Web based information systemes

Dynamic Webpublishing with XML

D R A F T – Revision: 1.25

Thorbjørn Ravn Andersen

April 18, 2000

Dette dokument er under udarbejdelse.

Al tekst i dette skriftsnit er kommentarer til mig selv som jeg skal kigge på senere, og skal derfor vurdere

0.1 Abstract

This thesis discusses database backed websites; their current use as presentation engines, construction, pros and cons, list the current state of the art software, and how web sites in general function as *information presenters*. Hereafter it is discussed how end-users best can utilize such a system, without giving up their current software, allowing such websites to be used to ease the distribution of information between users, and let the web site provide *information sharing*.

as well as list my path to interest for meta-data and content-descriptive languages.

Technologies are discussed with an emphasis on Open Source software, and two sample programs - "Cactus" and "Consensus" are presented, and discussed. Cactus allows any user to do automated SSP on an Intranet without a human webmaster to convert documents and maintain links. Consensus allows an open group of people to maintain a set of documents on a webserver.

Stuff to place: Java has unicode support, multithreads.

Note: Framed text is unexpanded keywords which eventually will be replaced with prose.

Chapter 1

Overview

This section is rudimentary – it must be written at the end. Currently there is just keywords.

In this thesis I refer to the author Annie, and the layouter Larry.

1.1 What is a "database backed webserver"?

A web-server is a program which hands out files to any web-browser that asks.
A database "is a spread-sheet that several people can update simultaneously" (*greenspun*).
A database backed webserver then is simply a web-server which knows how to talk to a database server while servicing bypassing web-browsers.

1.2 Why use a webserver in combination with a database?

In one word: *Synergy*!

The web browser provide a simple, efficient and well known user interface available to almost every computer user in the world, and the database provide efficient dynamical access to data.

The combination of the two allow for the personalized Internet where a given webserver can give any user customized information upon request. Examples are:

- **Newspapers** – the user may specify that she wants in-depth coverage of international affairs, and do not want sports, and her personal web-edition will then just contain that. Payment may be per-article instead of per-paper.
- **Maps/Airlines/Hotels** – the user may request the best way to travel from A to B at a given time, see the route, and order accompanying reservations on-line.
- **Real stores** – the user may browse and order from the inventory. The ordered goods are then delivered by mail, delivery boy or similar.

- **Virtual stores** – the user may browse and order from a virtual inventory. The order is then analyzed and forwarded to the appropriate suppliers to the virtual store.
- **Banks** – there is a growing need from customers to be able to do home banking. If the bank offer a browser based solution, they can support customers regardless of their choice of computer. The transactions done through such a system will most likely be stored in a database to avoid data loss.
- **Program development** – The Mozilla project[1] has shown the benefit of up-to-date information regarding just about anything related to the source code.

Since its inception in 1993 the web has grown to be the "good enough for most things" graphical user interface, capable of showing text and graphics as well as provide navigation. Due to the simplicity of HTML it is not hard to build a plain basic client[2] meaning that just about any modern computer with Internet capabilites have browsers available for them.

Provide background information: Quick overview of the technology [webserver, database, http, sql, dynamically generated content vs] with a graph. Give simple example. Explain *when* things happen (on-the-fly vs statically generated pages). Explain the advantages and disadvantages databases have over flat filesystems [transactions(protection, atomicity), extra attributes, speed, indexed columns, complex queries, scalability(linear search in filesystems, multihost db's),]

1.3 The new role of the webserver

Compare original functionality with static HTML-files with content and a few CGI-scripts, to the current dynamically generated sites with many hits pr day. Since HTML is generated and provides little abstraction it is hard to use on a higher abstraction level, and with unreadable code in ASP and PHP3 it is harder. CSS was not the answer since it did not provide any abstraction from the presentation (seperate content from layout).
Different needs of the user - wap/pdf/html/braille. CPU, storage, webmaster time prevents all formats being pregenerated.

Documents must be written in a format describing *content* and not layout.

[1]...Mozilla project...`http://www.mozilla.org`

[2]Plan basic client implies no Java, no Javascript, no cookies, no plugins, no cascading style sheets, no HTTP-1.1 and almost everything else that characterize the modern browsers from Netscape, Microsoft, Suns and Opera $\boxed{??}$. However, if a webpage is well-written it can still be displayed on such a plain basic client.

Chapter 2

Terms and concepts

"HTML is a SGML DTD!"
Me – 1994

This report uses a lot of abbreviations, many of which may not be widely known. This section lists mosts of them.

The list will be sorted

entity A named "unit" in XML/SGML which expands to a string. This is very similar to a macro without arguments in other languages.

XSL-FO XSL-formatting objects. A generic description of the physical layout of a XML-document on paper. These files can be converted to PDF with FOP or PassiveTeX

SGML A family of languages. The *DTD* specifies which language it is. SGML can be formatted with FOSI or DSSSL style-sheets. "`jade`" can format to HTML, plain text and RTF.

XML XML is a light-weight version of SGML (both defines document languages) designed to be embeddable in a browser. XML-documents *may* comply to a DTD, or be stand-alone ("DTD-less"). XSL is used to convert a XML document into another XML document (XSLT) or into the generic page description language XSL-Formatting Objects (XSLFO).

***ML** Common description of both XML and SGML, where the two have the same applications.

DTD The Document Type Description is the specification which describes the exact syntax of the *XML

XSLT Either the process of *transforming* a XML-document into another XML-document using a XSL-stylesheet, or the process of *formatting* a document into FO, which then can be further formatted in PDF.

DOM The Document Object Model is W3C's first model for representing an XML tree internally. It requires the whole XML dataset to be read into memory which is a limitation for large projects. The later SAX interface is more lenient in its memory requirements.

SAX Simple API for XML. An event based approach to XML-parsing and representation. Has an advantage in that the design allows processing to begin before the whole XML-tree has been read in. Parsers and processors which implement SAX can be interchanged freely, currently allowing for 20 different combinations of parsers and processors.

HTML Hyper Text Markup Language. The "language of the web" which was originally designed by a physicist to present articles to other physicists. Was later hacked upon by Netscape and Microsoft to do things it was never originally meant to do.

DSSSL SGML Document Style Semantics and Specification Language. It is used to render an SGML document to another format. "`jade`" can render to RTF, TeX (must be post-processed with "`jadetex`"), SGML and XML.

FOSI Another style sheet for SGML, which I do not know anything about. The US Navy have a big repository[1].

SQL The Structured Query Language is the standard language for communicating with a database.

XSP `XML Servlet Pages` - A technique for combining code with XML in a single page, which is parsed and executed by the webserver when it is requested.

RTF *Rich Text Format* - a document description language designed by Microsoft. Widely supported. May contain extensions to the original RTF-specification.

Jade A DSSSL-engine for SGML documents by James Clark.

JPEG An efficient method for representing photographs in computer files. Efficiency is achieved by discarding the least visible parts of the visual information to the human eye.

GIF An image format well suited for computer generated images with few distinct colors, like icons. Is hampered by a patent on the compression scheme used and a maximum of 256 colors. Superceeded by PNG.

PNG The successor to GIF. Is supported in newer browsers only.

CGI The Common Gateway Interface. The original way to generate pages and other files dynamically. A CGI-script can be written in any language supported by the web server.

IIS Microsoft Internet Information Server. The webserver from Microsoft.

ftp File Transfer Protocol. This is both the name of the *protocol* (the method) as well as the *program* implementing the protocol. ftp can transfer files between a client computer and a server computer, and is a standard on the Internet

http Hyper Text Transfer Protocol. This is the protocol a web browser needs to speak with a web server, in order to retrieve documents. Http is intentionally very simple.

TCP/IP This is the procotol which a computer needs to speak to connect to the Internet. It allows two computer to establish a data stream, where one machine pushes bytes in one end of the stream, and the other machine retrieves the bytes from the other end of the stream, without either computer being concerned about the underlying network.

SP SP is a SGML parser[2] by James Clark.

[1]...The US Navy have a big repository...`http://navysgml.dt.navy.mil/dtdfosi/repository.html`
[2]...SP is a SGML parser...`http://www.jclark.com/sp/index.htm`

Perl A scripting language which has become popular with Unix system administrators, and which "was there" when a need for scripting languages for CGI arose. Has eminent regular expressions and is generally well-suited for handling text.

ASP Microsofts version of a scripting language intertwined with HTML. Works on Microsoft web servers only.

JSP Java Server Pages. Suns version of Java intertwined with HTML. Requires a servlet-capable web server.

PHP The Internet version of a scripting language intertwined with HTML. Can run as a CGI script in most browsers, but as a module (which is faster) in at least Apache.

Apache Open Source webserver which runs on a lot of different platforms. Most Linux distributions include it.

PassiveTeX A package for TEX by Sebastian Rahtz which allows FO files to be typeset by PDFTEX directly into PDF. PassiveTeX understands MathML, but does not yet support tables.

MIME Multipurpose Internet Mail Extensions[3] which extends the format of Internet mail to allow foreign character sets and inclusion of files.

MathML a XML-dialect to describe mathematical formulas.

DocBook A set of DTD's freely available from the Internet. See 6.1 on page 63.

TEI A DTD freely available from the Internet.

MIP The Maersk Mc-Kinney Moller Institute for Production Technology

[3]...Multipurpose Internet Mail Extensions... `http://www.oac.uci.edu/indiv/ehood/MIME/MIME.html`

Chapter 3

Dynamically generated documents

"404 Document not found"
–

3.1 The way it started: Handmade web-pages in a filesystem on a web-server

Back in the old days when the world wide web was designed, a web server could basically do only two things:

- Serve *static* files directly from an underlying file system. These files were either pages written in HTML, or images in GIF or JPEG formats.
- Call a program complying to the *Common Gateway Interface* with user-submitted parameters, and return its output as the document to send.

That was all.

Such web servers still exist - for example the NCSA server[1] (which has been unmaintained since 1996 – NCSA recommend that the Apache server should be used instead). Even so the February 2000 Netcraft survey[2] reported that 17172 NCSA servers were active, and could be reached from Netcraft.

In August 1995 the number of webservers running the NSCA server was 10835[3], meaning that number has increased slightly in that period. For comparison the total number of public webservers on the Internet in the same period grew from 19 thousand to 11 million, with Apache and Microsoft Internet Information Server accounting for 8.8 million of these.

Even with such a simple model, it works well for even *very* large web-sites which only generates few documents dynamically.

[1]. . . the NCSA server. . . `http://hoohoo.ncsa.uiuc.edu/`
[2]. . . the February 2000 Netcraft survey. . . `http://www.netcraft.com/Survey/Reports/0002/`
[3]. . . webservers running the NSCA server was 10835. . . `http://www.netcraft.com/Survey/Reports/9508/ALL/`

A well-tuned server which serves documents directly from the filesystem can reach impressive numbers. ACME software has a webserver benchmark[4] from 1998, which have benchmarked several webservers, where Apache did not come out top.

On a modern PC the Apache webserver has no problem saturating a 10 Mbps Ethernet connection , meaning that even for very large and demanding sites the bottleneck should be determined by the hardware! Phillip Greenspun talks about databases with added webserver capability in [Greenspun(1999)], and conclude that these are generally much slower than there non-database relatives.

On a related note: The ftp-server `ftp.cdrom.com` serves a *lot* of files every day. On March 14 2000 the transfer statistics[5] said that the average output that day was 79.3 Mbit/s (with a peak of 95.8 Mbit/s). The average output on a yearly basis was 86.2 Mbit/s (which is 933 Gb/day in average). This single machine runs FreeBSD[6] (see www.freebsd.org[7] for details). Since an ftp-session have a notably larger overhead than an http-session, it is not unreasonable to expect similar performance from a suitably tuned web-server serving static files.

3.1.1 CGI-scripts

Calling CGI-scripts is another matter. For the occasional dynamically generated document CGI-scripts turned out to work well. All kinds of documents – images, webpages, progress reports – could be generated comparatively easy in the favorit language of the CGI-script author. Libraries to deal with the decoding of parameters, and encoding of the generated result were soon abundant, providing for many enhancements to the original “static web site” model.

One of the first demonstrations of dynamically generated images was the Xerox PARC Map Viewer[8] (see figure 3.1 on the next page) which allowed the user to navigate a virtual atlas, where each “window” to the map was generated on demand. This initial quick hack has since been superceeded by professional map companies which produce maps of an uneven quality on demand, like Yahoo Maps[9] (USA only) and MapQuest[10] (covers Europe too).

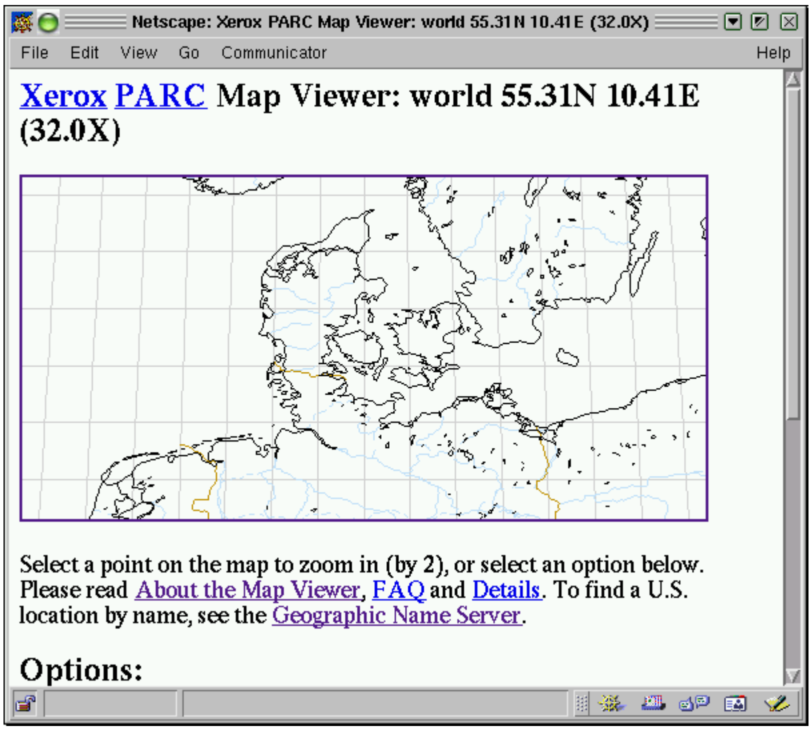
The problems with CGI-scripts have several reasons:

- **Scripts run as subprocesses** – the script is executed with the same permissions as the webserver itself, including access to `/etc/passwd` and other possibly sensitive files. Such scripts had to be *trusted* or - for Internet Service Providers - inspected before installation to ensure that the script would behave properly.

  This has been addressed with the development of “safe languages” where the execution environment is guaranteed that the CGI-script is confined to a “sandbox”.

- **Incompatible platforms** – the CGI-programmer may not have access to a C-compiler which can generate binary code for the machine running the webserver.

[4]. . . ACME software has a webserver benchmark. . . `http://www.acme.com/software/thttpd/benchmarks.html`
[5]. . . the transfer statistics. . . `http://www.emsphone.com/stats/cdrom.html`
[6]. . . This single machine runs FreeBSD. . . `ftp://ftp.cdrom.com/.message`
[7]. . . www.freebsd.org. . . `http://www.freebsd.org`
[8]. . . Xerox PARC Map Viewer. . . `http://mapweb.parc.xerox.com/map`
[9]. . . Yahoo Maps. . . `http://maps.yahoo.com/py/maps.py`
[10]. . . MapQuest. . . `http://www.mapquest.com`


Netscape: Xerox PARC Map Viewer: world 55.31N 10.41E (32.0X)
File
Edit
View
Go
Communicator
Help

Xerox PARC Map Viewer: world 55.31N 10.41E (32.0X)

Select a point on the map to zoom in (by 2), or select an option below. Please read About the Map Viewer, FAQ and Details. To find a U.S. location by name, see the Geographic Name Server.

Options:

Figure 3.1: The original Xerox PARC Map Viewer

This has caused interpreted languages like Perl to be very popular. Perl programs can be run immediately on a given platform without needing to be recompiled, and are for most purposes as fast as the equivalent programs written in `C` or `C++`, especially for users with slow modems. Recently Java Servlets (see section A.5 on page 124) have appeared as a popular and well-supported alternative to CGI-scripts, which is being supported by more and more web servers.

- **Slow** – the overhead just for invoking the CGI-program in a subprocess is substantial even for moderate load on the web server, since the web-server must "fork" a new process for each request.

  This has been addressed by putting the execution environment inside the web server, using threads instead of processes, and caching compiled versions of scripts.

Several solutions to the above problems have emerged. Of these, Java Servlets are discussed in section A.5 on page 124, PHP3 scripts in section A.5 on page 124, and Perl (with the mod_perl acceleration module) in section A.5 on page 124).

Server side scripting never really took off before the PHP3 and ASP languages provided "persistent connections" to a database servers, which was unavailable to the standard CGI-scripts.

Persistent connections provide a real performance boost! Database connections are notoriously "expensive" to establish, so needing to open one for each and every CGI-script were a true performance bottleneck. Just by keeping the connection open (and for ASP-scripts - retain the particular database connection for the session) was extremely beneficial. That combined with a very simple way to switch between code and HTML plus that every user could use it without needing to ask the webmaster to install a potentially dangerous CGI-script, meant that the ability to generate webpages dynamically became available to everyone.

3.2 When a site grows, it becomes hard to maintain

With the growth of the Internet many webmasters find that their site has grown way beyond what they originally expected, and that

- their handwritten HTML pages are hard to keep up-to-date. A minor change in site layout, must be manually updated in all the affected files. Creating scripts to do this, helps but is still a non-trivial task
- The information tree described by the links is very difficult to change, since all links must be checked and potentially changed.
- When pages are moved or deleted, it becomes difficult to ensure that all links pointing to these pages are dealt with appropriately. Jakob Nielsen[11] uses the term *linkrot* to describe web sites with many broken links.
- In the case of external links it becomes even harder to ensure that all links are valid at all times.

The answer to these problems is usually *automation*. A script to ensure that the “Last updated 2000-03-12 by whoever” string at the bottom of each page is up to date. A script to ensure that all internal links are valid. A script to update the layout of tables. The list is endless.

