Internet literacy as a networking enabler

Martin Vermeer

April 10, 2003

1 The need for digital literacy

The Internet, and computers in general, are a relatively new addition to the tool set available to people networking together to achieve common ends. The relative newness means that

1. most people practicing the art have had little or no formal training in using the new tool effectively, and
2. society has not yet settled on a "conventional wisdom" on how to to offer such formal training at an early age as an integral part of general literacy.

As a result, the vast majority of Internet and computer users are "amateurs", in the sense of being naive about the various technologies involved. While a desire exists to use the new tools effectively, most users are in a phase of their working lives where they have no excess personal resources to invest in this. People will only learn just enough to get by, and often not even that. The commercial software market is finely attuned to this constraint.

Things were quite different within the community in which the Internet, and computer use for networking, first took form; in this, the UNIX community, true amateurism existed and exists, a drive to excellence and becoming productive using and combining the best tools for the job, acquiring any skills needed. This is the UNIX culture with its RTFM ("read the f*ing manual"), dubiously humorous abbreviations, and compulsive textuality ("on the Internet, nobody knows you're a dog"). UNIX is still the system of choice when excellence rather than comfortable mediocrity is the goal.

Culturally, the Internet is still a UNIX network.

Normally, for individual computer users, it wouldn't matter too much if they are functionally computer illiterate; it's their problem, they are suffering the productivity loss, the hassle, the heartache. For companies, and for the national economy as a whole, it is of course a bad thing. But what really makes digital literacy a critical issue, is the Internet.

In these Internet days, people are responsible for what their computer does to others. The Internet is a traffic network; just like the road network, it is reasonable to have to pass a driver's examination before being allowed on it. Currently we have an "information super-highway" driven on by people without licences – often even proud of it –, and the vehicle manufacturers advertising it as a feature, no less!

Driving the Internet without a licence can make your computer do the following things:

1. Send out viruses to the fine people listed in your address book. Those viruses may lift all kinds of confidential files from your hard disk for the world to read
2. Send out Internet worms to thousands of other Internet hosts, trying to break into other systems
3. Allow spammers (distributors of unsolicited "junk mail") to relay their wares over your system, making it appear to come from you
4. Attack, concertedly together with thousands of similarly compromised systems, the White House web server or any other suitable target, by saturating it with a stream of suitably crafted IP packets (DDoS, Distributed Denial of Service, attack).

Trust me, you don't want to be part of that. Make sure you have your drivers' licence. And now: enjoy the drive!

2 Uses and abuses of email

2.1 Email in a networking setting

Email is a very widely used and useful tool for electronic communication, without which many people couldn't live anymore. The use of email as a preferred means of communication among members of a network is very convenient. A strong point of email is its asynchronicity; you don't have to answer immediately but can do so at your convenience. This facilitates networking, e.g., across many time zones. It bridges time as well as space.

There are a number of tasks for which email is *not* the most appropriate choice, however. If what you are doing involves anything else but sending around small, quickly composed textual messages, you should ask if email is really the best. If you find yourself collaboratively authoring – be it natural language text or code –, it is not. If you find you are "publishing", it is probably not. Acquaint yourself with better tools to do these tasks. They exist.

2.2 The MIME standard

Email is based on a set of open standards developed by the IETF (Internet Engineering Task Force), which allow for very general usage patterns. One of these open-ended facilities is MIME, or Multipurpose Internet Mail Extensions. This is, summarizingly, the standard that makes it possible to add *attachments* to email messages.

Sometimes nowadays the abbreviation MIME is read as "Microsoft Internet Mail Extensions", as it is typically used for attaching MS Word or MS Excel files to a message. Unfortunately, because, while the MIME protocol is open, these document formats are not.

Under MIME, an attachment is given a content type, which may be, e.g., text/plain, text/html, application/pdf, application/msword, image/jpeg, or whatever. In a properly configured mail program, this content type (or *MIME type*) is used to infer the application that should be used to open it.

E.g., for a content type of image/jpeg or image/gif, typically an image viewer is started up to view the image in the attachment. If it is application/pdf, the Acrobat Reader software is started to view the document, application/msword again starts MS Word, etc. This is a very practical arrangement that allows a user to immediately view attachments with the appropriate software without manual intervention.

There is a certain similarity with the use of file name extensions for files on your hard disk: if a file is called mydocument.doc, clicking on its icon in your file manager will automatically start up MS Word, if the file is mysheet.xls, the application started will be Excel.

But there the similarity stops. According to the MIME standard, for every attachment you send, the *correct* MIME type must be attached to it. And this is done very, very often wrong.

