

3 CAN WWW BE SUCCESSFUL?

Hermann Maurer

IICM, Graz University of Technology, Graz, Austria

hmaurer@iicm.edu

Abstract: Considering the fact that the number of WWW servers and users keeps exploding the title of this paper sounds strange, to say the least. However, while every organisation without their own proper URL starts to feel like an outcast, there are clouds over the future of the WWW.

After all, the number of “broken links” keeps rising every month, not just in absolute numbers but also percentage-wise. Estimates have it that by the year 2000 almost 10 % of all links will be broken, i.e. will (when clicked at) produce the infamous message “Error 404. Object not found”. The situation is not restricted to the Internet, but applies as well to WWW based Intranet solutions. A well-known Fortune 100 company, when systematically checking their Intranet, found 4000 links that were not working. The overall situation is worse: search engines are unable to cope with the chaos on the web, returning too many hits, documents found are obsolete, secure payment procedures are still a rarity, and so is making money with what one offers on the WWW, etc.

Thus, it is no wonder that insiders are starting to ask the question: will WWW be able to evolve into a truly usable environment, or will it be necessary to “start all over again”, with an entirely new “Web2”, or whatever you might call it. After all, such things have happened before: the amateur film has not evolved to but has been replaced by video; musical records of once have been replaced by CD’s; and Videotex, even the fairly wide-spread Minitel of France, was wiped out by WWW. Are we going to witness another revolution? Or is there a reasonable migration-path to what we surely need: a Web2?

Some of the main weaknesses of ordinary WWW will be discussed in this paper. It will also be argued that having WWW interfaces on top of traditional databases does not solve the problem. Rather, a more integrated approach of four competing paradigms is necessary to solve the problems we encounter today. Solutions of this type are starting to be used in a growing number of large Intranets, making it possible that they will “spill over” into the Internet. For the majority of users unnoticeable, this may well mean the replacement of the WWW as we now see it by more modern solutions, by a Web2.

3.1 INTRODUCTION

The World Wide Web (WWW) keeps exploding. The majority of persons concerned with the WWW believe, without much reflection, that the expansion witnessed will continue, and that the WWW will be the infrastructure of the often quoted information society of the future. There is little doubt that the information society will be based on a sophisticated network of service providers that can be reached (eventually) by most people at any point and at any time. However, it is our contention that it is not at all clear whether such a network of the future will evolve from WWW, or whether it will derive from new basic concepts. In this paper we will argue that such a “Web2” will have to have much functionality that is difficult to fit on top of current WWW architecture. Thus, the question arises whether there is an evolutionary path from current WWW to a Web2, or whether a Web2 will emerge that will make WWW as obsolete as Videotex was made by WWW, or as 8 mm (amateur) movies were made obsolete by videocameras. We will not answer this question in this paper, but we will show that the question is justified, and that the challenge to an evolutionary path from WWW to a Web2 is indeed formidable.

The paper is structured as follows. In the next section we will present some arguments why a Web2 is needed. In the following section, the main part of this paper, we will point out various serious shortcomings of the original WWW concept and add some comments on how they are currently being addressed. We will then briefly mention that some, but by far not all, shortcomings of the WWW concept have been taken care of by some developments the author of this paper was involved in. In the final section we return to the main point raised in this paper: can WWW evolve to a Web2, or do we need a new approach altogether?

3.2 WHY WWW WON'T KEEP EXPANDING IN ITS PRESENT FORM

It has been often said that the WWW is the “largest information repository the world has ever had - but also the most chaotic one”. Both points are correct, yet the significance of the second one is often not taken serious enough. The WWW is not just chaotic, it is getting more and more so: the percentage of broken links has risen from 4 % in 1996 to over 6 % in 1998 and - as the hypertext pioneer Nielsen is being quoted - is likely to hit 10 % by 2000. The rise in working WWW servers that contain obsolete data is equally alarming: a recent sample shows that less than 20 % of all servers contain reliable and up-to-date information. Relevant information is ever harder to find: the enthusiasm with which search engines like Alta Vista have been welcomed is being replaced by the feeling that without human intervention too much useless information is found; the resulting new search engines that work with substantial teams of specialists to weed out and categorize information are trying to cure symptoms, but not the underlying disease; and this is true, independent of whether such human-assisted directories are still called search engines, or “meta-servers”, or “Web portals”. At the same time as “relevant” information is every harder to

