Sigma$ is the class belonging to $\mathcal{G}$, then $\mathcal{G}$ has the abstract structure $\Sigma$. According to this definition isomorphic constructs have the same abstract structure. The power of these notions is that one can choose the most suitable representative of a class and then generalize to the entire class—this is why we noted earlier that we want to find aa many models as possible..

The isomorphism relation is compatible with the property of being an $\mathcal{A}$-construct. This means that if $\mathcal{G}$ is a model of $\mathcal{A}$ and $\mathcal{G}^*$ is isomorphic to $\mathcal{G}$, then $\mathcal{G}^*$ is a model of $\mathcal{A}$. For a type of structure given by a system of axioms $\mathcal{A}$ there is a set of concrete structures and also a well-determined set of abstract structures.

* The relations are: first operation (addition), its neutral element (zero), its inverse (negation), second operation (multiplication), its neutral (one), its inverse (reciprocal), order (less or equal than).

## 8 References


[1] Phil Agre, personal communication, May 1996.

[2] Paul A. Allaire, "Chairman's Message," in Xerox Corporation 1995 Annual Report, Stamford, March 1996.

[3] Richard Beach and Maureen Stone, "Graphical Style: Towards High Quality Illustrations," in *Computer Graphics*, 17, 3, 127–135, July 1983.

[4] Giordano Beretta *et al.*, "XS-1: An integrated interactive system and its kernel," in *Proceedings 6th International Conference on Software Engineering*, 340–349, September 13–16, 1982, Tokyo.

[5] Giordano Beretta, *Scanner Considerations for Color Facsimile*, HP Laboratories Technical Report HPL-96-39, March 1996.

[6] Christine L. Borgman, personal communication, May 1996.

[7] Nicolas Bourbaki, *Elements of the History of Mathematics*, Springer-Verlag, Berlin, 1994.

[8] Vannevar Bush, "As We May Think," in *The Atlantic Monthly*, July 1945.

[9] Jim Gosling, personal communication, April 1996.

[10] Martin Haeberli, personal communication, June 1996.

[11] Ho John Lee, personal communication, May 1996.

[12] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," in *The Psychological Review*, **63**, 2, 81–97, March 1956.

[13] Jay Nievergelt and Jean Weydert, "Sites, Modes, and Trails: Telling the User of an Interactive System Where He Is, What he Can Do, and How to Get to Places," in *Methodology of Interaction*, edited by Richard A. Guedj et al., 327–338, Elsevier North-Holland, Amsterdam, 1980.

[14] Edward M. Reingold et al., *Combinatorial Algorithms*, Prentice-Hall, Englewood Cliffs, 1977.

[15] Bruce Sterling, "Short History of the Internet," in *The Magazine Of Fantasy And Science Fiction*, Cornwall CT, February 1993.

[16] Carl A. Waldspurger et al., , "Spawn: A distributed Computational Economy," in *IEEE Transcations on Software Engineering*, **18**, 2, 103–117, February 1992.


WWW

## 9 Index

## I



## J



## K



## L



## M



## N



## O



## P