Often, MS Office attachments are transmitted as application/octetstream. What this conveys is, "*I don't really know what this is. A long, long stream of bytes. Figure out for yourself which application is needed to open this*."

Not very helpful. Luckily, many mail programs on Windows are able to look not only at the content-type, but also at the file name extension of the attached file, and draw the right conclusion from that. *But that is not according to the standard.*

So:

1. Don't ever use the MIME type application/octetstream. It conveys no information and a standard-compliant mail program will *not* know what to do with it. Manual intervention will be necessary.
2. Use the fillowing MIME types for MS Office documents:

   application/msword
   application/msexcel
   application/mspowerpoint

etc. This way, the document will be opened in an appropriate application.

3. Do *not* use the MIME type application/msword for RTF (Rich Text Format) documents. Use application/rtf.

   If you have MS Office, or only MS Word, installed, it will make no difference; this application knows internally how to load documents in the RTF format. The same applies to other office suites like Sun's StarOffice.

   For recipients who don't have such software installed, it makes a world of difference. There exist applications that are able to open RTF documents, but cannot handle the native, binary MS Word format. Complying with the standard will make these recipients happy at no cost to the others.

These simple rules can be easily complied with by properly configuring your program used for outgoing mail. Do this, or have your administrator do it.

2.3 Use of mailing lists

Mailing lists are an extremely convenient medium of communication for networked groups of people. However, it is also a medium that is easily abused or used inpractically, inconveniencing and annoying the users.

There are a number of simple practical rules to adhere to when using a mailing list:

1. Keep your messages small. This means, don't ever use large attachments, be they Office documents or imagery. Small attachments are acceptable at a pinch.

   The proper way to distribute large docs, if you have to, is posting them on a web site and mailing around links to it.

2. *Don't ever use HTML* for formatting your messages. Use plain text only. Also don't use multipart/alternative containing both plain text and HTML versions of your message. Doing so you are exhausting disk space, bandwidth and the patience of your audience.

   > Plaintext – don't settle for more!

An easy alternative to setting up a formal mailing list using mailing list software, is to use the "alias" or "group" feature present in many mailing programs' address books. This allows you to define a symbolic name for a group of email addresses of recipient; at send time, the name will be expanded to the list of recipients.

When you do this and the list is large, consider *putting the alias in the Bcc:* (Blind Carbon Copy) *header.* The advantages of this are

1. Every recipient does not have to read or scroll through an endless list of other recipients before finding the actual message;
2. the addresses will not end up in numerous Outlook address books for the benefit of mail viruses.

2.4 Other group communication tools

Actually, mailing lists are not the only and not even necessarily the best protocol for group communication. An older protocol, newsgroups (or NNTP, Network News Transport Protocol) exists which is in fact better suited, as it does not store the messages on a local hard disk. At the Helsinki University of Technology, we use good old NNTP a lot.

A fascinating application of NNTP for building a rich group collaboration environment is described by Byte Magazine: `http://www.byte.com/art/9709/sec7/art1.htm`.

A more recent alternative is web-based discussion groups, where messages are entered and read through a web browser.

It is possible to have the advantages of all three alternatives: software exists for converting mailing list content to newsgroup content (`http://www.gmane.org`), or to archive it on the World Wide Web.

There also exists an alternative called WiKi. For an example, look here: `http://emergent.brynmawr.edu/wiki/index.cgi/FrontPage`. Setting up your own WiKi on an intranet is possible; there are several implementations to choose from, e.g., `http://twiki.org/`. WiKi offers an environment where people can collaboratively author "webs" of Web pages on any subject of their interest. One success story is the "wikipedia", a free content on-line encyclopedia (`http://www.wikipedia.org/`).

2.5 Public groupware hygiene

Whenever you decide to archive a mailing list you are responsible for to a newsgroup or to the Web, *don't do so without knowledge and consent of your subscribers*. It makes their correspondence public. Also, when setting up the archival software (e.g., `http://www.mhonarc.org/`), take care that *email address harvesting software* as used by spammers (senders of unsolicited commercial email) cannot get hold of your subscribers' addresses.

3 Document standards and their uses

3.1 Plaintext

The file containing the string "Hello World" saved as a plain text file will occupy 12 bytes on your hard disk: ten letters, a blank and a newline.

Now, write an MS Word document containing the string "Hello World". The size becomes over 7 kilobytes. A growth of over 50 000 percent!