locate, more and more robots “harvesting” servers keep increasing the load on servers intolerably; also, large WWW sites are exponentially hard to administer. A plethora of tools developed to assist do indeed help to some extent, yet basically just cover up flaws in a system that was designed for certain purposes but is now used for entirely different ones: It is important to understand that all that is going to be mentioned in terms of shortcomings of the WWW is not a criticism of the original WWW design: the team of Berners-Lee, Cailliau et al [1] at CERN developed WWW for a specific purpose: to make a growing archive of text-only physics reports available to the world. Little could they know that WWW would be (mis)used as a network with huge amounts of dynamically changing multimedia data, used in an increasingly interactive fashion. For their purposes, the original concepts of WWW were sufficient, and ingeniously simple. It is this very simplicity that created the WWW explosion, yet it is the same simplicity that is now turning into a handicap. Programming became popular because of simple-to-learn and simple-to-use programming languages like FORTRAN or BASIC. Yet nobody in his right mind would design large application programs using such languages today. It is the contention of this paper that a similar paradigm shift must and will happen with respect to WWW. FORTRAN is still alive as FORTRAN version X (whatever X is) and is useful for special-purpose applications; WWW version X may still be alive in 10 years and be used for some niche applications. But maybe the world of information will not be dominated by such better WWW but by entirely different Web2 systems.

Summarizing, the WWW has serious deficiencies for today’s and tomorrow’s applications, that prohibit both users and administrators to get the best out of a system of distributed services we have come to expect the WWW to be. We will analyze some of the major weaknesses of WWW in the next section. The future will show whether WWW can evolve to embrace all such features or whether new systems might supersede WWW at some stage.

3.3 SHORTCOMINGS OF WWW THAT WILL EVENTUALLY STIFFLE FURTHER GROWTH

In this section we discuss some of the problems inherent in WWW architecture. This is by no means a complete list, but it will prove the point this paper wants to make: some dramatic changes at the very basis of WWW will be necessary if WWW is to evolve to a useful Web2.

3.3.1 The communication protocol

The standard HTTP protocol used in WWW is a stateless, non-connection oriented protocol and offers no way to provide a “quality guarantee”, i.e. a guaranteed bandwidth as required for many streaming multimedia applications, particular audio and video. The fact that the protocol is stateless makes it necessary to use crutches such as “cookies” and “session keys” to preserve information on what users have done before. Connection oriented protocols are

much more efficient when substantial sessions are carried out with one server. And the fact that security issues are not addressed in HTTP has required the addition of e.g. SSL (secure socket layer) constructs, or new protocols such as HTTPS.

The weaknesses of HTTP are well-known to the community. And it has to be said there - and will not be repeated in all sections (although it applies to most) that the W3C (the WWW Consortium headed by Berners-Lee) [18] is trying very hard to push improvements. Unfortunately, the W3C is hampered by two major facts: (i) the main industry players often act independently of the W3C and (ii) the issue of backward compatibility is always looming. And this is exactly the hot issue addressed in this paper: is it possible to carry through the innovations necessary under such circumstances?

3.3.2 *The document model*

The basic document in a WWW server is an HTML-page. HTML is a simple logical mark-up language derived from SGML that allows users to specify titles, subtitles, the start of a new paragraph, etc, but leaves the actual representation on the computer screen to the viewer that interprets the HTML code. Under pressure from groups that wanted more control “on how things look on the screen” the simplicity and device independency of HTML was soon destroyed by introducing presentation mark-ups (allowing to define font size, colors, in-line images, etc.) and hence creating device (e.g. screen-resolution) dependence. Further, HTML today does not really exist anymore: there is one version by Microsoft, another by Netscape and despite the commendable efforts of the W3C mentioned earlier no real new HTML standard that is accepted by all major players is in sight. The problems with HTML are accerbated by the fact that all “meta-information” including link information is embedded in the document. We will return to the problem of why embedding information about information x into information x is undesirable, but we want to mention the general problem of meta-information at this point: storing a document without information about it (when it was created, who created it, what is the topic of the document, who owns it, etc.) is almost like putting photos in a photo-album without noting down when and where the photo was taken, what is seen on the photo, etc. In ordinary HTML a “meta-tag” does exist, but with no further structure behind it the use of it is very limited [6]. Yet meta-information is crucial for finding relevant documents, for defining the semantic structure of a document, etc.