3.3 Navigational structure should not be maintained by the author

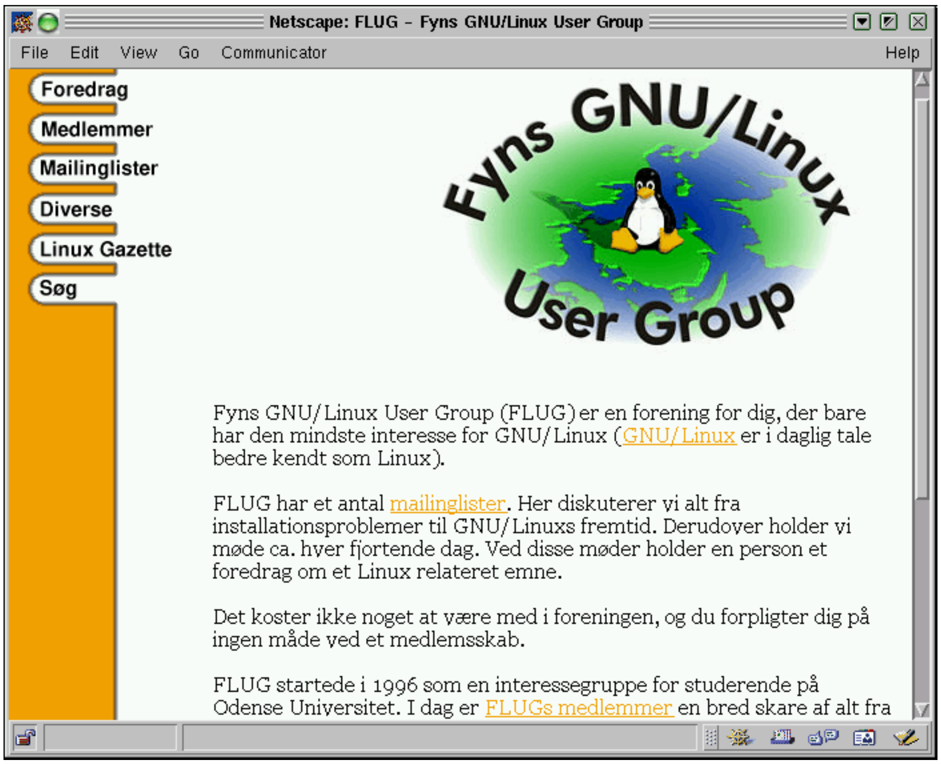
One of the easiest things to automate for a webmaster is the insertion of a few snippets of HTML in each web page to give a characteristic site layout. This relieves the webmaster of the tedious task of updating all the HTML-documents on the site manually.

Figure 3.2 on the facing page show a sample page from the Fyns GNU/Linux User Group web-site[12], showing the way FLUG chose to do it. The framework shown is added to each HTML-file with a small perl script.

This is a typical static framework for a site, where every web page basically is “inserted” in the home page since the set of links is identical for all pages. Usually this involves a link to a search engine for the site.

In the original web servers this required that every web page was modified to include the snippet of HTML which produced this header, which is a relatively easy task with a modern scripting language like Perl. Note that care must be taken not to change the modification times of the files, since failing to do so invalidates local copies in browsers and webcaches, even though that the contents of the pages is unchanged as such. (Modern webservers allow a webpage to include other files with Server

[11]...Jakob Nielsen...`http://www.useit.com/alertbox/980614.html`
[12]...Fyns GNU/Linux User Group web-site...`http://www.flug.dk`



Fyns GNU/Linux User Group (FLUG) er en forening for dig, der bare har den mindste interesse for GNU/Linux (GNU/Linux er i daglig tale bedre kendt som Linux).

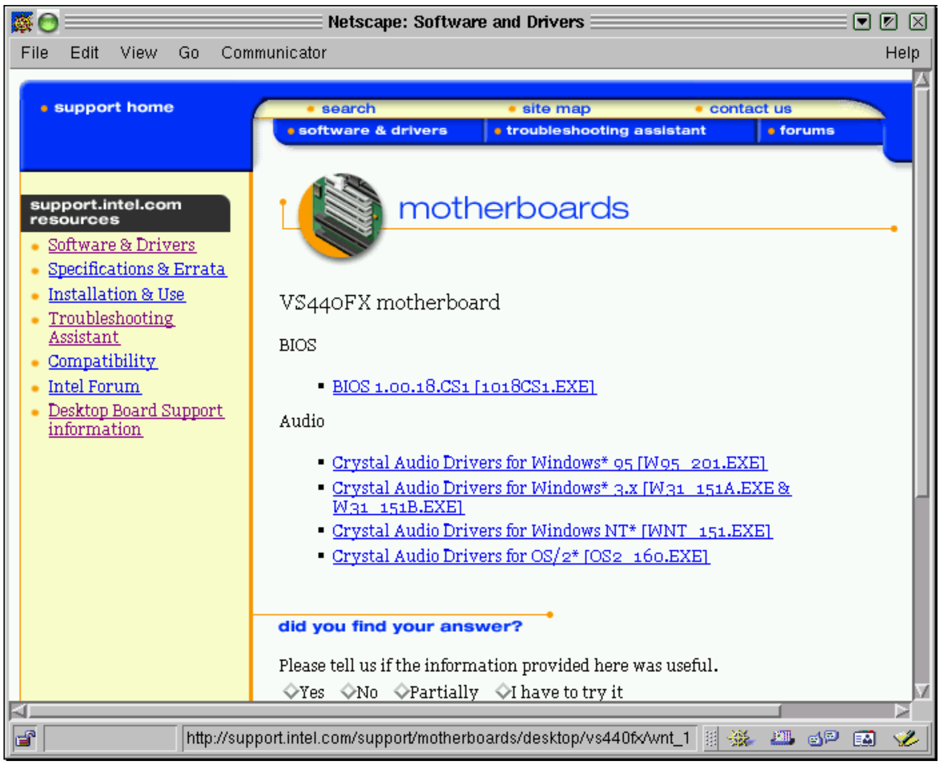
FLUG har et antal mailinglister. Her diskuterer vi alt fra installationsproblemer til GNU/Linuxs fremtid. Derudover holder vi møde ca. hver fjortende dag. Ved disse møder holder en person et foredrag om et Linux relateret emne.

Det koster ikke noget at være med i foreningen, og du forpligter dig på ingen måde ved et medlemsskab.

FLUG startede i 1996 som en interessegruppe for studerende på Odense Universitet. I dag er FLUGs medlemmer en bred skare af alt fra

Figure 3.2: The Fyns GNU/Linux User Group web-site. The side bar and the logo was added automatically by a script by Tobias Bardino

Side Includes[13], which takes care of all this - except doing it unconditionally on every page. The user must still remember to insert the appropriate command sequence in the server).


Netscape: Software and Drivers

File Edit View Go Communicator Help

• support home

• search • site map • contact us

• software & drivers • troubleshooting assistant • forums

support.intel.com resources

- Software & Drivers
- Specifications & Errata
- Installation & Use
- Troubleshooting Assistant
- Compatibility
- Intel Forum
- Desktop Board Support information

motherboards

VS440FX motherboard

BIOS

- BIOS 1.00.18.CS1 [1018CS1.EXE]

Audio

- Crystal Audio Drivers for Windows* 95 [W95_201.EXE]
- Crystal Audio Drivers for Windows* 3.x [W31_151A.EXE & W31_151B.EXE]
- Crystal Audio Drivers for Windows NT* [WNT_151.EXE]
- Crystal Audio Drivers for OS/2* [OS2_160.EXE]

did you find your answer?

Please tell us if the information provided here was useful.

◇Yes ◇No ◇Partially ◇I have to try it

http://support.intel.com/support/motherboards/desktop/vs440fx/wnt_1

Figure 3.3: The web page for drivers for the Intel VS440FX motherboard

Many modern websites put a given page in a *context* where the navigational framework reflect this context. See figure 3.3 for a sample from Intel regarding the VS440FX motherboard[14], where the actual web page has a top bar with generic links, a column with links specific to motherboards in general, and a review form at the bottom asking whether this information was useful to the reader (this question is asked on every page with technical content). My personal experience with the Intel site is that their search engine is efficient, but that the link column is too uninformative - you often have to actually follow the link to see what is there. An expanded view would be nice.

This site layout require a bit more discipline for the web author writing each page, but is still easily implementable since it can still be implemented by including the appropriate HTML-snippet, as these are the same for all pages regarding this motherboard. A way to easily ensure this, would be to let the name of the physical file determine the virtual location in the web server hierachi - the shown file could have been named `motherboards/vs440fx/drivers.html`. This may be difficult to manage (if the webmasters work on different machines) and it is not very easy to change if the global layout of the web server changes.

In order to ensure integrity in the navigational framework for a site this size, it should be generated automatically. Doing so requires meta-data about the individual pages in order to place them

[13]...Server Side Includes...`http://www.apache.org/docs/mod/mod_include.html`
[14]...Intel regarding the VS440FX motherboard...
`http://support.intel.com/support/motherboards/desktop/VS440fx/software.htm`

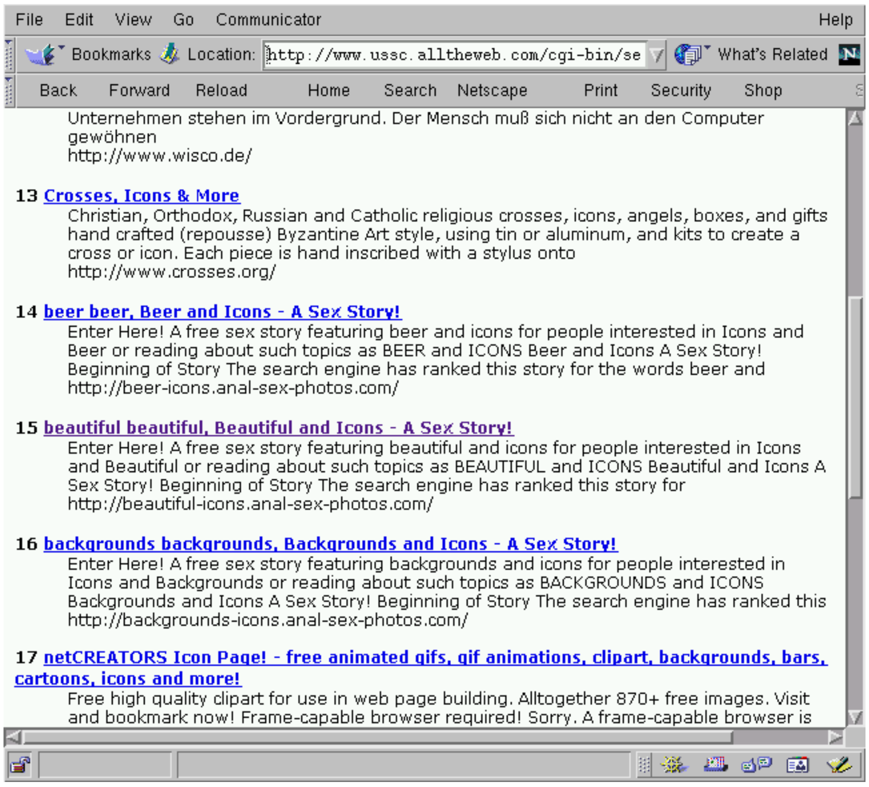
correctly in the framework.

3.4 A good website needs meta-information about its documents

What to a webmaster is a nice, and well maintained web-site, is to a computer just a bunch of directories of files with bytes in them. In order to get any use of a computer in maintaining these, it is important to have easily accessible information about the desired functionality and the files to work with. Even though complicated rules can be constructed to extract information from webpages, the basic rule is still that

A human must enter the basic information for categorizing a given webdocument.

Computers are simply not good enough yet to guess this information themselves.


File Edit View Go Communicator Help
Bookmarks Location: http://www.ussc.alltheweb.com/cgi-bin/se
What's Related
Back Forward Reload Home Search Netscape Print Security Shop

Unternehmen stehen im Vordergrund. Der Mensch muß sich nicht an den Computer gewöhnen
http://www.wisco.de/

13 Crosses, Icons & More
Christian, Orthodox, Russian and Catholic religious crosses, icons, angels, boxes, and gifts hand crafted (repousse) Byzantine Art style, using tin or aluminum, and kits to create a cross or icon. Each piece is hand inscribed with a stylus onto
http://www.crosses.org/

14 beer beer, Beer and Icons - A Sex Story!
Enter Here! A free sex story featuring beer and icons for people interested in Icons and Beer or reading about such topics as BEER and ICONS Beer and Icons A Sex Story! Beginning of Story The search engine has ranked this story for the words beer and
http://beer-icons.anal-sex-photos.com/

15 beautiful beautiful, Beautiful and Icons - A Sex Story!
Enter Here! A free sex story featuring beautiful and icons for people interested in Icons and Beautiful or reading about such topics as BEAUTIFUL and ICONS Beautiful and Icons A Sex Story! Beginning of Story The search engine has ranked this story for
http://beautiful-icons.anal-sex-photos.com/

16 backgrounds backgrounds, Backgrounds and Icons - A Sex Story!
Enter Here! A free sex story featuring backgrounds and icons for people interested in Icons and Backgrounds or reading about such topics as BACKGROUNDS and ICONS Backgrounds and Icons A Sex Story! Beginning of Story The search engine has ranked this
http://backgrounds-icons.anal-sex-photos.com/

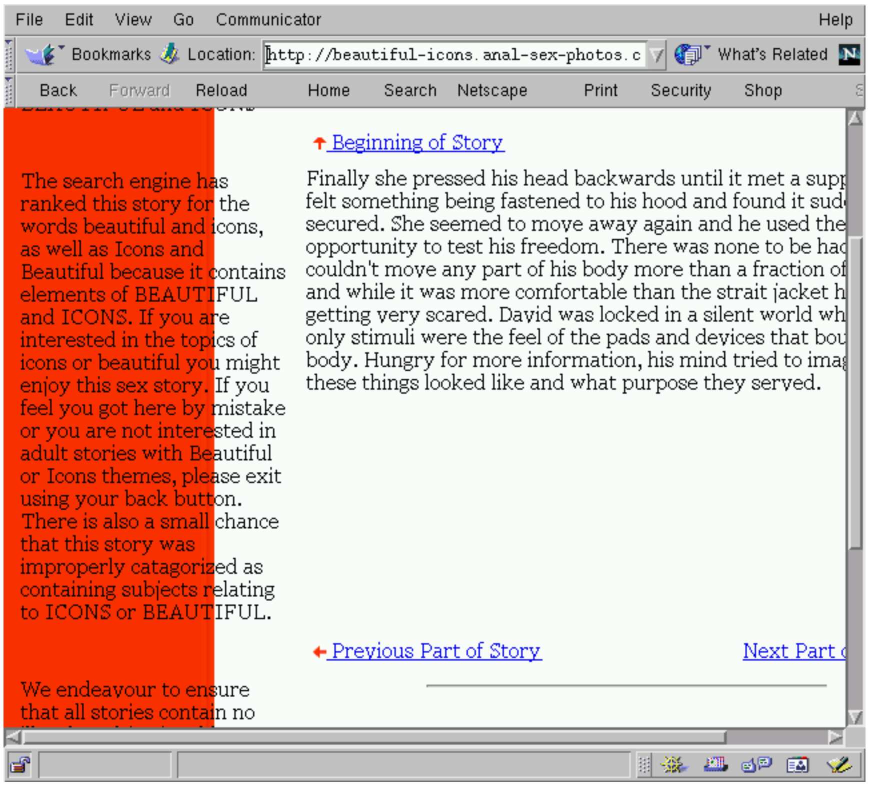
17 netCREATORS Icon Page! - free animated gifs, gif animations, clipart, backgrounds, bars, cartoons, icons and more!
Free high quality clipart for use in web page building. Alltogether 870+ free images. Visit and bookmark now! Frame-capable browser required! Sorry. A frame-capable browser is

Figure 3.4: AllTheWeb search for "background icons"

Incidentially this is also the reason why Internet search engines like AltaVista[15] needs elaborate information extraction techniques in order to remain useful. AltaVista was the first Internet Search engine that claimed to cover *all* webpages on the Internet. It was immediately a great success since the Internet had already grown to a size where help was essential to find any page if you didn't have a direct URL to it already.

[15] ...AltaVista...`http://www.altavista.com`

It didn't take long for web authors (especially those with adult material) to realize that the best way to get their web pages frequently returned in a top position at AltaVista was by putting as many potential search terms in META-tags in each and every of their web pages, showing that just blindly extracting keywords *uncrititically* is too simple a method. Additionally this drives users away – when they feel that they are not getting "the best results" from the search engine they will look for another one that can. The Google search engine[16] (see section 4.8 on page 43) was such an engine, and users quickly started to use it. Recently links to Amazon.com and Fatbrain.com have begun to crop up in the top of the search results on Google, and as a direct result users are going elsewhere.


File Edit View Go Communicator Help
Bookmarks Location: http://beautiful-icons.anal-sex-photos.c What's Related
Back Forward Reload Home Search Netscape Print Security Shop

The search engine has ranked this story for the words beautiful and icons, as well as Icons and Beautiful because it contains elements of BEAUTIFUL and ICONS. If you are interested in the topics of icons or beautiful you might enjoy this sex story. If you feel you got here by mistake or you are not interested in adult stories with Beautiful or Icons themes, please exit using your back button. There is also a small chance that this story was improperly catagorized as containing subjects relating to ICONS or BEAUTIFUL.

We endeavour to ensure that all stories contain no

Beginning of Story

Finally she pressed his head backwards until it met a supp
felt something being fastened to his hood and found it sud
secured. She seemed to move away again and he used the
opportunity to test his freedom. There was none to be had
couldn't move any part of his body more than a fraction of
and while it was more comfortable than the strait jacket h
getting very scared. David was locked in a silent world wh
only stimuli were the feel of the pads and devices that bou
body. Hungry for more information, his mind tried to imag
these things looked like and what purpose they served.