This is an extreme case, but shows the disadvantage of using MS Word or any other word processor format instead of plain text. Surely you, like me, have received MS Word documents that contained nothing but text, simply formatted, no tables, lists or graphics. In these cases, use of the Word format is *wasteful and dangerous* and should be avoided.

Reasons for using plaintext instead of MS Word whenever possible (see `http://www.netby.dk/Oest/Europa-Alle/vermeer/plain.html`):

1. Wasting resources (disk space, bandwidth).
2. Not everybody can read an MS Word document.
3. The danger of Word virus infection.
4. Other things you didn't plan on sending...

So:

> Plaintext – don't settle for more!

3.2 Write-only formats

There are situations, quite many actually, where plaintext won't do. You need to apply complex formatting to a document, it contains tables, lists or graphics, etcetera.

If your intent is to just communicate something to your recipients one-way, it is still a bad idea, generally, to use a word processor format. The virus problem is still the same, and another problem is that the visual appearance on-screen will depend on the version of Word, and printer drivers, the recipient happens to have installed.

Instead, *use PDF* (Portable Document Format). This format was designed by Adobe, but is a public standard that can be written and read by various non-Adobe software as well. PDF is used extensively on the World Wide Web and a version of Acrobat Reader is pre-installed on many desktops.

Other alternatives include PostScript (an older format by Adobe somewhat similar to PDF), HTML (the language of the World Wide Web), or RTF (Rich Text Format, a textual representation of word processor documents).

3.3 Collaboration formats

Collaborative authoring of documents require the use of a true word processor, e.g., Microsoft Word. Before settling on a word processor, make sure that all participants in the collaboration will have access to it. E.g., access

to MS Word presupposes access to the Windows operating system[1], which again only runs on Intel processor hardware. The choice of MS Word as the collaboration platform thus immediately and severely constrains other aspects of platform choice.

Fortunately this situation is changing. The software StarOffice is functionally equivalent to MS Office, to the point of being near-100% document format compatible[2]. Also the user interface is very similar to that of MS Office. So, a working group settling on using MS Word as their authoring platform could advise those members without access to, or unwilling to settle for, MS Windows or Intel hardware – or reluctant to part with the not unsubstantial licence fee for MS Office – to install StarOffice instead.

StarOffice, currently a Sun Microsystems product, is available free of charge for educational use. A parallel Open Source product based on the same source code, OpenOffice, is available free of charge as well. The software runs on a variety of platforms including Windows, Linux, Macintosh OS X, Solaris and other UNIX versions.

Going one step further, once some members in a collaboration group are using StarOffice, an easy way to re-establish 100% intercompatibility would be to ask the whole group to do so. As StarOffice does not inflict any expences on the user (above and beyond the one-time effort of installing it, and its footprint on the hard disk), this seems like an obvious choice.

4 Web publishing

4.1 Why Web publishing?

When you have prepared a document that you want to share with many people, of course you can email it to everyone that might be interested. However, you put stuff into all receivers' inboxes, on all receivers' hard disks. A useless duplication of disk storage, and a uselessly duplicated use of bandwidth.

So, especially if your document is sizable – and especially if it is a word processor document, it tends to be –, put it up on your web site, and mail a *link* to the people interested in it. They can either click on it or ignore it, and their inbox doesn't fill up. Do this also if you are not actually intending to publish your document, i.e., it is aimed only at a closed circle of readers.

An aspect to consider especially if you are *publishing*, is the permanency and context that come with publishing on the Web. This is for two important reasons.

[1] . . . or to a Macintosh operating system, for those so inclined, similarly limiting choice of processor hardware.

[2] . . . at least for fairly simple documents, the kind most people write. But already things like formulas, multi-level numbering or active "fields" may cause conversion problems. But then, such documents are better written in a mature structured document format like LyX/LaTeX anyway.

Firstly, Web documents come in hypertext formats which make linking, *rich* linking, both to other related material on the outside, and to other parts of the same document set, easy. This is well known in the case of HTML; all popular HTML authoring environments make linking easy and straightforward, and the user is well advised to use this facility to maximum advantage.

Not so well known is that also the PDF format allows the inclusion of links, including links to URLs (Universal Resource Locators, i.e., linkable content) elsewhere on the Internet. Not all PDF generating tools allow this easily or at all, but it *is* possible to create documents containing not only hyperlinks to outside Web material, but links internal to the document too, making, e.g., the table of contents, literature references and referrals to figures and tables, "live" links that you only have to click on to be taken to your destination. A major useability feature.