As a result of the deficiencies mentioned, a powerful derivative of SGML, the so-called XML standard was introduced in 1996. It is much more powerful than HTML, allows logical and semantical mark-ups, allows presentation specific instructions using style sheets, allows the incorporation of meta-information and a somewhat more flexible link management than HTML [18].

It is to be seen whether XML will catch on on a large scale. Some say it is “too little, too late”. The reason for this is that many organisations, dissatisfied with HTML, have started to use their WWW servers as repository for

PostScript, PDF, Winword, etc. files. I.e. the uniform presentation protocol of the original WWW has already made place to a mixture of many formats. And while this is evident in case of text-oriented material the situation concerning video and animation formats is still more chaotic. The idea to use JAVA applets (running on a “virtual machine” in “any environment”) has made the situation concerning media-rich applications a bit simpler, yet does not present a solution on its own.

Worse, however, is the fact that important other issues have not even been seriously considered, yet: such issues include the definition of various types of access rights, provisions for versioning, for charging, for transclusions in the sense of Ted Nelson [3], for distributed editing, for document locking during transactions, etc.

3.3.3 *The access model*

WWW is based on the idea that information is accessed using links. We will discuss in section 3.4 why the link concept and related issues as available in today’s WWW systems are not suitable for a Web2, but we want to make another important claim in this section.

The claim, in a nutshell is that information systems should not be built around a single data access paradigm, but should seamlessly combine the four major paradigms known in computer science. Those four paradigms are:

- (i) Access by searching: In titles, keywords, fulltext with all kinds of extensions and combinations, as is e.g. done in search programs such as Verity and Fulcrum. Note that searches (including full-text searches) should also be supported in non-HTML files (e.g. in PDF and Winword files), i.e. any good search program needs “filters” for a large set of file formats.
- (ii) Access by structure: Much of our thinking is in terms of categories, classifications, menus, directories (or whatever you want to call it). It is thus natural that the document space of a WWW server should allow (arbitrary many, possibly overlapping, possibly user-defined) hierarchical or DAG-like views. The advantages of such approach not only for locating information but also for link reduction, re-usability of modules and easing server administration have been discussed in detail before, e.g. in [4] and [5].
- (iii) Access by attributes: This type of access, also called access by meta-data, is closely related to relational databases. Documents or document groups can have a set of attributes and those can be used in SQL-type (or simple versions thereof) queries to locate data.
- (iv) Access by links: The classical WWW approach with all its advantages (“intuitive”, “associative”) and all its disadvantages (“the Web is a huge Spaghetti-bowl of links by now” according to R. Cailliau, one of the inventors of WWW; “the getting lost in hyperspace syndrome”, etc.)

It is our contention that neither of the above approaches on its own is sufficient. Hence pure WWW systems relying on links only (iv) will not work well in some situations, nor will systems that are purely based on attributes (iii) like relational databases; nor are systems based only on search engines (i) or menus (ii) suitable for all situations. It should also be clear that a “serial combination” like a WWW (link-based) interface to a relational database will be sub-optimal in some cases, nor will any fixed combination of the four paradigms (i) - (iv) provide an ideal solution for all applications.

Rather, a seamless applications dependent mix of the four access strategies is what is needed, and with the exception of Hyperwave (see [4], [6], and [7]) no current WWW technology provides this feature that will be essential for any Web2.