Previous Part of Story Next Part

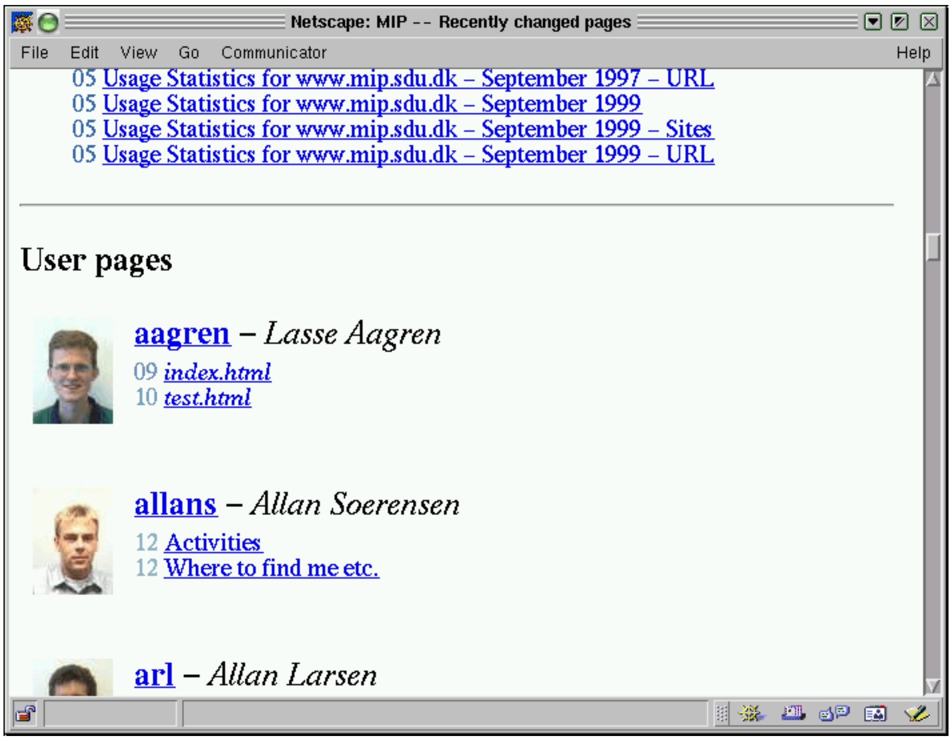
Figure 3.5: And what hit 15 was linking to in 3.4 on the page before

As an alternative to Google I have been recommended AllTheWeb[17], but which after a bit of use appears to be prone to the same polluting documents as AltaVista. See the result of "background icons" in figure 3.4 on the preceding page and what the fifteenth hit actually linked to in figure 3.5.

AltaVista failed then as AllTheWeb fails now, because they had no control over authors and the authors had a desire for "breaking the system". For further reading, Douglas Hofstadter talks a lot about the impossibility of building an unbreakable system for automatic detection of bad input in [Hofstadter(1979)].

An example of a document generated from meta-data which has been automatically extracted, is the MIP "Recently Changed Pages" (see figure 3.6 on the facing page) the original version of which I wrote while working for MIP. The script traverses three disjunkt set of web pages on the server -

[16]...The Google search engine...`http://www.google.com`
[17]...AllTheWeb...`http://www.alltheweb.com`


Netscape: MIP -- Recently changed pages
File Edit View Go Communicator Help

05 Usage Statistics for www.mip.sdu.dk – September 1997 – URL
05 Usage Statistics for www.mip.sdu.dk – September 1999
05 Usage Statistics for www.mip.sdu.dk – September 1999 – Sites
05 Usage Statistics for www.mip.sdu.dk – September 1999 – URL

User pages

aagren – *Lasse Aagren*
09 *index.html*
10 *test.html*

allans – *Allan Soerensen*
12 Activities
12 Where to find me etc.

arl – *Allan Larsen*

Figure 3.6: The MIP Recently Changed Pages. The last of the system pages and the first user pages are shown.

System pages, System Administrator pages and User pages (one set per user) - and generates a list for each set sorted by title. Each file listed has its age in days next to it.

The meta-information extracted was:

- **Title** - used for the link, and sorting the entries in a set. Extracted from the content of each document, with a default of the filename if no title was present.
- **Age** - extracted from the underlying filesystem which registers the last change of the document
- **User** (for the user pages) - also extracted from the filesystem. It is also used to look up the picture of the user.
- **Size** - which is intended to provide a hint to the user about the size of the page in question. It is not used in the depicted page.

This is just about all the useful information a flat Unix-filesystem can provide. Additionally nothing at all is guaranteed about the contents of a HTML-file - even the title is not even always there!

If more meta-data than the above is needed, the authors must be involved and as a part of their web-authoring, deliberately and carefully ensure that the meta-information needed by the automatic processes is correct and up-to-date.

Such an effort requires that the data is much more systematically organized than they are today, including better mark-up. I have not seen any site on the Internet yet, which is capable of *categorizing* the content of the Internet.

The *Yggdrasil system* (see section 4.12 on page 44) was my first attempt to generate a navigational framework from information extracted from webpages. Yggdrasil was to function as the automatic webmaster on the intranet, in order to avoid having to assign staff to do so. Then the individual employee could publish information in form of a webpage, add a category either in the title or as a META-tag, and trigger the next update of the framework. This update would scan all web-pages, and extract tuples of (author, category code, publishing date, title, size) of those web-pages which had a category code, and generate a tree structure of web-pages to navigate the categories. Additionally lists of "Documents sorted by author" and "Documents sorted by date" were generated to help users locate documents.

> This was on a closed Intranet, where the users were interested in using this as a tool. The incentive for "breaking the system" was very low, and the tool worked well.
>
> Yggdrasil worked very well initially especially when its very low amount of meta-information is taken in consideration.

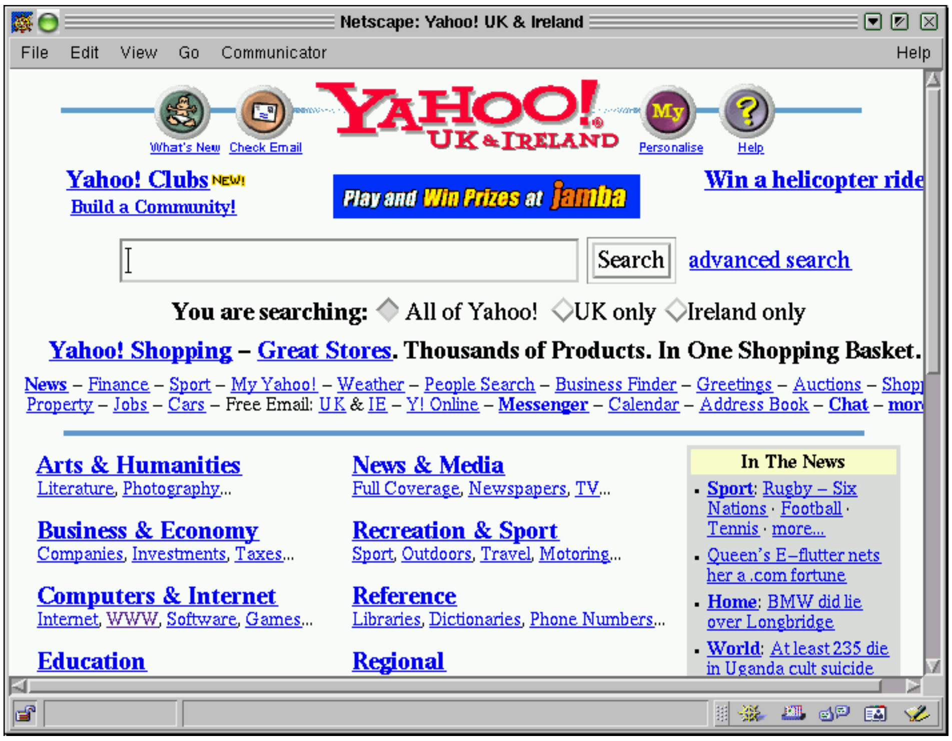

Netscape: Yahoo! UK & Ireland
File Edit View Go Communicator Help
What's New Check Email
YAHOO! UK & IRELAND
Personalise Help
Yahoo! Clubs NEW!
Build a Community!
Play and Win Prizes at jamba
Win a helicopter ride
Search
advanced search
You are searching: All of Yahoo! UK only Ireland only
Yahoo! Shopping – Great Stores. Thousands of Products. In One Shopping Basket.
News – Finance – Sport – My Yahoo! – Weather – People Search – Business Finder – Greetings – Auctions – Shopp
Property – Jobs – Cars – Free Email: UK & IE – Y! Online – Messenger – Calendar – Address Book – Chat – more
Arts & Humanities
Literature, Photography...
Business & Economy
Companies, Investments, Taxes...
Computers & Internet
Internet, WWW, Software, Games...
Education
News & Media
Full Coverage, Newspapers, TV...
Recreation & Sport
Sport, Outdoors, Travel, Motoring...
Reference
Libraries, Dictionaries, Phone Numbers...
Regional
In The News
Sport: Rugby – Six Nations · Football · Tennis · more...
Queen's E–flutter nets her a .com fortune
Home: BMW did lie over Longbridge
World: At least 235 die in Uganda cult suicide

Figure 3.7: The British Yahoo site

An example of a good site built with meta-information is Yahoo[18] (see figure 3.7) which started as a directory over web pages where the maintainers added meta-data to a lot of web-sites by categorizing them manually, and use this information to regularly generate static navigational pages. At a time Yahoo really suffered by a lot of broken links as the meta-data was not maintained, but today Yahoo usually links to a page that *is* there (by verifying that a target page existed).

[18]...Yahoo...`http://www.yahoo.com`

3.5 Documentation - meta-data of programs

Given there is a need for meta-data about the documents, the question is then *where* should the authors put the meta-data?

In the programming world, a very visible example of meta-data is the documentation for programs, since the computers themselves do not need them. Experience has shown that it is notoriously hard for programmers to keep the documentation up-to-date, since it is usually considered a part of the coding process that is not really necessary and definitively not felt to be a part of the “creative, fun” process of writing programs! If the writing of documentation was well established as an integrated part of program development, and the programming language and the programmers working environment itself helped as much as it possibly could, it would be easier for the programmers to keep the documentation synchronized with the code.

Experience has shown (*references man month*) that documentation should be as close to the thing it documents as possible. For programs, this means that the documentation should be *in the same file* as the code. Comments are usually just that – brief comments – and are often meant to explain *what* the code does instead of *why*? The abstraction level is not high enough in the traditional comments.

Andrew S. Tanenbaum writes excellent books about Computer Science, and his Operating Systems book [[Tanenbaum(1987)]] contains the complete source listing with line numbers of his Minix operating system, along with a cross-reference of all identifiers. That was the best the printing industry could do in 1987 (and is typical for several other printed versions of source code), and that was not good enough for teaching. It is very hard to get a grasp of what the code does, without long and careful studies, and much flapping back and forth. My guess is that today Tanenbaum would write a hyperlinked book readable in a browser, perhaps even using the Literate Programming techniques discussed in section 3.5.3 on page 21. The only other publicized works of large programs I know of is the books by Knuth about TEX and the PGP 5.0 source code (which was not printed for humans to read, but in order to scan the PGP source to circumvent the USA crypto export regulations[19].

The great news is that the industry recognizes the usefulness of documentation on the web, and the need for programmers themselves to create such documentation. In order to gain wide acceptance such meta-data management systems must be *standards*, either as *de-facto standards* or defined by a standards body like ANSI or W3C. No such standard has arisen yet, but a XML-dialect will most likely be written for the purpose.

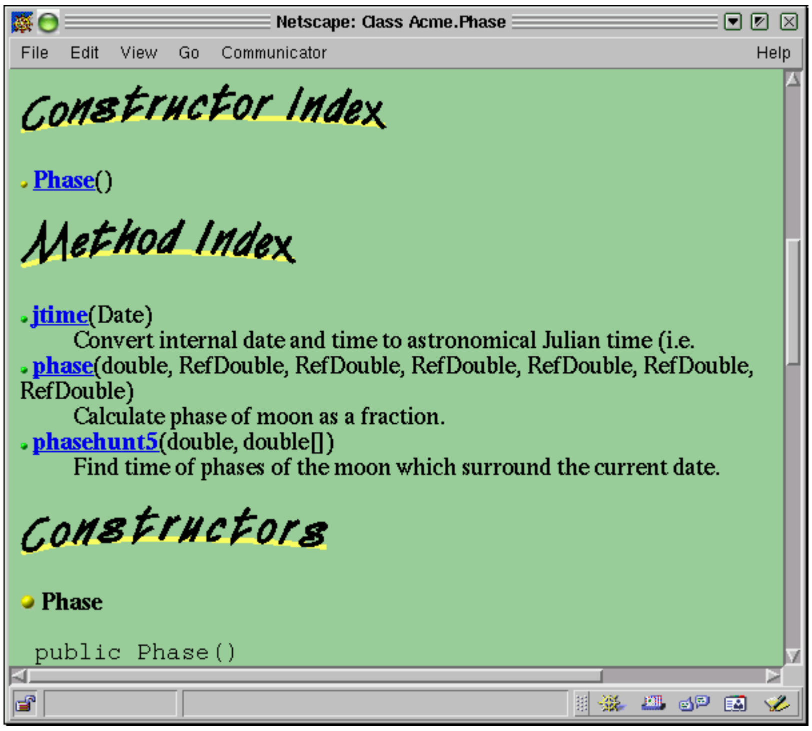
The rest of this section detours into other areas of computer science, where having several views on the data in question has proven beneficial.

3.5.1 Javadoc - embedding web-information in programs

This is perhaps the most visible and generally available meta-data tool for Java programmers today. JavaDoc[20] is a tool that creates documentation in form of HTML pages from Java source code with

[19]...scan the PGP source to circumvent the USA crypto export regulations...
`http://zone.pspt.fi/pgp/faq/pgp55scan.txt`

[20]...JavaDoc...`http://java.sun.com/products/jdk/1.2/docs/tooldocs/solaris/javadoc.html`


Netscape: Class Acme.Phase
File Edit View Go Communicator Help

Constructor Index

- Phase()

Method Index

- jtime(Date)
  Convert internal date and time to astronomical Julian time (i.e.
- phase(double, RefDouble, RefDouble, RefDouble, RefDouble, RefDouble, RefDouble)
  Calculate phase of moon as a fraction.
- phasehunt5(double, double[])
  Find time of phases of the moon which surround the current date.

Constructors

- Phase

```
public Phase()
```

Figure 3.8: The ACME documentation of the `Acme.Phase` class generated by JavaDoc

embedded comments, where all definitions are parsed and hyperlinked to give a full overview of the Java source code in question. See figure 3.8 for a JavaDoc rendered source code at ACME labs.[21]

JavaDoc is a specialized tool which is only suitable for generating web-pages from Java source code, but as Sun use JavaDoc for their reference documentation[22] as well as ship it with every copy of Java Development Kit, it is a tool which is widely available. Java programmers who need to document their code, will most likely use JavaDoc to do so.

Javadoc has also raised the expectations of the programmers, since they have become used to hyper-linked documentation in a browser to accompany any code they are to use. This tendency is a good step in the right direction.

Others recognize this too. On March 13, 2000 an open letter pleading for the release of JavaDoc as OpenSource[23] was posted to the Internet. They argued amongst other things that a number of bugs needed to be fixed, and new techology like XML/XSL should be employed too. Hopefully this will influence Sun to help the developers as much as possible by opening up this source too.

Meanwhile, an Apache effort to parse JavaDoc documents into XML[24] is underway. This has great potential when it gets running, since the dynamic publishing abilities of XML will complement

[21]. . . ACME labs.. . . `http://www.acme.com`

[22]. . . use JavaDoc for their reference documentation. . .
`http://java.sun.com/products/jdk/1.2/docs/api/overview-summary.html`

[23]. . . an open letter pleading for the release of JavaDoc as OpenSource. . .
`http://relativity.yi.org/WebSite/opensource-javadoc/`

[24]. . . an Apache effort to parse JavaDoc documents into XML. . . `http://xml.apache.org/cocoon/javadoc.html`

the target audience of JavaDoc, namely programmers who need reference documentation in an electronic form.

3.5.2 The Plain Old Documentation format for Perl

The perlpod[25] format was designed to be simple to write and easy to use in Perl programs, and the overwhelming amount of freely available documentation for Perl[26] confirms that this goal was reached. This format allows you – with very elementary markup – to put documentation and code in the same file, and you may leave out either one.

All the info-files at Unixsnedkeren.dk[27] (the website for my small firm) have been written as POD files, and converted to HTML, with some finishing touches done with another Perlscript modified for the one I wrote for the FAQ for the Unix echo in the Danish Fidonet[28]. Code truly lives forever, even though it is very difficult to locate the start and end tags of a given HTML-construction without parsing the whole file.

Due to this and all the rest of the *ad-hoc* code in both "`pod2html`" and my own code, I am convinced that another approach to rendering POD-files should be taken, and I have submitted patches to the "`pod2docbook`" command in order to create DocBook XML in addition to the current DocBook SGML. This will add the rendering possibilities of the XSL-world to those Perl already know about.

Recently a `Lip::Pod` filter have been added to CPAN which allows a mixed mode documentation output – where both code and documentation is output allowing a simple form of *Literate Programming*.

3.5.3 Literate Programming - Knuths approach to different views of the source

*"I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: "Literate Programming." Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do."*

Donald Knuth – [Knuth(1992)] quoted from The Literate Programming web site[29]

Donald E. Knuth has written the TEX-system used to typeset this report, and documented it by publishing the source code to the entire system in four books. The TEX-system has impressed since it is of an extremely high quality, both in code and documentation, and actually offer money to those who find errors in his code[30].

In order to write both documentation and code of such high quality, Knuth developed his own method of coding which he named "Literate Programming", in which the author works with a

[25]...perlpod...`http://www.cpan.org/doc/manual/html/pod/perlpod.html`
[26]...documentation for Perl...`http://www.cpan.org`
[27]...Unixsnedkeren.dk...`http://www.unixsnedkeren.dk`
[28]...the FAQ for the Unix echo in the Danish Fidonet...`http://www.fido.dk/faq/unix-faq/unix_r23.htm`
[29]...The Literate Programming web site...`http://www.literateprogramming.com/`
[30]...actually offer money to those who find errors in his code...`http://truetex.com/knuthchk.htm`

single file containing the documentation *as it is to be presented to the reader* written in a TEX-dialect, with the actual code (with a few characters escaped) listed in named chunks along with their documentation, in the order it fits the author to present them. This file format is the `web`-format! This way of programming is discussed in detail in [Sewel(1989)], and a lot of references are available on literateprograming.org[31]. A web-program can be processed in two ways:

- The `weave` tool converts the web-file to a `tex`-file to be processed by TEX, and printed.
- The `tangle` tool generates the actual code to be compiled, by joining chunks with the same name, and inserting them in other chunks that reference them (a simple macro expansion), eventually producing one or more flat files which can be compiled, along with information links each code chunk back to the original web document.

```
          sample.web
            /   \
           /     \
       weave    tangle
         /         \
        /           \
    sample.tex    a.c b.c ...
       |             |
       |             |
     ....         compiler
                     |
                     |
                  executable
```

The problem with Knuth is that things that work well for him, does not always work quite as well for the rest of us. No editors - not even Emacs - are currently capable of helping where a given file is separated in a lot of logical regions, being either TEX-code or source code, meaning that the editor cannot provide the supporting functions which a programmer might have become accustomed to.

You cannot code verbatim - some characters are reserved for other purposes and must be written differently, which may be very annoying to learn. Additionally the concept of chunks all over the document may be counter-productive if these are hard to navigate.

Many people have experimented with the possibilities of a weave/tangle pair resulting in several software packages, some of which are FunnelWeb[32], noweb[33], fweb[34] and nuweb[35].