A second reason is enabling the use of *search engines*. These Web facilities, the first of which was Alta Vista, are continually traversing the Web and indexing the text of new and modified pages they find, enabling users to find things based on subject, rather than having to have the correct URL or following only links that thoughtful Web authors have provided. Of all facilities that the World Wide Web has brought us, search engines are undoubtedly the one which adds the most value and has prevented the Web from degenerating into an "information jungle" where you can read anything and find nothing.

But... using seach engines competently is a seriously non-trivial skill. Invest some effort in learning it, it will pay off handsomely.

Finally, even when "publishing" only within a limited corporate or institutional circle on an "intranet", it is possible to take advantage of the two value added features: linking and indexed search. The second one does require to set up your search engine and indexing robot, though.

4.2 Do's and don'ts

The Web is an ideal medium for sharing information with many people. However, to make the sharing optimal and painless, there is a number of things one should, and a number of things one should not do.

On the positive side:

- Keep it simple
- Split pages that are getting too big
- Stick to (World Wide Web Consortium, W3C) standards.

On the negative side:

- Don't use facilities that some of your viewers are unlikely to have. Flash, Java, JavaScript. Sometimes using these facilities makes sense; if so, you will know about it. If you don't, don't use them.

- Don't put up huge graphics, as this is both unnecessary and wasteful of bandwidth. In the rare case where you have a legitimate need to post a huge graphic, put it behind a clickable "thumbnail image".

  As a rule of thumb though, pictures should not occupy more than 10% of the nominal storage size (see `http://www.hut.fi/~mvermeer/picturethis.pdf`), where the nominal storage size is computed from the height and width of the picture, and the number of bytes per pixel needed to represent the colours (typically 3 bytes for full colour, 1 byte for greytone, and 1 bit, or 0.125 bytes, for monochrome). For video clips, multiply by the number of frames. Effective compression should yield file sizes 1-5%, at most 10%, of nominal size.

- Don't let your file names end in .htm, use .html instead. My toes curl upward when seeing this fossil of the MS-DOS era!

4.3 Learning HTML

While excellent WYSIWYG HTML editors are available that make the job of maintaining sets of Web pages easy and produce quality code to boot, actually learning to write HTML code the brute force way is an investment in knowledge that pays off handsomely.

The problems with many commercially available HTML authoring tools are:

- They automatically rely on things that are not available in all browsers. This make Web pages produced by them less widely readable.
- A proprietary product like FrontPage produces visually very good looking Web pages – in Internet Explorer. There are already thousands of Web pages that look good only in IE, but look poor to illegible in any other browser such as Mozilla or Opera. So avoid tools that are specific to browsers from the same company.
- A product that was never really meant to produce Web pages, but can be harnessed to this job, like Microsoft Word, tends to produce messy, over-tagged HTML code (which again tends to be IE-specific). Avoid this, or clean up the produced code manually.

Good WYSIWYG Web authoring tools do exist, e.g., Adobe GoLive. At a pinch, Netscape Composer will do for individual pages. If you go for WYSIWYG tools, make sure to have a look at the produced code in an ordinary text editor sometimes.

A versitile tool for cleaning up existing Web pages, and detecting things that are wrong with them, is HTML Tidy (`http://tidy.sourceforge.net/`). A Web interface is at `http://www.thedumbterminal.co.uk/services/tidy.shtml`. It even does a decent job on MS Word -generated pages. The World Wide Web Consortium also provides a testing tool for standards compliance of HTML documents (`http://validator.w3.org`). *Use these.*

And have a look at the HTML editing mode of Emacs!

4.4 Accessibility

Imagine if you are blind or of poor eyesight and you are trying to read many of the all-singing, all-dancing web sites out there. First off, if the web site contains text fields implemented as graphics – GIF, JPEG, PNG or whatever – you won't be able to read it. The text is not present as text, but as "pixel soup".

This is the reason why there is such a thing as the ALT tag. It allows you to provide an alternative representation of a graphic, or just a short text explaining what the graphic is, for those that – by choice or by necessity – are using a text-only browser such as lynx.

For visually impaired people, using a text-only browser is a necessity: the text is either converted to speech in a synthesizer, or to Braille script on a tactile interface, a process that is nigh-impossible to do with "pixel soup" text.

Try to imagine also what it must be like for vision-impaired people to try to read web pages that are layed out chaotically, with glaring colours, flashing animations and things that change on mouse-over. No, *don't* imagine it, look for yourself: a text-only browser like lynx will confront you "graphically" with what visually impaired Web users have to put up with every day.

A lot of relevant material can be found at the W3C's Web Accessability Initiative (WAI, `http://www.w3.org/WAI/`).