3.3.4 *The link model*

Links in ordinary WWW servers are compared with goto-statements already in [8]. This leads to the obvious consequences that links should be replaced by structure as much as possible. The remaining links should be at least maintained automatically by the system, and should have a number of other desirable properties. In ordinary WWW links are uni-directional, have no attributes, and their anchors are embedded in the HTML page. It has been argued at length, particularly in [9], but also in [4] and [5] that links (as implemented already in Brown University’s Intermedia system almost 10 years ago!) must be bi-directional, be treated as objects with their own attributes (so that e.g. some links are only visible to some people, or only for a certain time) and their anchors must not be embedded within the documents (so that e.g. adding a link can be done without write-access to the document where the link is added).

Indeed there are more general issues at stake: links (and other “tags” that are commonly seen as part of the document) should not only be separate from the document, but one document may well have different sets of such tags associated with it. Depending who and how one reaches a document completely different sets of links, but also different items in different forms might be displayed.

There is also another issue that has to be addressed rather sooner than later: at the moment, anyone can link to other pages. However, owners of pages may resent that links point to their pages (e.g. since they want users to first go through some title page with advertisements). Thus, future WWW systems have to provide means for blocking links to certain areas of the database!

3.3.5 *Making WWW more interactive*

Ordinary WWW is basically a “static”, “read-only” affair. Mind you, using forms or JAVA certain kinds of interactions are possible. In general, however, WWW allows to retrieve some information but does not allow to work with it. Work with it means that users should be allowed to create notes and links on

any document for themselves or a specific group of users, thus providing the possibility for individualisation, customization and cooperation.

Private and shared workspaces to re-arrange and transclude information should be provided, news-group like discussion facilities integrated, and bookmarks should be kept on servers rather than on clients to allow bookmark-sharing as a powerful tool of communication and cooperation.

Current ordinary WWW technology is a far cry away from such facilities, although serious attempts to integrate them can e.g. be seen in [10] and [11].

It is our contention that a Web2 will have to be orders of magnitude more interactive than current WWW. And simple “push-technology” now available on some WWW systems is not sufficient, although some variantes of it like the “query object” in Hyperwave [12] are first steps in the right direction.

3.3.6 *Other issues*

The above subsections have listed a few of the important issues unresolved in ordinary WWW, yet essential for a Web2. The list can be expanded arbitrarily. We just want to show the complexity of the issues by mentioning a more or less random selection of three other points.

- WWW does not allow distributed editing: it is the webmaster who controls what is inserted where, how it is accessed, what links go into a document and emanate from it. However, much more flexibility is needed: not only must it be possible to appoint subadministrators who can work in certain parts of the WWW database, but this process should be recursive, providing a full hierarchy of administrators, i.e. fully distributed editing. Moreover, adding links and views for personal or group use must be possible even without document editing privileges.
- Caching information in servers is crucial for performance and for decreasing network load. At the moment, however, caching poses serious problems: when accessing a document in a cache how can it be assured that no obsolete copy is shown; and if a document is cached from a server (even if no pricing is involved) caching distorts the hitrate of pages, the “holy grail” of WWW advertising. Consequently, many pages carry the information “don’t cache me”. First, such information might be ignored by some servers; second if it is not ignored the benefits of caching disappear. What is clearly needed is a more sophisticated caching protocol: the server A knows when a document X of A is cached in server B. If X is changed in A, B is notified of the change, and accessing X on B results in fetching a new copy from A; further, if X is accessed on B, a message to that extent is sent to A so that the hit-rate of X on A remains correct despite the caching; a similar mechanism can be used even if charges are attached to X.
- The whole area of subscriptions, payments, micro-payments, charging, etc. is still much in flux, yet is clearly as essential for e-commerce as

truly secure transactions. For example, billing mechanisms on servers that are time-dependent (news get cheaper as they get older!) and a host of similar situations can be handled today, if at all, only by “tons of” CGI scripts, see [6], not “exactly” what is desirable!

3.4 WHAT CAN BE DONE ABOUT THE SHORTCOMINGS OF ORDINARY WWW?

Many but by far not all of the problem areas mentioned have been tackled in the meantime by a sheer endless list of add-ons, like CGI scripts, JAVA applets, JAVA scripts, Active-X components, etc. Some of the more fancy WWW applications resemble a shaky platform (ordinary WWW) onto which - using a less than perfect operating system (Windows) and a less than fully reliable “platform independent” programming language (JAVA) - fancy buildings of amazing complexity have been built. This cannot continue.