I did a few projects with nuweb just to try it, which was language independent and therefore well suited to Perl, and I found it reasonably well working, when TEX and Emacs had been taught that the .tex and .pl files were derived from a web-file. I submitted patches for using nuweb with AUC-TeX.

There is a webring for Literate Programming[36], which is an excellent place to look for further

[31]...literateprograming.org...`http://www.literateprograming.org`
[32]...FunnelWeb...`http://www.ross.net/funnelweb/`
[33]...noweb...`http://www.eecs.harvard.edu/ nr/noweb/`
[34]...fweb...`http://w3.pppl.gov/ krommes/fweb.html#SEC3`
[35]...nuweb...`http://www.tu-chemnitz.de/ ukn/UNIX/nuwebdoc/nuwebdoc.html`
[36]...a webring for Literate Programming...`http://www.webring.org/cgi-bin/webring?ring=litprog;list`

information. The Collection of Computer Science Bibliographies provides a search engine in literate programming publications[37].

Oasis has a page on Literate Programing with XML and SGML[38] where a number of approaches are listed with references. This appears to be the only resource so far.

My personal experiences with Literate Programs can be summarized as:

- The documentation gets bigger and better, simply because the printed documents look better that way. What would pass as a single line comment in a source file, looks very small when typeset. The full power of TEX also encourages usages of illustrations and graphs.
- The program development gets cumbersome. You may have trouble using your favorite tools for editing, compiling and debugging. Users of most integrated development environments cannot use this model since the IDE does not provide hooks to provide this processing.
- A critical point is whether the intermediate files are visible to the author! In order to be usable, *all* derived files *must* refer to the original document in a transparent fashion, meaning that the user should not have to worry about intermediate files. (In the same way that the C-processor works under Unix).

My conclusion was that the Literate Programming paradigm does not pay off as a individual programmer, but may work very well for a larger programming team where good, current documentation is critical.

3.5.4 Rational Rose - the other way around

A software product which has been very successful in recent years, is the Rational Rose visual modeling tool[39] which is used by several people at MIP to do software development.

Rational Rose provides for a lot of things relevant to a large software project like reverse engineering of existing code, but also for several views of the document[40] depending on what abstraction level the authors are working on. In this way they are able to provide the author an environment where the software model can be designed seperately from the writing of the code.

Design patterns are used to provide commonly known building blocks to build up the abstractions.

Bo Nørregård Jørgensen told me that he would expect it to be beneficial to use Rational Rose in a software project when it has an underlying model and has 7 modules or more.

3.5.5 Compiling source code into an executable

Most software projects known to me does not consist of a single huge source file which would take ages to compile, but of several smaller files, which can be compiled individually and finally linked to an executable file. If a file with source code is changed, it is not necessary to recompile each and

[37]...a search engine in literate programming publications...`http://liinwww.ira.uka.de/searchbib/SE/litprog`
[38]...a page on Literate Programing with XML and SGML...`http://www.oasis-open.org/cover/xmlLitProg.html`
[39]...Rational Rose visual modeling tool...`http://www.rational.com/products/rose/index.jtmpl`
[40]...several views of the document...
`http://www.rational.com/sitewide/support/whitepapers/dynamic.jtmpl?doc_key=350`

every file but only those which depend on this particular source file, and then relink the executable to incorporate the changes.

This is possible because the authors has provided meta-data about the system, regarding which source files the system contains, which commands to call to compile and link the source files, which libraries should be included, etc. Normally an Integrated Development Environment (like Borland C++Builder[41], Microsoft Visual C++[42], or the Visual Age products from IBM[43]) knows about these things, or a Makefile[44] is constructed to utilize the known relations built into "`make`" as well as allow the author to add new ones.

The real advantage comes in environments where several compilations can run in parallel due to multi-threaded and/or multi-CPU systems. This can be done safely by using meta-data to localize compilations which are independent from one another and execute these simultaneously. MIP used to have a 24-CPU Silicon Graphics machine where a parallelizing "make" could reduce compilation times with a factor of 20 or more. Not all software projects were easily parallelized - if the Makefile did not list all dependencies explicitly but expected that the normal compilation order would produce the unlisted files, the parallelized compilation would fail.

To summarize, *adding meta-data to the project allow the program development process to go faster*.

3.6 Multiple views of a web document

Some web-sites have so many advertisements and auxiliary links in their layout, that they need an additional version which is well-suited for printing (i.e. only has a single ad). See figure 3.9 on the facing page for an example from ZDNet[45], and figure 3.10 on page 26 for the "Printer-friendly version".

It is interesting to notice that websites realize that they cannot just offer an advertisement ladden service, without allowing for a reasonable paper version. Since their HTML cannot "remove" the ads for printing, they have to offer two different version of each article.

These are two different *views* of the same basic document. ZDNet are expecting their visitors to either see their documents on a modern display unit with high resolution and millions of colors rendered by a reasonably capable browser or print them out on paper. That will probably change within a few years when it becomes common for other devices than just traditional computers to be connected to the Internet. Mobile phones using WAP do not have as much screen real estate (the Nokia 9110 is shown in figure 3.11 on page 26) so such users will probably prefer short and crisp versions of the documents when they use a mobile phone. Nokia is working with 3com to produce a hybrid between a Palm Pilot and a mobile phone.

This tendency might as a side-effect be beneficial for the currently rather overlooked blind computer users (see the Blind Web-ring[46] for more information) . They must normally use

[41]...Borland C++Builder...`http://www.borland.com/bcppbuilder/`

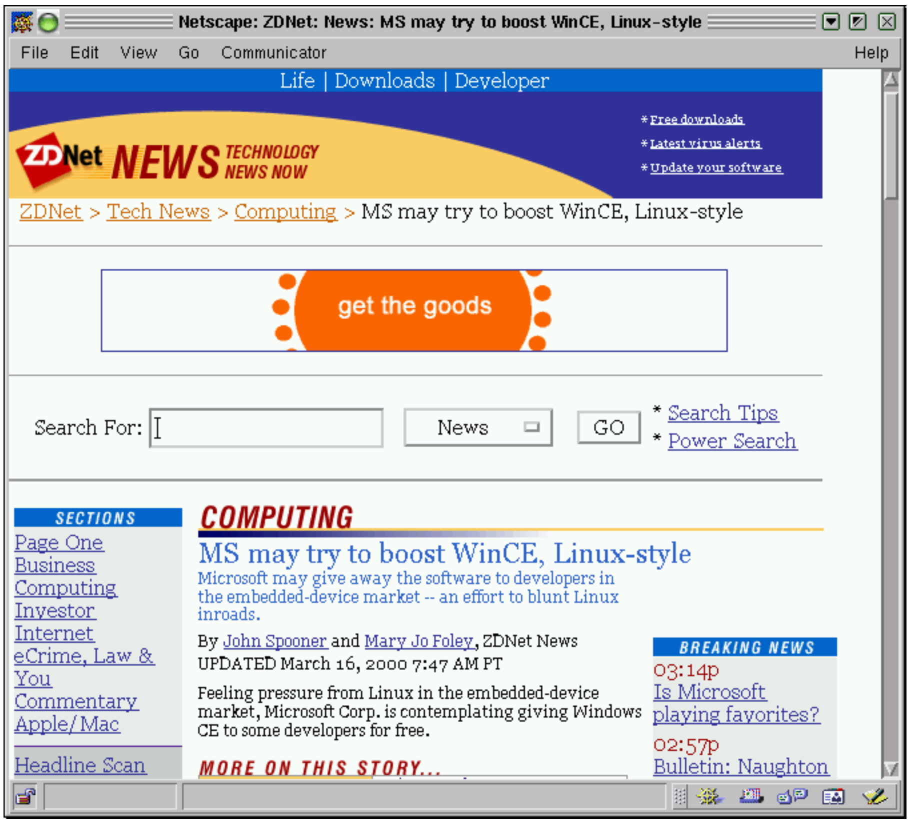
[42]...Microsoft Visual C++...`http://msdn.microsoft.com/visualc/`

[43]...the Visual Age products from IBM...`http://www-4.ibm.com/software/ad/`

[44]...Makefile...`http://www.eng.hawaii.edu/Tutor/Make/`

[45]...for an example from ZDNet...
`http://www.zdnet.com/zdnn/stories/news/0,4586,2468874,00.html?chkpt=zdhpnews01`

[46]...the Blind Web-ring...`http://www.webring.org/cgi-bin/webring?ring=blind&list`


Netscape: ZDNet: News: MS may try to boost WinCE, Linux-style

File Edit View Go Communicator Help

Life | Downloads | Developer

ZDNet NEWS TECHNOLOGY NEWS NOW

* Free downloads
* Latest virus alerts
* Update your software

ZDNet > Tech News > Computing > MS may try to boost WinCE, Linux-style

get the goods

Search For: News GO

* Search Tips
* Power Search

SECTIONS
Page One
Business
Computing
Investor
Internet
eCrime, Law & You
Commentary
Apple/Mac
Headline Scan

COMPUTING

MS may try to boost WinCE, Linux-style

Microsoft may give away the software to developers in the embedded-device market -- an effort to blunt Linux inroads.

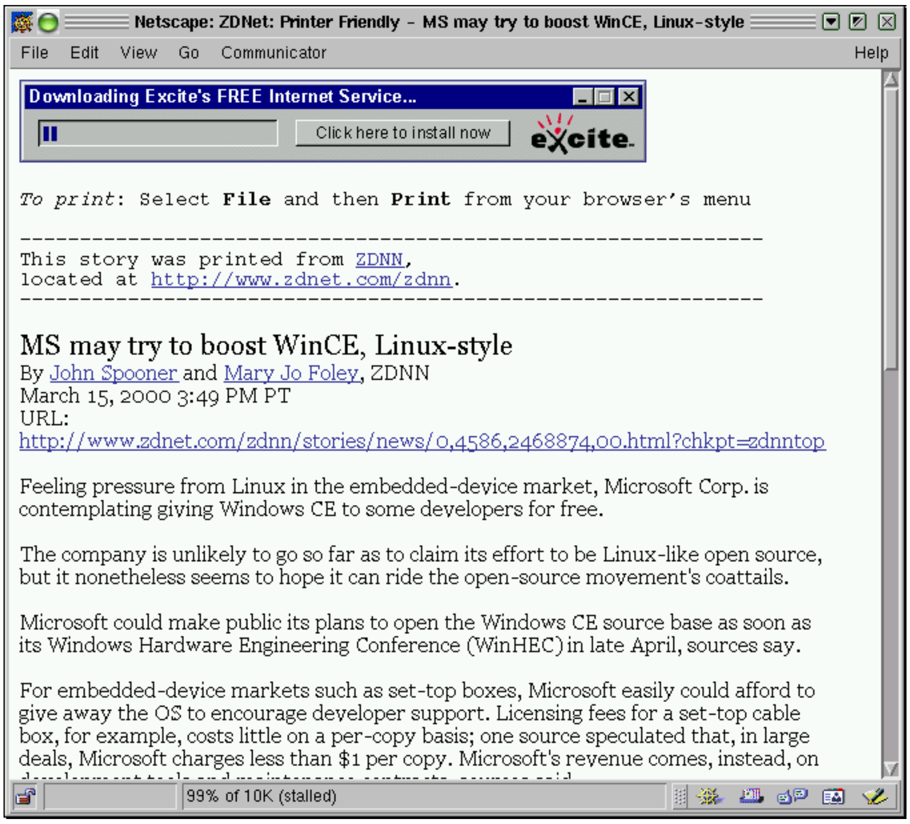
By John Spooner and Mary Jo Foley, ZDNet News
UPDATED March 16, 2000 7:47 AM PT

Feeling pressure from Linux in the embedded-device market, Microsoft Corp. is contemplating giving Windows CE to some developers for free.

MORE ON THIS STORY...

BREAKING NEWS
03:14p
Is Microsoft playing favorites?
02:57p
Bulletin: Naughton

Figure 3.9: The ZDNet presentation of a story – notice how little of the initial screen that is dedicated to the story about “Microsoft may try to boost WinCE, Linux-style”. The “Printer Friendly version” icon is about half way down the page from the start


Netscape: ZDNet: Printer Friendly – MS may try to boost WinCE, Linux-style
File
Edit
View
Go
Communicator
Help
Downloading Excite's FREE Internet Service...
Click here to install now
excite.
99% of 10K (stalled)

To print: Select **File** and then **Print** from your browser's menu

---

This story was printed from ZDNN,
located at http://www.zdnet.com/zdnn.

MS may try to boost WinCE, Linux-style
By John Spooner and Mary Jo Foley, ZDNN
March 15, 2000 3:49 PM PT
URL:
http://www.zdnet.com/zdnn/stories/news/0,4586,2468874,00.html?chkpt=zdnntop

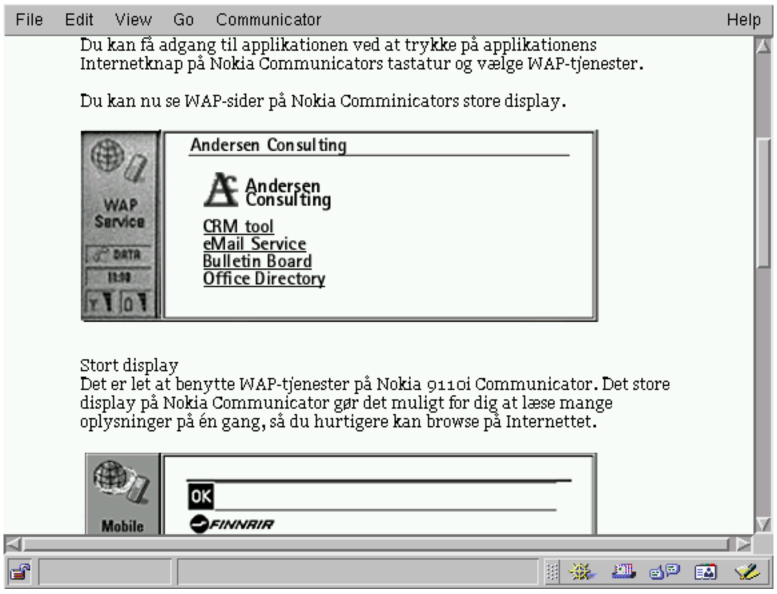
Feeling pressure from Linux in the embedded-device market, Microsoft Corp. is contemplating giving Windows CE to some developers for free.

The company is unlikely to go so far as to claim its effort to be Linux-like open source, but it nonetheless seems to hope it can ride the open-source movement's coattails.

Microsoft could make public its plans to open the Windows CE source base as soon as its Windows Hardware Engineering Conference (WinHEC) in late April, sources say.

For embedded-device markets such as set-top boxes, Microsoft easily could afford to give away the OS to encourage developer support. Licensing fees for a set-top cable box, for example, costs little on a per-copy basis; one source speculated that, in large deals, Microsoft charges less than $1 per copy. Microsoft's revenue comes, instead, on

Figure 3.10: The "Printer-friendly version" of figure 3.9 on the page before



Du kan få adgang til applikationen ved at trykke på applikationens Internetknap på Nokia Communicators tastatur og vælge WAP-tjenester.

Du kan nu se WAP-sider på Nokia Comminicators store display.

Andersen Consulting
CRM tool
eMail Service
Bulletin Board
Office Directory

Stort display
Det er let at benytte WAP-tjenester på Nokia 9110i Communicator. Det store display på Nokia Communicator gør det muligt for dig at læse mange oplysninger på én gang, så du hurtigere kan browse på Internettet.

Figure 3.11: Browsing the Internet on the large [sic] display of a Nokia 9110

text-to-braille software or a "screen reader" in order to use computers, neither of which work well with web pages loaded with graphics, but when web authors must write for many kinds of media instead of just a particular version of a given browser, this will also be beneficial to these users since it will be easy to write a renderer for their preferred format. Of course, it is possible to write ordinary HTML in a way that is usable by these users, but that is rare these days.

Both Jakob Nielsen[47] and Statens Information [in Danish][48] have good guidelines for making pages with web content accessible to people with disabilities based on the work done by the W3C Web Accessory Initiative[49]. Pages written with these guidelines in mind will work – not only for disabled readers – but should be ready for the technological advances in the near future.

3.7 SGML/XML: Generic formats for storing data and meta-data

The problem of representing content in a generic way is not new at all, and was solved for the printing industry in the 1970'ies by the design of the Standard Generalized Markup Language (SGML) which is widely used. HTML and XML are both SGML-dialects abeit with different capabilities. Chapter 5 on page 49 talks a lot more about this.

For now it is sufficient to say that it is possible to store both data and meta-data in a convenient form in a SGML or XML file, but that these files must be rendered into the formats expected by the users, like HTML, PDF or what else tomorrow may bring of new, exiting formats. Since SGML describes the *content* and not the layout, a new output format is "just" a matter of creating a renderer for any new format and add it to the collection.

This has been recognized by almost all of the "Documentation Projects" (The Linux Documentation Project[50], The FreeBSD Documentation Project[51] and others) accompanying the various free operating systems available on the Internet.

Chapter A.5 on page 124 talks about rendering documents to a given format on demand (for example in a web browser).

3.8 The user should use familiar tools to publish documents

Editing SGML manually is hard and tedious - this is one thing that the folks in the `comp.text.sgml`[52] and `comp.text.xml`[53] newsgroups agree upon, and quite sophisticated tools are essential to make authoring directly in SGML viable.

Very few tools exist, and the good ones are quite expensive. The general consensus is that the best OpenSource tool is the Emacs editor with the PSGML package (see A.5 on page 124), and that this

[47]...Jakob Nielsen...`http://www.useit.com/alertbox/990613.html`
[48]...Statens Information [in Danish]...`http://www.si.dk/netsteder/publ/tilgaeng/index.html`
[49]...the work done by the W3C Web Accessory Initiative...`http://www.w3.org/TR/WAI-WEBCONTENT/`
[50]...The Linux Documentation Project...`http://www.linuxdoc.org/`
[51]...The FreeBSD Documentation Project...`http://www.freebsd.org/docproj/docproj.html`
[52]...`comp.text.sgml`...`news:comp.text.sgml`
[53]...`comp.text.xml`...`news:comp.text.xml`

tool is primarily suited for programmers and other people who are well acquainted with SGML and XML.

Computers should *help* you do your work, and users are generally most productive with the tools they know. Why should an author use SGML with an editor she dislikes, if almost the same result can be achieved by using a word processor she is familiar with?

Automated conversions from one document format to another is a non-trivial matter, in particular when the target format contains more meta-data than the original. It is therefore important that the author is fully aware of the underlying processes and use only those constructions that the conversion program understands. This is most likely given as a set of macros that must be used, as well as a verification program which the author can use at any time to check whether her document is complying to the DTD. These rules must be strictly enforced during the writing process in order for the author to be as productive as possible.