There are many people that are not visually impaired but still prefer a text-only browser – or turn graphics off in their browser – for the following reasons:

- Over a slow line, not downloading graphics speeds up everything else;
- In trained hands, navigating by keyboard beats mousing hands down;
- Most irritating graphics are advertisements;
- Somehow the most worthwhile web sites, offering most useful information and least fluff, view best in text-only browsers anyway.

Therefore: cater for these people!

5 The future of collaborative authoring

5.1 Versioning systems for software

Version control, or versioning, systems for software source code have existed for a long time. The idea of a versioning system is to keep track of the development path of the software. Ideally, it should be possible to

1. extract any historical version of a source code file;
2. extract any difference between successive, or non-successive, versions of the same code;
3. extract the identity of the developer that applied a change, or delta, as well as any explanatory comments he added.

A source code version control system that can be used for single files is RCS, Revision Control System. Typically this is not enough, however, and we want to be able to

1. track the development of complete file sets or trees together, and
2. allow multiple developers to work on different parts of the same body of code simultaneously.

For this purpose we have systems like CVS – Concurrent Versioning System. CVS keeps a copy of the current state and history of the source file set in a *repository*, from which developers can extract it to their local disks. After making modifications, they can check these in again to the repository.

Different developers can work upon different files within the file set without conflict. Also working on different parts of the *same* file works without problem. It is only when developers work *simultaneously* on the *same* part of the *same* file, that a problem arises, a conflict, which must be resolved manually at check-in time.

Software version control systems can be harnessed for sets of textual documents as well, e.g. for trees of HTML documents on a Web site. They are often used in this capacity. It does require, however, that the documents handled are plain text and *line based*. This makes it unsuited for binary, paragraph based word processing documents.

5.2 Visual versioning

Modern word processors contain their own versioning systems. E.g., Microsoft Word, and Sun StarOffice, contain a facility to mark text modifications – additions and deletions – as done by a particular author, and display these changes in colour.

This is a very convenient feature in collaborative environments where a single, official document has to be commented on in the draft stage by a number of contributors or stakeholders. After the document has "done the round", the changes proposed can be referred to and merged individually, and the result is a document including all of the accepted changes without any further correspondence needed.

But even for a single author, versioning is useful: one can "roll back" a change that upon further consideration appears unfortunate or wrong, as all historical versions can be extracted.

One unfortunate practice still often used when sending documents around for commenting or modification is the use of email for this purpose. What this leads to is a cyber-trail of "corpses", outdated versions of a document, littering inboxes around the planet. Taking up substantial hard disk space, as word processor documents have a habit of not being small.

5.3 WebDAV and Delta-V

Wouldn't it be great if it were possible to post a document on a Web site – as can be done for publishing it in read-only mode – but in such a way that people could also edit it? This is precisely what the WebDAV protocol (`http://www.ics.uci.edu/~ejw/authoring/`) tries to do. It comes as close as it gets to the dream of the "writeable Web".

DAV stands for Distributed Authoring and Versioning; the idea is that anybody with the proper user ID and password can access a document posted on a Web site – or a set of related documents, like the Web site itself – and modify it. A locking mechanism is provided so that not inadvertently two people are editing the same file at the same time, which would lead to the "who saves last, saves best" syndrome.

WebDAV is still a very young protocol, but already many applications exist that support it. Delta-V, the versioning system that goes with WebDAV, is younger still and not yet very widely supported. There also exist *filesystems* that support WebDAV: for Windows, there are "Web folders" which can be handled like any other disk drive. Files on it can be copied, moved, edited or deleted. For Linux, there is the similar davfs.

So, this is the thing to do in order to collaboratively edit a document: place it on a DAV-enabled Web server! Apache with mod_dav will do, as will IIS. After that, with proper client software installed, the document will appear in the local file manager and can be clicked upon and edited just as if it were a local file. Even if the server is in Europe and the author in Australia!

Do remember, though, to take regular backups of your document...

5.4 Useful links

A WebDav and Delta-V tutorial:

`http://www.webdav.org/deltav/WWW10/deltav-www10.htm`

`http://www.webdav.org/deltav/WWW10/deltav-www10.pdf`

A list of group work, networking and collaborative authoring solutions:

`http://www.svpal.org/~grantbow/groupware.html`

An example of proposed use of WebDAV:

`http://www.cio.noaa.gov/hpcc/2002/col_ce_11.html`

A report on WebDAV adoption:

`http://lists.w3.org/Archives/Public/w3c-dist-auth/2000JanMar/0119.html`