More solid approaches use standard databases (such as Oracle) as platform and provide the feeling of WWW through a WWW interface. This works perfectly for some well-structured applications but tends to fail in complex Web Based Information Systems [7], when Web Based Knowledge Management [6] is attempted, or even when just heterogeneous information with a dynamically changing structure is to be dealt with.

A more integrative approach is taken by Hyperwave [12] and hence warrants evaluation when considering the implementation of complex WWW-based applications. Looking at the issues raised Hyperwave is not much help concerning protocols (its own connection-oriented protocol cannot be used with standard clients, hence Hyperwave is also forced to work with HTTP); it does support a much more general document and access model, solves most of the problems of link management, goes a long way towards making WWW more interactive by allowing private and group views, annotations and links but scratches only the surface as far as distributed editing, caching or billing is concerned.

3.5 WILL WWW SURVIVE?

The shortcomings of current WWW as outlined make it clear that a Web2 will be necessary to handle the kind of interactive, communicative, information- and transaction-oriented system that is desperately needed.

Attempts by the W3C, and systems such as Hyperwave are trying to build an evolutionary path from current WWW to a Web2. However, the Web2 will differ from current WWW to an extent that it is not clear whether the evolutionary path that always has to take into account backward compatibility will not exact too high a price, and that a new Web2 that is not compatible with current WWW might not be the better solution.

One thing is clear: if anyone is building a complex application based on WWW with “tons of” customized CGI scripts those solutions will not carry over easily to a Web2. Hence one recommendation is fairly obvious: for small applications one can use simple WWW servers; for large applications one should

use a standard database with WWW interface or an integrated solution such as Hyperwave. Solutions that are taylor-made by adding many customized CGI scripts to simple servers will be hard to maintain, and the roll-over to a Web2 will be difficult, to say the least.

The important fact to notice is: WWW will not survive as is, it will be replaced by a Web2, sooner rather than later. However, this will not be very noticeable for users. Rather, the brunt of the "conversion" will be on the shoulder of those running WWW services today and it is up to them to use tools that will make the migration to a Web2 easy.

References

- [1] Berners-Lee, T., Cailliau, R., Groff, J.-F., Pollerman, B. (1992). World-Wide Web – The Information Universe, *Electronic Networking* 2,1 , 52-58.
- [2] *W3C - The World Wide Web Consortium*, see <http://www.w3.org>
- [3] Nelson, T.H. (1987). *Literary machines*. Edition 87.1, 702 South Michingan, South Bend, IN 46618, USA
- [4] Maurer, H. (1997). What We Want from WWW as Distributed Multimedia System, *Proceedings of the VSMM'97*, Geneva, IEEE, 148-155.
- [5] Maurer, H. (1998). Large WWW-Systems: New Phenomena, Problems and Solutions, *Proceedings of the 21st Annual Conference of the GfKI, 1997, Classification, Data Analysis, and Datahighways*. Balderjahn, Mathar, Schader (eds.), Springer, 270-276.
- [6] Maurer, H. (1998). Web-Based Knowledge Management; Internet Watch, *Computer* (March 98), IEEE, 122-123.
- [7] Maurer, H. (1998). Modern WIS, *Communications of the ACM* 41,7, 114-115.
- [8] Van Dam, A. (1998). Hypertext'87, Keynote address; *Communications of the ACM* 31, 7, 887-895.
- [9] Maurer, H. (ed.) (1996). *HyperWave: The Next Generation Web Solution*. Addison-Wesley Longman, London.
- [10] Dietinger, Th., Maurer, H. (1997). GENTLE – (General Networked Training and Learning Environment), *Proceedings of ED-MEDIA & ED-TELECOM 98*, Freiburg, Germany, AACE, 274-280.
- [11] Maurer, H. (1998). Using the WWW System Hyperwave as the Basis of a General Networked Teaching and Learning Environment, *CIT*, vol. 6, 1 (special issue).
- [12] *Hyperwave*, see <http://www.hyperwave.com>