In addition to this, the publishing process should be as simple as possible. For comparison is it quite a complicated task to print a document. Numerous variables regarding choice of printer driver, location of physical device, transporting the bytes across the cables, and communicating with the printer to ensure that all is well during printing. This complicated task is today as simple as pressing a “Print” button and checking a few things in a dialogue box. (This is the same approach Adobe uses in their Acrobat software – they capture the output to a printer and convert it directly to PDF).

Web publishing is not any more complicated or difficult than that, and this should be reflected in the working procedure. My conclusion is:

Publishing a document to the web should be as easy as printing a document.

This conclusion, combined with my experiences with the Yggdrasil system, have been one of several design goal in the Cactus system which is described in section 10 on page 87, and which has been fulfilled.

3.9 Small search engines for a web site

Remember something about this

Chapter 4

Sample websites and projects

slashdot.org (examine code - ask developer), Politiken, DynaWeb (SGI - oooold check on fyn, when was it first developed?), valueclick (banners - ask Ask for full story. What are they running), php3 documentation online. photo.net (reread). Sun's docs.sun.com (DocBook SGML)
Talk about standard search engines on web pages.
http://www.useit.com/alertbox/991017.html

4.1 slashdot.org - high volume information site for nerds

SlashDot[1] was started in September 1997 with the slogan "News for nerds - Stuff that matters", and do so by providing dynamically generated web pages with a lot of articles categorized with an icon, where a short summary is shown along with links to the full story, several references to the orignal sources, the authors email address, plus an overview of the number of comments in the discussion forum. See figure 4.1 on the next page for the top of the start page at March 14 2000.

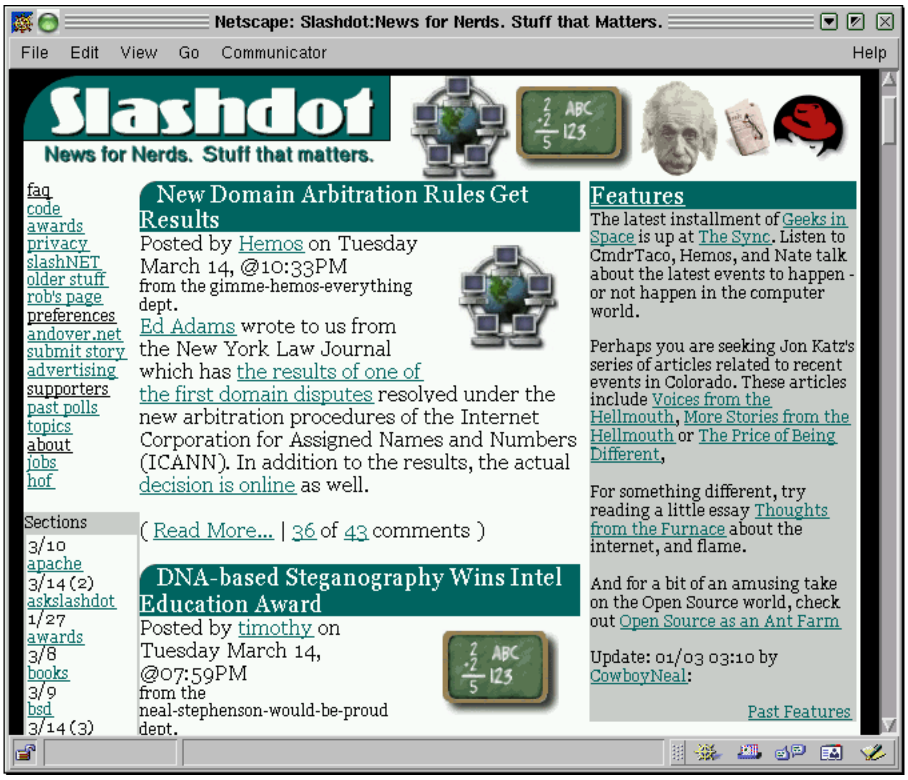
When the "Read more" link to a given article is followed, a page is shown with the full text of the posting, followed by a long list of comments submitted by readers (see figure 4.2 on page 31), and a login box.

Unregistered readers are allowed to post under the generic name of "Anonymous Coward", where registration allows readers to select their own handle, full name, home page link, page layout and what kind of articles they want to see.

After logging in, the "Welcome Back" screen in figure 4.3 on page 32 is shown. Note that the layout has changed, and that I have not moderated anything. Slashdot uses moderation where everybody can grade other posters comments which is then available to users for their selection of articles. Personally I have chosen a minimum level of two, meaning that I never see anything from Anonymous Cowards (which improves the quality immensely). Meta moderation is intended to control the quality of the moderation, and is a volunteer action.

```
Slashdot Stats
```

[1]...SlashDot...`http://slashdot.org`


Netscape: Slashdot:News for Nerds. Stuff that Matters.
File Edit View Go Communicator Help
Slashdot
News for Nerds. Stuff that matters.

faq
code
awards
privacy
slashNET
older stuff
rob's page
preferences
andover.net
submit story
advertising
supporters
past polls
topics
about
jobs
hof

Sections
3/10
apache
3/14 (2)
askslashdot
1/27
awards
3/8
books
3/9
bsd
3/14 (3)

New Domain Arbitration Rules Get Results

Posted by Hemos on Tuesday March 14, @10:33PM
from the gimme-hemos-everything dept.
Ed Adams wrote to us from the New York Law Journal which has the results of one of the first domain disputes resolved under the new arbitration procedures of the Internet Corporation for Assigned Names and Numbers (ICANN). In addition to the results, the actual decision is online as well.

(Read More... | 36 of 43 comments)

DNA-based Steganography Wins Intel Education Award

Posted by timothy on Tuesday March 14, @07:59PM
from the neal-stephenson-would-be-proud dept.

Features

The latest installment of Geeks in Space is up at The Sync. Listen to CmdrTaco, Hemos, and Nate talk about the latest events to happen - or not happen in the computer world.

Perhaps you are seeking Jon Katz's series of articles related to recent events in Colorado. These articles include Voices from the Hellmouth, More Stories from the Hellmouth or The Price of Being Different.

For something different, try reading a little essay Thoughts from the Furnace about the internet, and flame.

And for a bit of an amusing take on the Open Source world, check out Open Source as an Ant Farm

Update: 01/03 03:10 by CowboyNeal:

Past Features

Figure 4.1: The Slashdot start page

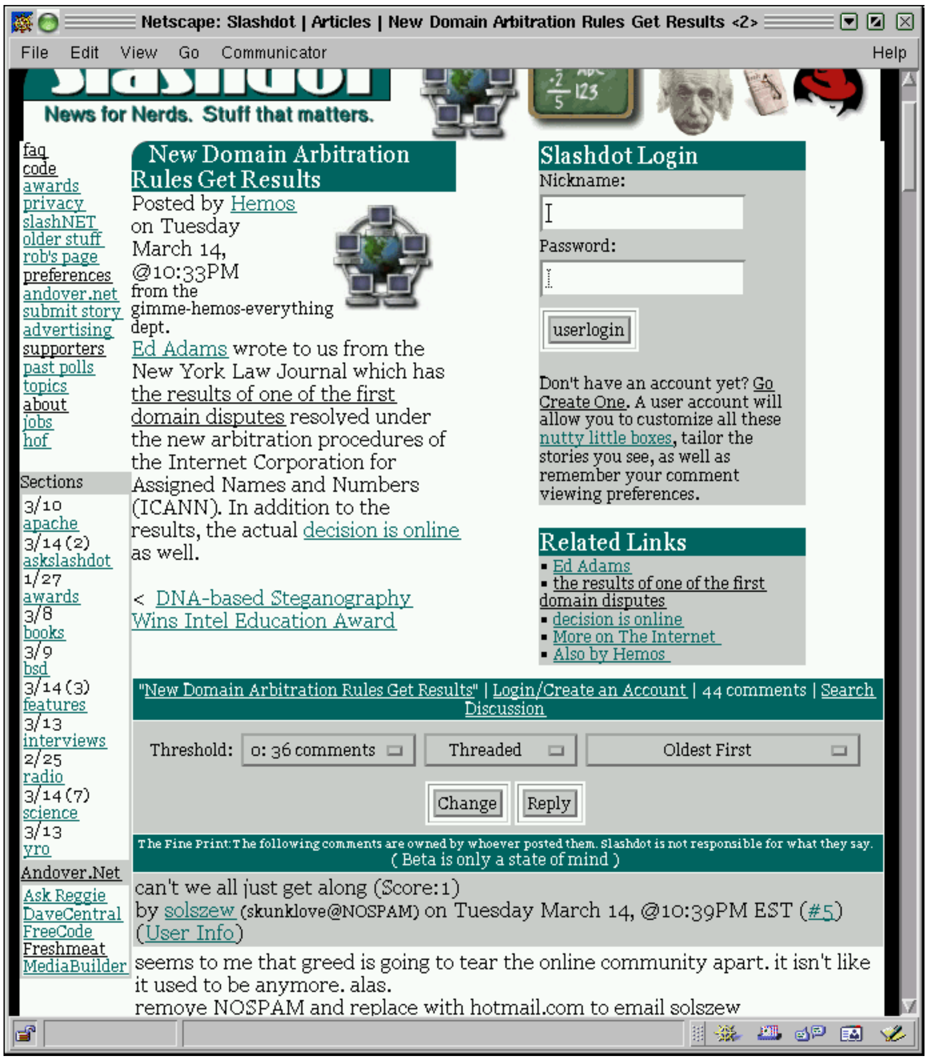
```
date: 2:37am
uptime: 68 days, 5:56, 2 users,
load average: 0.18, 0.19, 0.18
processes: 65
yesterday: 209994
today: 1
ever: 278137148
```

Slashdot provide the “magnet content” Greenspun talks about. Stories are carefully selected, and the user may even select just a few categories in her personal preferences.

All pages are dynamically generated. The discussion forums for any story is active for a reasonable time, where anybody can comment on any other comment, and then “frozen” when the story is archived. If a user at a later date wishes to see the story with the discussion, it is retreived from the backup database.

The “Ask Slashdot” feature is a direct consequence of the fact that Slashdot is targeting power computer users. By allowing carefully selected questions to a highly visible spot, the original inquirer is usually able to get a very qualified answer. The question along with all the answers are stored in the same way as a story.

Due to the prominent position of new postings up front, many readers follow the given links causing the so-called `Slashdot effect`


Netscape: Slashdot | Articles | New Domain Arbitration Rules Get Results <2>
File
Edit
View
Go
Communicator
Help
News for Nerds. Stuff that matters.

faq
code
awards
privacy
slashNET
older stuff
rob's page
preferences
andover.net
submit story
advertising
supporters
past polls
topics
about
jobs
hof

Sections
3/10
apache
3/14 (2)
askslashdot
1/27
awards
3/8
books
3/9
bsd
3/14 (3)
features
3/13
interviews
2/25
radio
3/14 (7)
science
3/13
yro

Andover.Net
Ask Reggie
DaveCentral
FreeCode
Freshmeat
MediaBuilder

New Domain Arbitration Rules Get Results

Posted by Hemos
on Tuesday
March 14,
@10:33PM
from the
gimme-hemos-everything
dept.

Ed Adams wrote to us from the New York Law Journal which has the results of one of the first domain disputes resolved under the new arbitration procedures of the Internet Corporation for Assigned Names and Numbers (ICANN). In addition to the results, the actual decision is online as well.

< DNA-based Steganography Wins Intel Education Award

Slashdot Login

Nickname:

Password:

userlogin

Don't have an account yet? Go Create One. A user account will allow you to customize all these nutty little boxes, tailor the stories you see, as well as remember your comment viewing preferences.

Related Links

- Ed Adams
- the results of one of the first domain disputes
- decision is online
- More on The Internet
- Also by Hemos

"New Domain Arbitration Rules Get Results" | Login/Create an Account | 44 comments | Search Discussion

Threshold: 0: 36 comments | Threaded | Oldest First

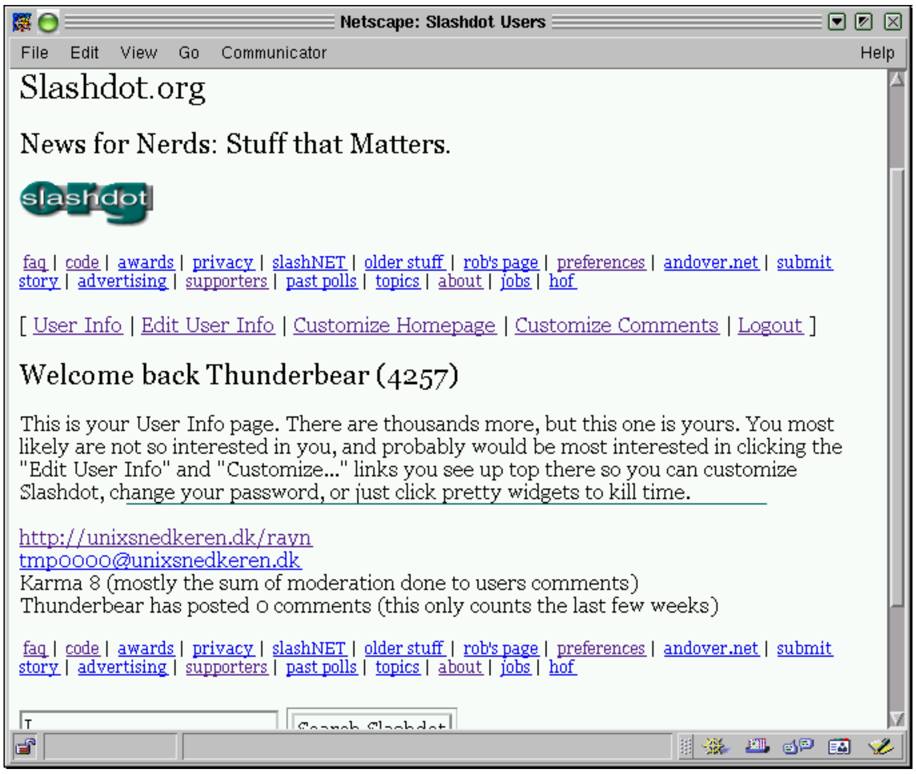
Change | Reply

The Fine Print: The following comments are owned by whoever posted them. Slashdot is not responsible for what they say.
(Beta is only a state of mind)

can't we all just get along (Score:1)
by solszew (skunklove@NOSPAM) on Tuesday March 14, @10:39PM EST (#5)
(User Info)

seems to me that greed is going to tear the online community apart. it isn't like it used to be anymore. alas.
remove NOSPAM and replace with hotmail.com to email solszew

Figure 4.2: A Slashdot article with the top of the comments



Slashdot.org

News for Nerds: Stuff that Matters.

faq | code | awards | privacy | slashNET | older stuff | rob's page | preferences | andover.net | submit story | advertising | supporters | past polls | topics | about | jobs | hof

[User Info | Edit User Info | Customize Homepage | Customize Comments | Logout]

Welcome back Thunderbear (4257)

This is your User Info page. There are thousands more, but this one is yours. You most likely are not so interested in you, and probably would be most interested in clicking the "Edit User Info" and "Customize..." links you see up top there so you can customize Slashdot, change your password, or just click pretty widgets to kill time.

http://unixsnedkeren.dk/ravn
tmp0000@unixsnedkeren.dk
Karma 8 (mostly the sum of moderation done to users comments)
Thunderbear has posted 0 comments (this only counts the last few weeks)

faq | code | awards | privacy | slashNET | older stuff | rob's page | preferences | andover.net | submit story | advertising | supporters | past polls | topics | about | jobs | hof

Figure 4.3: My personal page at Slashdot

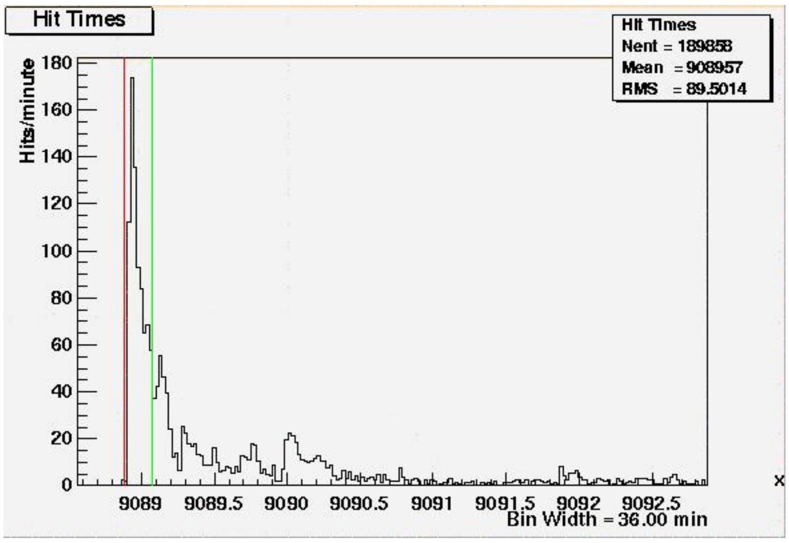


Figure 4.4: A webserver log showing the Slashdot effect – the red line at approximately 9050 shows when the announcement was made on Slashdot

4.2 Valueclick.com

ValueClick is one of several banner-ad sellers on the Internet. Banner ads are animated images which advertise for a given website. The advertiser pay ValueClick for a certain amount of "*click-throughs*" (users who actually click the image to go to the advertised site) as opposed to the traditional number of exposures. The former is harder to track than the latter.

ValueClick claims on their web that they send out 40 million banner ads every day – that is 460 ads a second – so it is vital for business that the counters for each advertiser is accurately updated. From private email I have learned that their original NT-cluster based solutions were not good enough for several reasons, most notably the webservers having to wait for the browsers to accept every byte sent to them, before they could service the next. By handling this and other similar problems, a single Linux based Pentium machine could outperform the rest of the server park.

Today ValueClick is based on a large number of FreeBSD machines which use MySQL as their database server, and apparently works very well.

4.3 Amazon - an Internet bookstore

The Amazon Internet bookstore[2] pioneered the virtual Internet bookstore, where the potential buyers use their browser to see the virtual inventory of literally millions of books, and order their selections. These are processed by Amazon and forwarded to subcontracters who deliver the purchased goods by mail to the buyers. The concept has since been copied by several other bookstores.

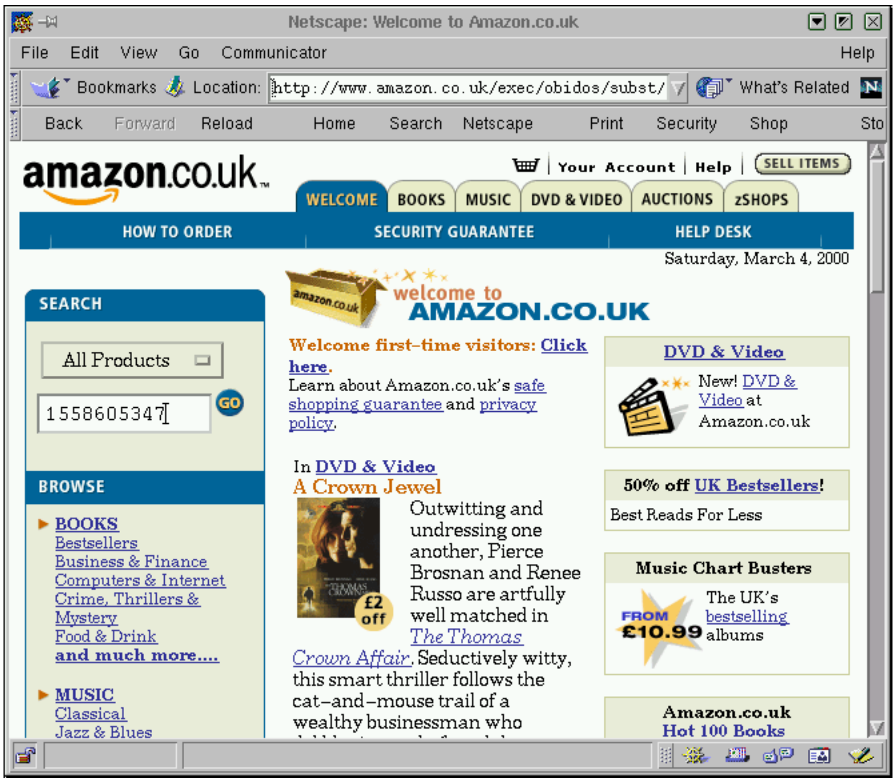
The Amazon frontpage is shown in figure 4.5 on the following page. Note the high amount of potentially interesting links, and the prominently placed search box. Due to the limited domain of books, the search engine can make assumptions about what the user is requesting information about. In this case a ten-digit number was entered which happened to be an ISBN-number, which in most other search engines should have been entered as "ISBN-1-55860-5347", and the page for the corresponding book was returned (see figure 4.6 on page 35).

This page shows the essential information like author, price, and the number of pages, but also an image of the cover, the rank on the Amazon sales list, and a direct link to purchase the book (Amazon remembers your credit card information, allowing for their patented "One-click-purchase" technology). What makes the Amazon web page truly better that a printed catalogue is that the users are encouraged to submit feedback, and the information Amazon collects from online sales to provide information to potential customers about other books that might interest them, because they interested other customers which bought the book they are looking at now.

These features are distinct for a page for a purchasable item at Amazon:

- **Average customer rating** - this is the essence of all the reviewers opinions, on a scale from 0 to 5.
- **Reviews** - readers take the time to create comprehensive reviews of the books, all of which Amazon then places prominently on the page, with due credit to the individual author. Recently a "Was this review helpful to you?" button was added to each review, and the result of

[2]...Amazon Internet bookstore...`http://www.amazon.co.uk`


Netscape: Welcome to Amazon.co.uk
File Edit View Go Communicator Help
Bookmarks Location: http://www.amazon.co.uk/exec/obidos/subst/ What's Related
Back Forward Reload Home Search Netscape Print Security Shop Sto
amazon.co.uk
Your Account | Help | SELL ITEMS
WELCOME BOOKS MUSIC DVD & VIDEO AUCTIONS zSHOPS
HOW TO ORDER SECURITY GUARANTEE HELP DESK
SEARCH
All Products
1558605347
GO
BROWSE
BOOKS
Bestsellers
Business & Finance
Computers & Internet
Crime, Thrillers & Mystery
Food & Drink
and much more....
MUSIC
Classical
Jazz & Blues
welcome to AMAZON.CO.UK
£2 off
FROM £10.99

Saturday, March 4, 2000

Welcome first-time visitors: Click here.
Learn about Amazon.co.uk's safe shopping guarantee and privacy policy.

In DVD & Video
A Crown Jewel
Outwitting and undressing one another, Pierce Brosnan and Renee Russo are artfully well matched in *The Thomas Crown Affair*. Seductively witty, this smart thriller follows the cat–and–mouse trail of a wealthy businessman who

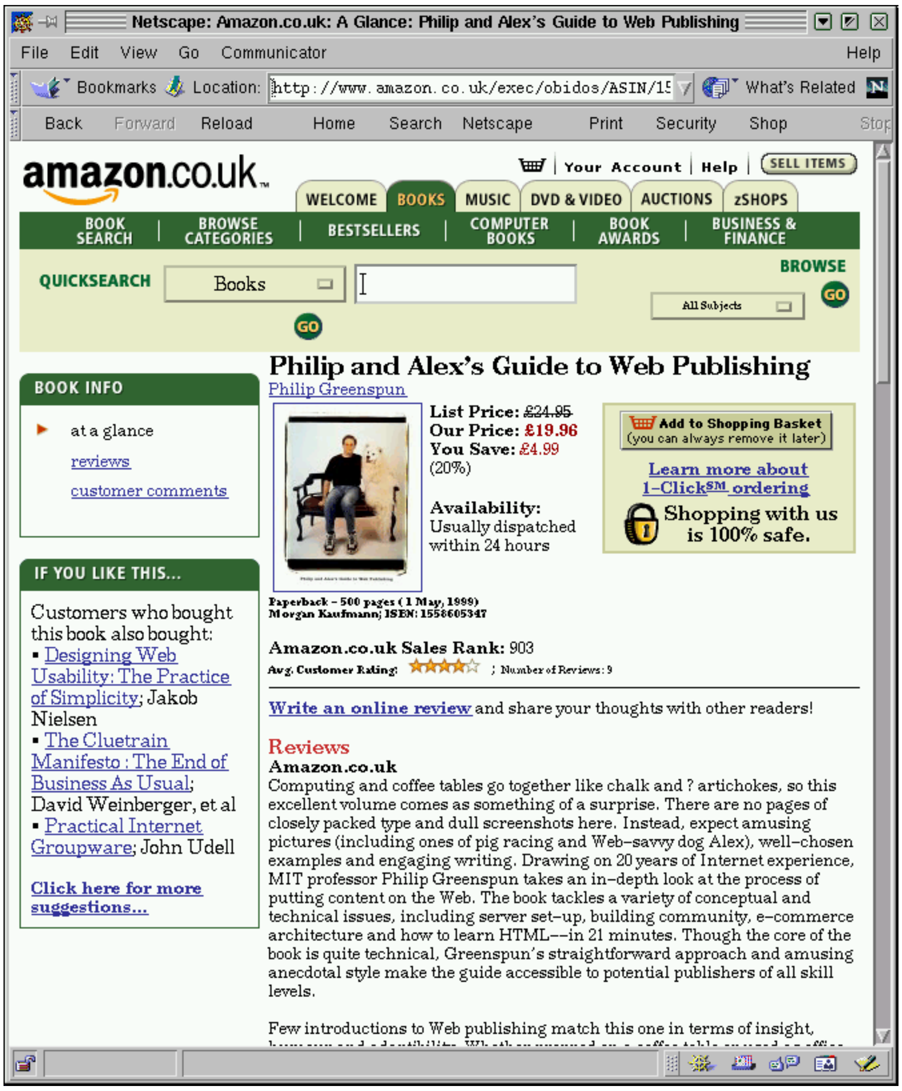
DVD & Video
New! DVD & Video at Amazon.co.uk

50% off UK Bestsellers!
Best Reads For Less

Music Chart Busters
The UK's bestselling albums

Amazon.co.uk
Hot 100 Books

Figure 4.5: The frontpage of Amazon.co.uk. An ISBN-number has been entered in the very prominent search box


Netscape: Amazon.co.uk: A Glance: Philip and Alex's Guide to Web Publishing
File Edit View Go Communicator Help
Bookmarks Location: http://www.amazon.co.uk/exec/obidos/ASIN/15 What's Related
Back Forward Reload Home Search Netscape Print Security Shop Stop
amazon.co.uk
Your Account | Help | SELL ITEMS
WELCOME BOOKS MUSIC DVD & VIDEO AUCTIONS zSHOPS
BOOK SEARCH | BROWSE CATEGORIES | BESTSELLERS | COMPUTER BOOKS | BOOK AWARDS | BUSINESS & FINANCE
QUICKSEARCH Books GO
BROWSE All Subjects GO

Philip and Alex's Guide to Web Publishing
Philip Greenspun

List Price: ~~£24.95~~
Our Price: £19.96
You Save: £4.99 (20%)

Availability:
Usually dispatched within 24 hours

Add to Shopping Basket
(you can always remove it later)

Learn more about 1-Click^{SM} ordering

Shopping with us is 100% safe.

Paperback – 500 pages (1 May, 1999)
Morgan Kaufmann; ISBN: 1558605347

Amazon.co.uk Sales Rank: 903
Avg. Customer Rating: ★★★★☆ ; Number of Reviews: 9

Write an online review and share your thoughts with other readers!

BOOK INFO
- at a glance
- reviews
- customer comments

IF YOU LIKE THIS...

Customers who bought this book also bought:
- Designing Web Usability: The Practice of Simplicity; Jakob Nielsen
- The Cluetrain Manifesto : The End of Business As Usual; David Weinberger, et al
- Practical Internet Groupware; John Udell

Click here for more suggestions...

Reviews
Amazon.co.uk
Computing and coffee tables go together like chalk and ? artichokes, so this excellent volume comes as something of a surprise. There are no pages of closely packed type and dull screenshots here. Instead, expect amusing pictures (including ones of pig racing and Web-savvy dog Alex), well-chosen examples and engaging writing. Drawing on 20 years of Internet experience, MIT professor Philip Greenspun takes an in-depth look at the process of putting content on the Web. The book tackles a variety of conceptual and technical issues, including server set-up, building community, e-commerce architecture and how to learn HTML--in 21 minutes. Though the core of the book is quite technical, Greenspun's straightforward approach and amusing anecdotal style make the guide accessible to potential publishers of all skill levels.

Few introductions to Web publishing match this one in terms of insight,

Figure 4.6: The result of the search in figure 4.5 on the facing page

each poll is shown along with the review, allowing a customer to rapidly determine how to rate the review.

- **Similar books** - if a previous buyer bought other books along with this one, they might be interesting to this person too. Links are presented to those books, as well as to the general categories this book was placed in by the Amazon librarian.
- **Expected content** - Amazon can be expected to have *any* English book in their database. Users expect to be able to find a given book in the Amazon database, and usually do. Personally I have never looked in vain at Amazon for English books.

Amazon explicitely invites those who read the page to review the book if they have read it, with special treatment for the author and the publisher. *As Greenspun observes, then this is the most efficient material*

At the time of writing, this book has 9 reviews at *amazon.co.uk*, but 198 at the mother site at *amazon.com*. The corresponding HTML-pages are 22 kb and 550 kb respectively, which gives an an impressive *Greenspun factor* (setting the UK-version to be 100% *??-provider*) of 25.

Amazon uses very aggressive marketing, which has caused it to be a well-known brand name in just a few years. By giving money to those who link to their pages, as well as pay search engines to have links to the Amazon site show up at the top in just about every search, they have managed to be just about the only Internet bookstore known to most people.

Conclusion: The Amazon family of sites are very good, and do what they can to improve with the help of visitors. Amazon have been able to keep their leader status in the virtual bookshop niche, by living up to the customers expectations. Hopefully, they will make money one day.

4.4 Bang and Olufsen

The Bang and Olufsen[3] web-site was previously very unusual in the way that they offered random navigation. Their front page contained very few links, and some product images which could be clicked and lead to a large set of pages (see figure 4.7 on the next page, with three random clickable images at the top. These were chosen randomly from the large set of such pages (600 or more) and put at the top. This has been highly efficient in keeping people wandering around and reading their pages, giving these people an unusual experience.

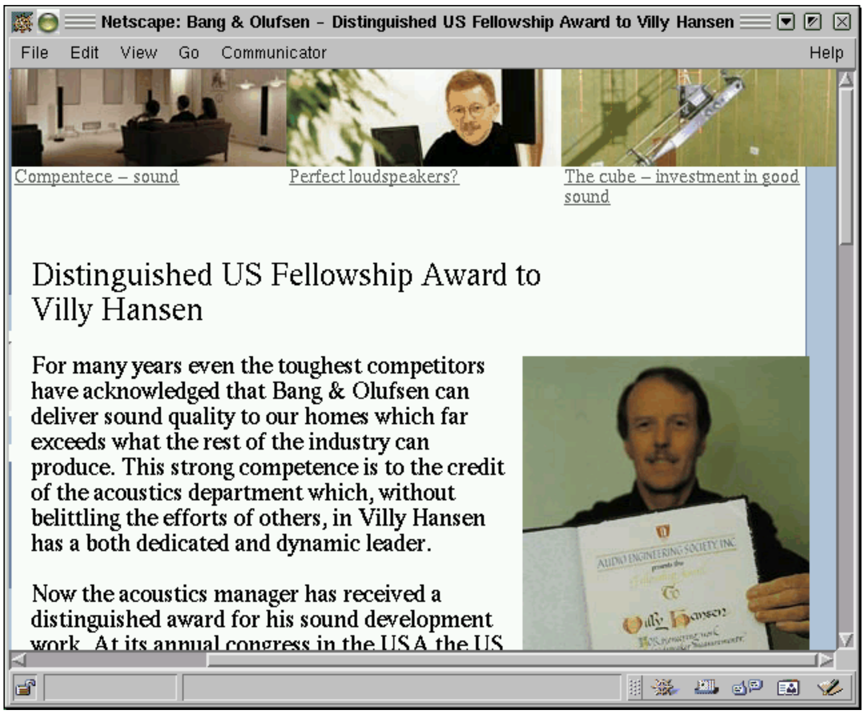
The underlying database is both used for storing the many pages, but also for tracking the users as they wander around the site.

4.5 Deja - where did Usenet go?

Write about this if there is time for it

Deja Usenet archive[4]

[3]...Bang and Olufsen...`http://www.bang-olufsen.com/`
[4]...Deja Usenet archive...`http://www.deja.com/usenet`


Netscape: Bang & Olufsen – Distinguished US Fellowship Award to Villy Hansen

File Edit View Go Communicator Help

Compentece – sound
Perfect loudspeakers?
The cube – investment in good sound

Distinguished US Fellowship Award to Villy Hansen

For many years even the toughest competitors have acknowledged that Bang & Olufsen can deliver sound quality to our homes which far exceeds what the rest of the industry can produce. This strong competence is to the credit of the acoustics department which, without belittling the efforts of others, in Villy Hansen has a both dedicated and dynamic leader.

Now the acoustics manager has received a distinguished award for his sound development work. At its annual congress in the USA the US

Figure 4.7: The Bang and Olufsen pages – notice the three randomly selected images at the top

4.6 www.krak.dk

Making maps on demand was one of the earliest applications for the web. The $\boxed{\textit{web map application was started}}$ was discussed in section 3.1.1 on page 10, and has since been followed by $\boxed{\textit{yahoo map, whoelse.}}$

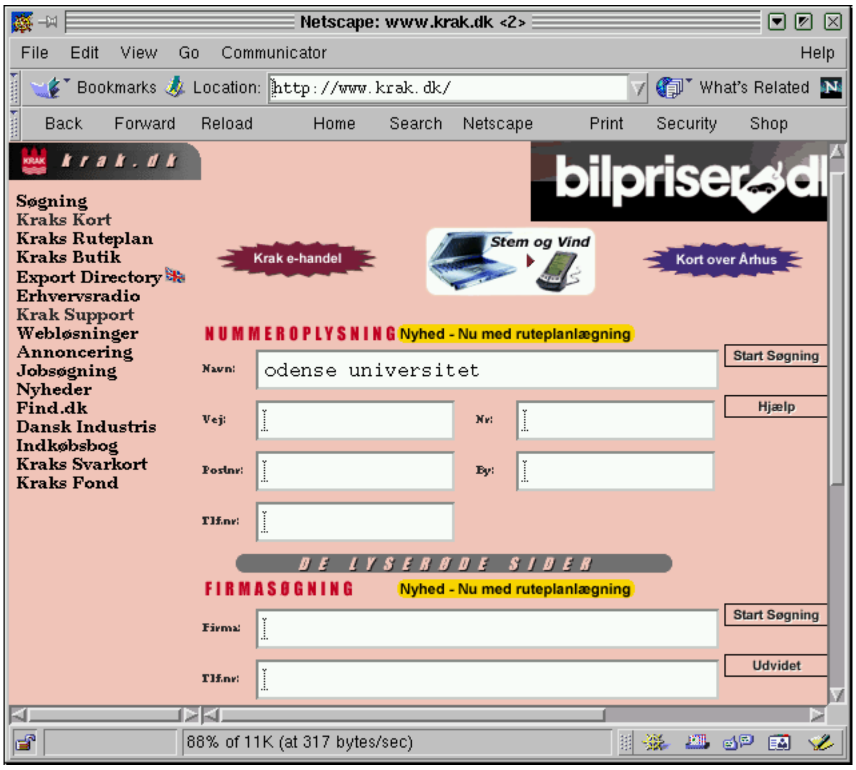
In Denmark, the well renowned Danish map manufacturer Krak A/S[5] opened their Internet phone directory and web map generation service in direct competition with numerous other directory services. Their advantage was up-to-date data from the databases of Tele Danmark combined with an ability to generate a map for a given address, whether it was entered directly or being a byproduct of a search for something else. This has proven to be so powerful a combination of services that all other phone directory services has been left far behind.

The website has been steadily improved with new facilities and cross-references, as well as more precise data (In $\boxed{??}$ 1999 it could not locate "Herlev Hovedgade 205" on the map, so the whole of the road was inked on the map. This is corrected today). The start page is shown in figure 4.8 on the next page, and is seperated in two parts; namely white pages (phone number based), and pink pages (company based).

The fields in the white page section are Name, Road, House number, Zip code, City and Phone number. The fields in the pink page section are Company and Phone number. The form has been filled out with a search request for "Odense Universitet", and the results are shown in figure 4.9 on the following page.

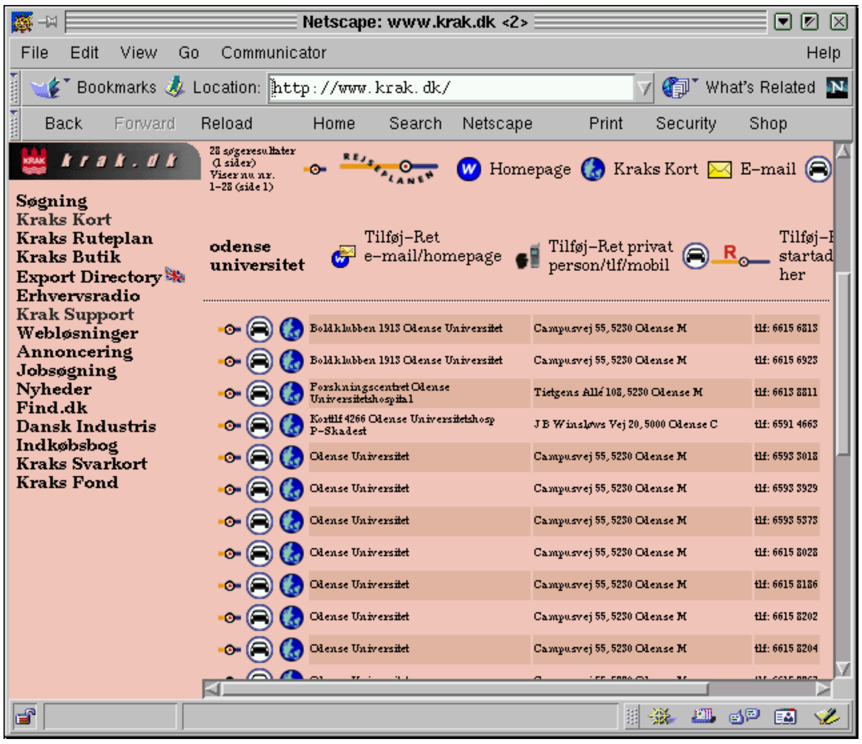
Each line containing an answer may have icons referencing to facilities regarding the location of

[5]. . . Krak A/S. . . `http://www.krak.dk`



Navn: odense universitet

Figure 4.8: Welcome page (and search form) for www.krak.dk



28 søgeresultater (1 sider) Viser nu nr. 1–28 (side 1)

odense universitet

Boldklubben 1913 Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 6813
Boldklubben 1913 Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 6923
Forskningscentret Odense Universitetshospital	Tietgens Allé 108, 5230 Odense M	tlf: 6613 8811
Korttlf 4266 Odense Universitetshosp P–Skadest	J B Winsløws Vej 20, 5000 Odense C	tlf: 6591 4663
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6593 3018
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6593 3929
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6593 5373
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 8028
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 8186
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 8202
Odense Universitet	Campusvej 55, 5230 Odense M	tlf: 6615 8204

Figure 4.9: Result of searching for “Odense University”

the answer. These answers have pointers to Rejseplanen, the Route Planner, and a map reference. Additionally pointers to a web page, and an email address could have been provided by the users. Clicking the map icon of the first line actually referring to Odense University brings us to figure 4.10.

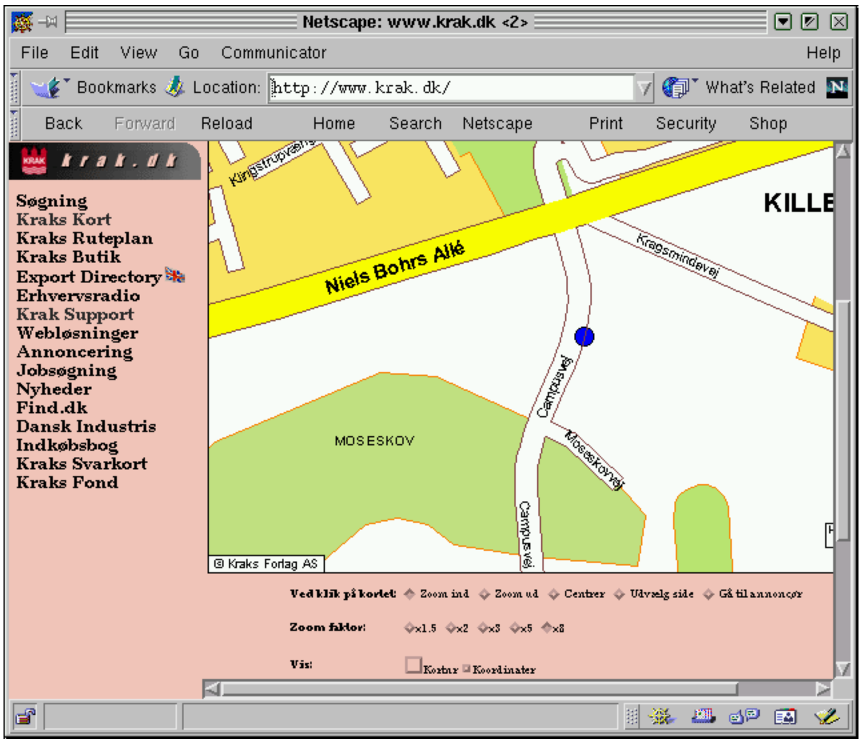


Figure 4.10: Following the map icon for the first "Odense University" reference

The blue dot usually indicates the location of the address but in this case the map is not 100% correct (Moseskovvejen does not go all the way south-east to the parking lot, and the University is located 500 meters further to the south). When a map is display, the navigation area below the image allows for zooming and moving the contents of the shown area. By selecting "Zoom out" and "x8" and clicking on the blue dot a new map is shown (see figure 4.11 on the following page), where it is evident that the database show a great deal of detail. Roads, residential areas, streams, train stations, the motor way and green areas are shown. These maps provide a level of information corresponding to Krak's printed maps.

Now go back to the list of results from the search for "Odense University". The Car-icon gives a route to the listed location, where you must fill in the source yourself. Figure 4.12 on the following page show the form filled in with my own address and the address of the university, and "Route on map" (Rute på kort) gives a visual route between the two locations, along with an estimate of the distance and time it will take.

This map is shown in figure 4.13 on page 41, and is perfectly adequate for a person driving a car. For cyclists it is often a good idea to look for short-cuts, if you are well known in the area.

If you need step-by-step directions, Krak can provide that too. Select the "$\boxed{???}$" button and print out the directions. Careful studies of these directions can give an idea of the network grid that the route planner use for finding the shortest road.

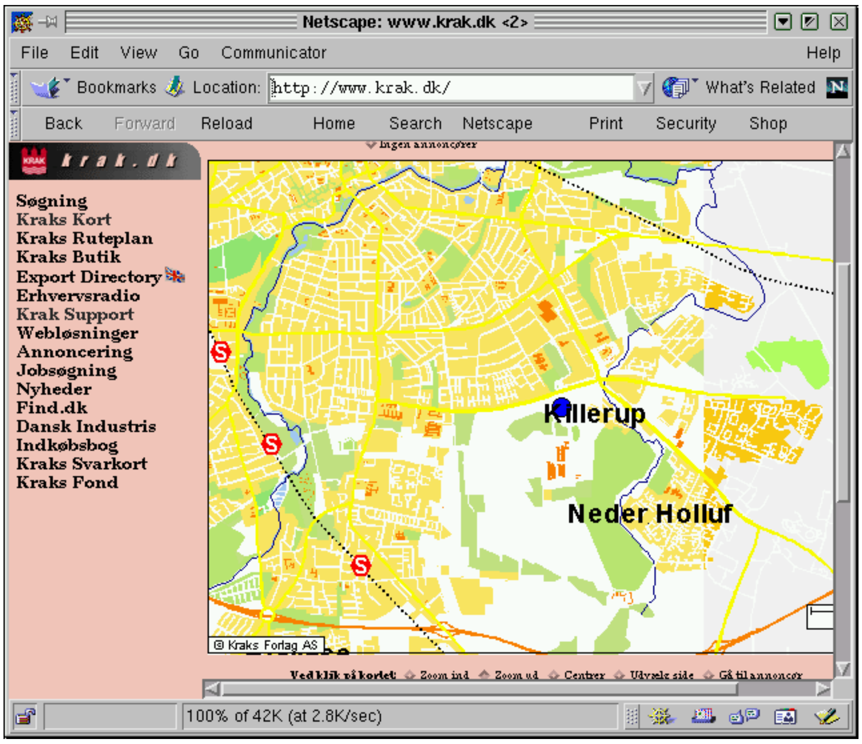
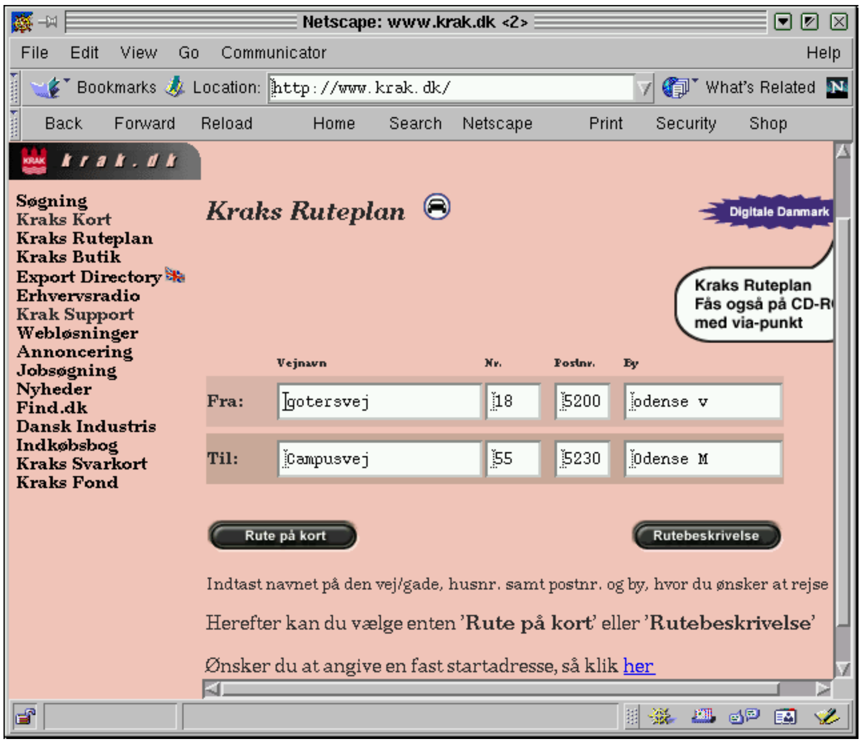

Netscape: www.krak.dk <2>
File Edit View Go Communicator Help
Bookmarks Location: http://www.krak.dk/ What's Related
Back Forward Reload Home Search Netscape Print Security Shop
Ingen annoncører
krak.dk
Søgning
Kraks Kort
Kraks Ruteplan
Kraks Butik
Export Directory
Erhvervsradio
Krak Support
Webløsninger
Annoncering
Jobsøgning
Nyheder
Find.dk
Dansk Industris
Indkøbsbog
Kraks Svarkort
Kraks Fond
Killerup
Neder Holluf
© Kraks Forlag AS
Ved klik på kortet: Zoom ind Zoom ud Centrer Udvælg side Gå til annoncør
100% of 42K (at 2.8K/sec)

Figure 4.11: The map in figure 4.10 on the page before zoomed out with a factor 8



Kraks Ruteplan

	Vejnavn	Nr.	Postnr.	By
Fra:	goter svej	18	5200	odense v
Til:	Campusvej	55	5230	Odense M

Indtast navnet på den vej/gade, husnr. samt postnr. og by, hvor du ønsker at rejse

Herefter kan du vælge enten 'Rute på kort' eller 'Rutebeskrivelse'

Ønsker du at angive en fast startadresse, så klik her

Figure 4.12: The entry box where the starting and ending point is entered

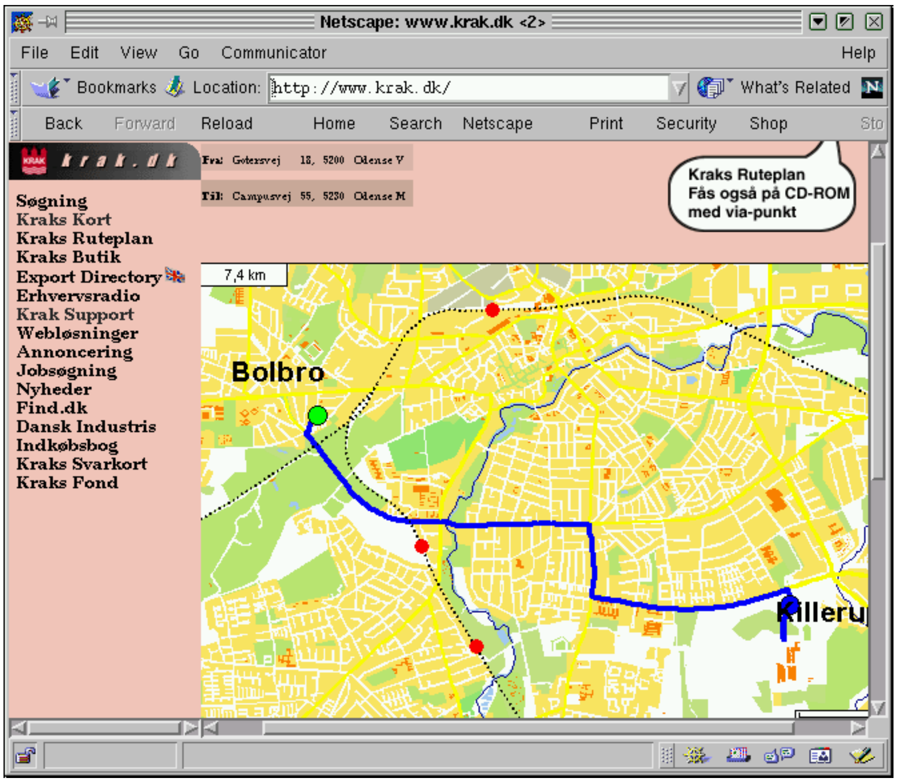


Figure 4.13: The route from my home to the University

4.7 The Journey Planner - www.rejseplanen.dk

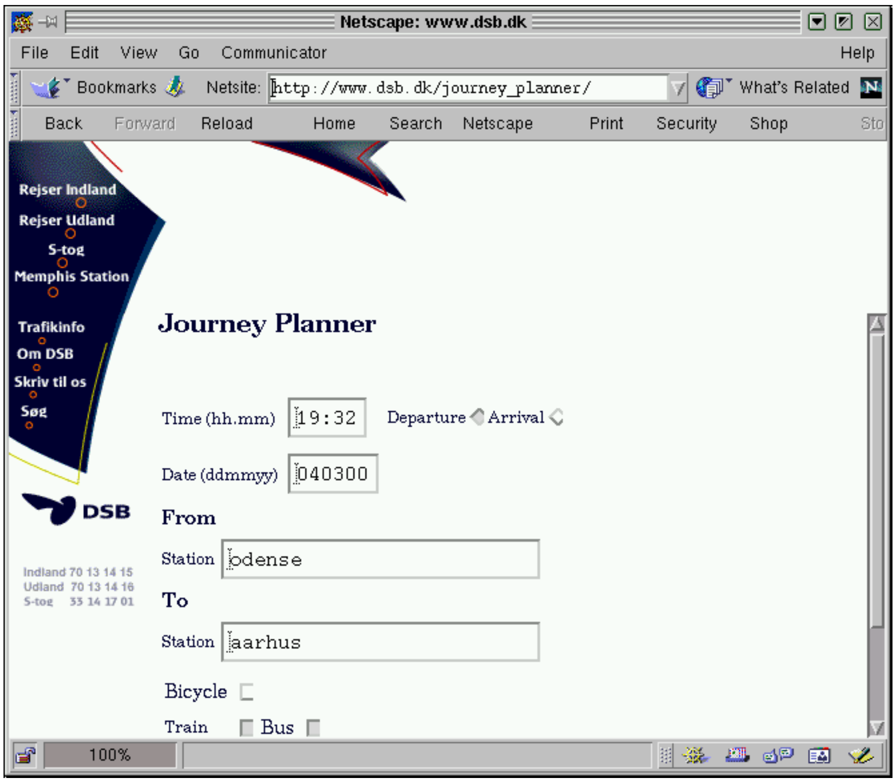
DSB (Danish Rail) have a journey planner which is based on train stations, and major bus stops. Figure 4.14 on the following page shows the initial form where the two end points for the journey as well as the arrival or departure time is indicated. Please note that the two surrounding frames (left side and top - the scroll bare on the right indicates the actual area available) leave only about 70% of the area to the application itself.

The search returns a number of potential journeys shown in figure 4.15 on page 43, with departure and arrival times, total expected travel time, and the number of changes necessary during the journey. The departure corresponding the best for the indicated period is highlighted with the blue bar. By accepting the default selection in the blue bar, and scrolling down to press a button, figure 4.16 on page 44 is shown. This is a direct connection, and it is possible to reserve a seat directly from this screen.

A more complicated request is shown in figure 4.17 on page 45 where a connection from Malmparken (suburb to Copenhagen) to Esbjerg (Western Jutland) is requested. A route involving three trains and a bus is suggested, which is very reasonable and probably the fastest too.

Rejseplanen is helped by the fact that the number of nodes in the grid can be small compared to `www.krak.dk`, but the presentation is not good enough. It is generally too hard to get to the information if you need a slightly alternate view from what the programmers expected.

I used to live next to Odense Sygehus, which is a stop on the Odense-Svendborg railroad, but


Netscape: www.dsb.dk
File Edit View Go Communicator Help
Bookmarks Netsite: http://www.dsb.dk/journey_planner/ What's Related
Back Forward Reload Home Search Netscape Print Security Shop
Rejser Indland
Rejser Udland
S-tog
Memphis Station
Trafikinfo
Om DSB
Skriv til os
Søg
DSB
Indland 70 13 14 15
Udland 70 13 14 16
S-tog 33 14 17 01
100%

Journey Planner

Time (hh.mm) 19:32 Departure Arrival

Date (ddmmyy) 040300

From

Station odense

To

Station aarhus

Bicycle

Train Bus

Figure 4.14: Rejseplanen1

where not all trains stop. The question regarding whether it would be faster to walk to the Odense Sygehus station or take my bicycle to Odense Station was very hard to answer by Rejseplanen, since it does not just simply allow access to the underlying database. I ended up calling the station and asking them to look in their paper copy.

4.8 Freshmeat.net

Freshmeat.net is a service for locating software. *a lot of products. users can announce software, and point peo*

Check if there still is companies which boost your presence on web search engines

Businesses - notably in porno - discovered this and polluted their web pages with keywords to boost thei

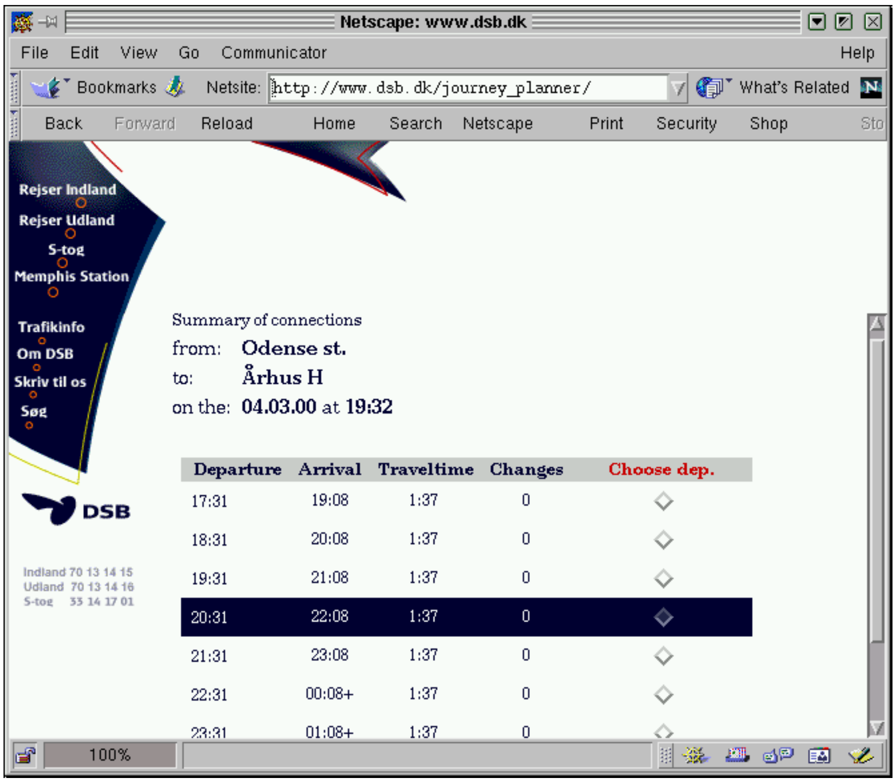
Look for statistics on search engines and "sex" - talk about that these businesses have a great interest in a

look for email about somebody who tried google and was pleased after being driven away from search e

ftpsearch.lycos.com[6] is a unique service provided as a byproduct of the main search engine at www.lycos.dk[7] where you may search for *images* on the web. The search engine does not look

[6]...ftpsearch.lycos.com... *http://ftpsearch.lycos.com*

[7]...www.lycos.dk...http://www.lycos.dk


Netscape: www.dsb.dk
File Edit View Go Communicator Help
Bookmarks Netsite: http://www.dsb.dk/journey_planner/ What's Related
Back Forward Reload Home Search Netscape Print Security Shop Stop
Rejser Indland
Rejser Udland
S-tog
Memphis Station
Trafikinfo
Om DSB
Skriv til os
Søg
DSB
Indland 70 13 14 15
Udland 70 13 14 16
S-tog 33 14 17 01
100%

Summary of connections
from: Odense st.
to: Århus H
on the: 04.03.00 at 19:32

Departure	Arrival	Traveltime	Changes	Choose dep.
17:31	19:08	1:37	0	◇
18:31	20:08	1:37	0	◇
19:31	21:08	1:37	0	◇
20:31	22:08	1:37	0	◇
21:31	23:08	1:37	0	◇
22:31	00:08+	1:37	0	◇
23:31	01:08+	1:37	0	◇

Figure 4.15: Rejseplanen3

inside the images it finds while traversing the net, but looks at the filename and the context in the web page for hits. It works reasonably well - most of the cactii images for my project was found this way. *Check on details how they do it.*

Search engines. How do they do it? www.909.dk, www.krak.dk, terraserver.microsoft.com. Encyclopedia Brittanica.
ASP+Access, tinderbox & bugzilla, javasoft (developers area), LXR+Bonsai. "Alle danskere paa nettet" - Dansk Journalistforbund, Pressemeddelelse. Greenspun selv?

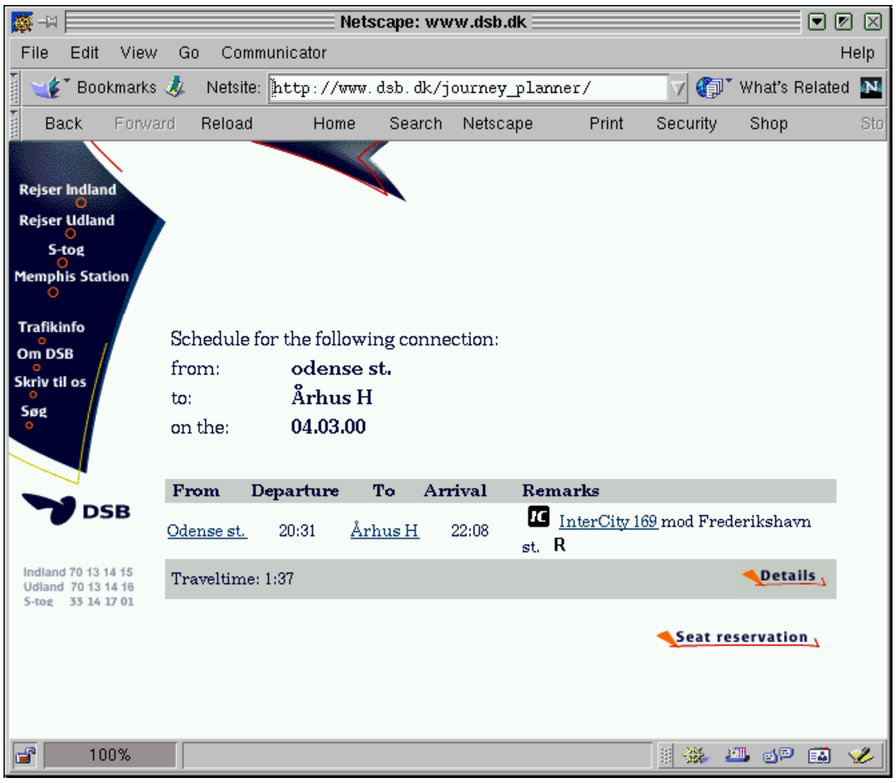
4.9 The Collection of Computer Science Bibliographies

http://liinwww.ira.uka.de/bibliography/index.html[8]

screens hot

This is a great web-site for scientists because in addition to the basic search facilities it allows you to download the BibTeX entry for the books and articles you find.

[8]. . . http://liinwww.ira.uka.de/bibliography/index.html. . .
`http://liinwww.ira.uka.de/bibliography/index.html`



Schedule for the following connection:

from:	odense st.
to:	Århus H
on the:	04.03.00

From	Departure	To	Arrival	Remarks
Odense st.	20:31	Århus H	22:08	IC InterCity 169 mod Frederikshavn st. R

Traveltime: 1:37 Details

Seat reservation

Figure 4.16: Rejseplanen4

4.10 Jydske Bank

THe only java-solution for home banking. Any comments?

4.11 3coms elektroniske papir

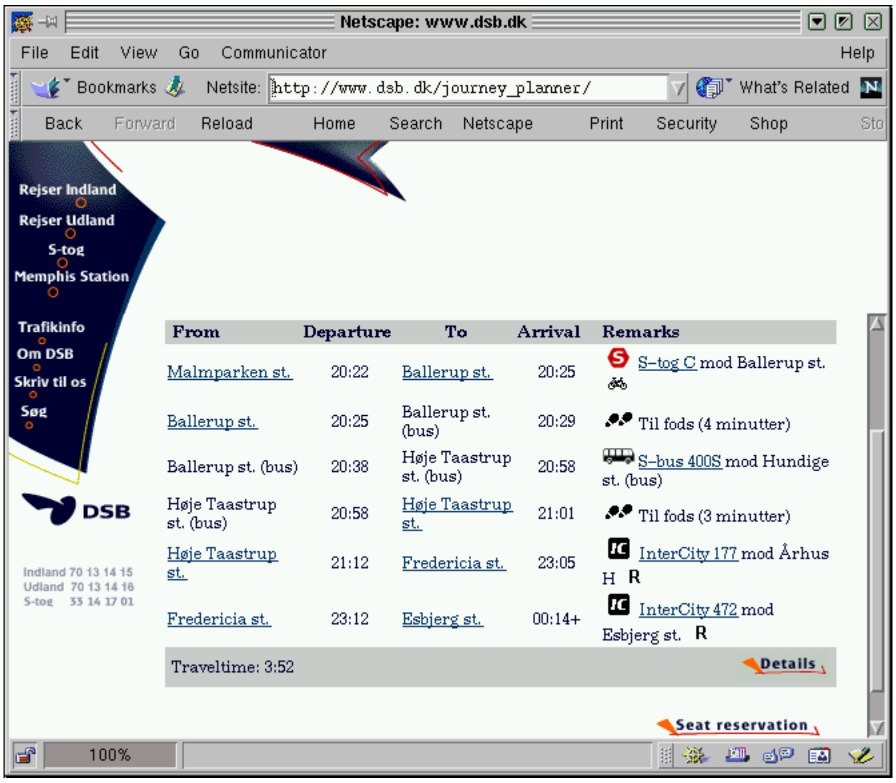
Beskriv projekt fra Jyllandspostudklip

4.12 Yggdrasil - a simple navigational framework

Based on an idea by Anders Stengaard Sørensen, I designed and implemented the Yggdrasil system to improve the use of the Intranet at AMROSE A/S. The basic idea was:

- Everybody should be allowed to publish document to the internal webserver. A string in the title of each document decides where it should go
- The navigational framework supporting user navigation between the various documents, should be generated by a program

The following quote is from the original announcement:



From	Departure	To	Arrival	Remarks
Malmparken st.	20:22	Ballerup st.	20:25	S-tog C mod Ballerup st.
Ballerup st.	20:25	Ballerup st. (bus)	20:29	Til fods (4 minutter)
Ballerup st. (bus)	20:38	Høje Taastrup st. (bus)	20:58	S-bus 400S mod Hundige st. (bus)
Høje Taastrup st. (bus)	20:58	Høje Taastrup st.	21:01	Til fods (3 minutter)
Høje Taastrup st.	21:12	Fredericia st.	23:05	InterCity 177 mod Århus H R
Fredericia st.	23:12	Esbjerg st.	00:14+	InterCity 472 mod Esbjerg st. R

Traveltime: 3:52 Details

Figure 4.17: Rejseplanen5

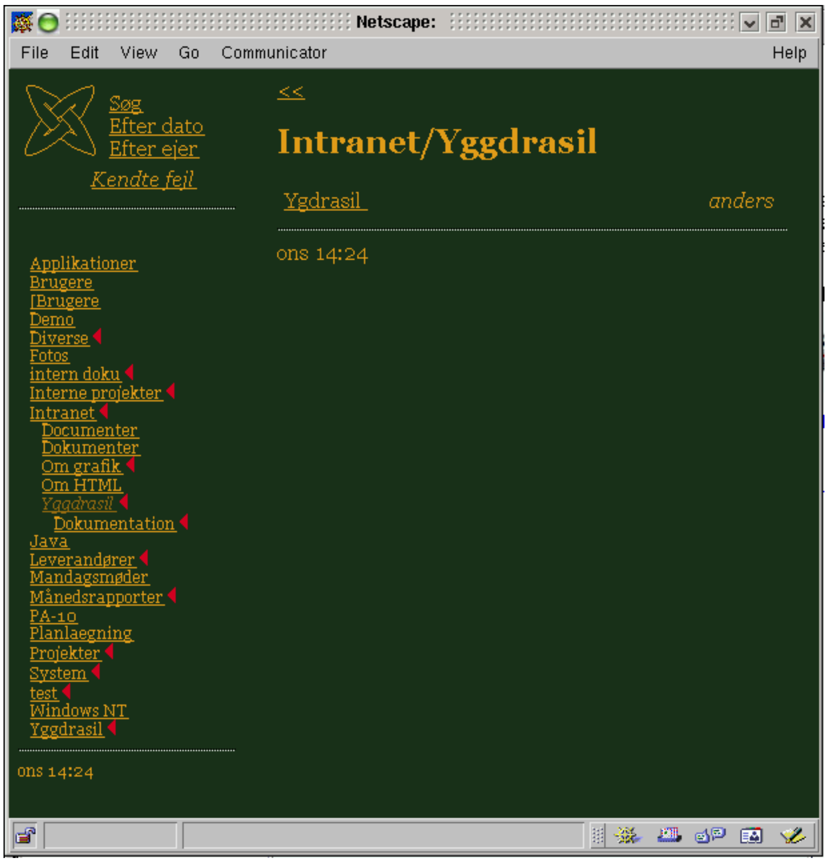
A need for an automatic maintainance of the central webpages was found during the construction of the intranet documentation in AMROSE A/S in 1997. These webpages primarily contain pointers to user written pages, as opposed to the more traditional approach of having a dedicated webmaster maintaining the various pages.

In AMROSE the need was more for the individual users to easily make additions to a common pool of documents, without having to appoint a webmaster. The Yggdrasil principle is to let the users put markers in the individual documents which identifies the placement of the link in the central webpages, and letting them maintain this information in the true spirit of the web, i.e. decentralised. Additionally, this is unrelated to any physical location of the given document, enabling users to provide information without having to obey rules regarding placements of physical files. Actually, a planned enhancement of Yggdrasil will allow the documents to span several webservers.

Yggdrasil provides several views on the document database. The primary one is the one where the user navigates the hierachy of the document categories , and where only the documents appropriate to your current selection are displayed. This goes very well together with the overview where documents are sorted according to their revision date. It is very easy to see which documents have been added since a given date.

Several prototypes were constructed, and the version in use when I left AMROSE, scanned all the individual web pages on the system, looked for the magic “[Class1/class2...]” string in the title, and created a large set of pages implementing

- A list of documents per user.



Søg
Efter dato
Efter ejer
Kendte fejl

Applikationer
Brugere
[Brugere
Demo
Diverse
Fotos
intern doku
Interne projekter
Intranet
Documenter
Dokumenter
Om grafik
Om HTML
Yggdrasil
Dokumentation
Java
Leverandører
Mandagsmøder
Månedsrapporter
PA-10
Planlaegning
Projekter
System
test
Windows NT
Yggdrasil

ons 14:24

<<

Intranet/Yggdrasil

Ygdrasil *anders*

ons 14:24

Figure 4.18: Typical Yggdrasil navigational screen

- A “Recently Changed Pages” list of documents sorted after entry date. This allowed for catching up quickly for a given period.
- A set of right and left side panes for the navigation. Clicking a link (like those shown in figure 4.18 on the facing page) changed the contents to another page where the wanted action had been taken.

The users were expected to

1. Write a document with the editor in Netscape Navigator
2. Set the classification string in the title
3. Upload it to their ftp directory on the server
4. Send an email to a special Yggdrasil user, which would trigger the rebuild of the pages (in addition to the nightly rebuild). The update would normally take less than 30 seconds.

Yggdrasil in itself consisted of about 10 pages of perl code. The search facility was provided by an ht://dig installation.

This worked very well in the beginning but had a fatal blow when it – due to the price of Netscape Navigator – was decided that the company was to have Microsoft Internet Explorer only. The users were to edit their documents in Microsoft Word, which had very poor support for exporting to HTML and the facilities for uploading documents to the webserver was also rather tedious to use. Users of Word could not enter the information fields recognized by Yggdrasil, making it even harder to publish documents.

Users started publishing raw Word documents with small HTML wrappers, and the usage quickly dwindled since it was too cumbersome to use.

Unfortunately, the management never fully backed the Yggdrasil project, new users were not informed about the system and required to use it, meaning that these problems were not resolved and after a time of neglect the system was no longer in use.

The Cactus system is my response to the experience learned from this project, as discussed in chapter 10 on page 87.

4.13 The TOM browser - converting documents online

screendump The TOM browser[9] is a document conversion system which allows users to upload files and provide URL's to document on the web, which is then converted to one of numerous formats. The default is to autodetect the file type, and convert to a viewable format.

[9]...The TOM browser...`http://wheel.compose.cs.cmu.edu:8001/cgi-bin/browse`

Chapter 5

SGML, XML and DTD's

"If you don't read the manual before you enter the dungeon, you might end up being killed"
Rumour in the Nethack computer game – about 1993

A major ingredient in developing large, complex systems with several layers of data and meta-data, is the ability to *abstract* between different levels of the system.

- *Operating systems* provide the "file" abstraction, so the programmer does not have to think about the underlying hardware when reading and writing information, and these operations are normally optimized for speed.
- Expensive printers provide the *PostScript language* which is an abstraction for "electronic paper". Any Postscript file can be printed to any PostScript printer, without considering the specifics of the device. This was a major step forward from the traditional matrix printer where the application had to generate bitmaps for the printer to print.
- *Programming languages*, like C, provides the abstraction of a "virtual machine". The programmer does not have to worry about how large an integer, and how the best way to implement a loop on this particular processor is. The compiler takes care of that, leaving the programmer to concentrate on problems at a higher abstraction level.

The same need arose in the printing industry, where SGML (Standard General Markup Language) was developed in the late sixties and early seventies[1] to facilitate a transfer from "specific coding" to "generic coding" (Goldfarb mentions using "heading" instead of "format-17"). This allows authors to concentrate on *what* they write, instead of *how* it is presented. In Goldfarb's own words in 1971:

> The principle of separating document description from application function makes it possible to describe the attributes common to all documents of the same type. ... [The] availability of such 'type descriptions' could add new function to the text processing system. Programs could supply markup for an incomplete document, or interactively prompt a user in the entry of a document by displaying the markup. A generalized markup language then, would permit full information about a document to be preserved, regardless of the way the document is used or represented.

[1]...the late sixties and early seventies...`http://www.sgmlsource.com/history/roots.htm`

Today, almost 30 years later, this is coming to the users of the world wide web, as the presenting technology moves into the web browsers - with XML being defined as a subset of SGML - and they will be able to fully present XML documents individually tailored to the users preferences. The author can provide richly annotated content, along with a suggested way to view the content (in form of a style sheet). The user may freely choose override some or all of these suggestions.

This is a radical change from the current situation where HTML allows an information provider to be so specific in markup that it makes it unusable in some browsers, and without any means for a user to specify otherwise. An example is the font size change commands, which can render a site totally unreadable in Netscape.

Hopefully the mechanisms for user customizations will be so well developed when XML is commonplace in browsers and web sites, that users can disable this behaviour.

It appears to me that there will be two major uses for XML, one is for transporting data between applications (like a database and a browser), and the other will be as a media for interchanging information between humans. My expectations is that a few, very well designed, standard XML definitions will be created and that most public XML documents will be written to comply with one or more of these standards, and web robots (like search engines and web crawlers) will know these standards and be able to extract meta-data from them.

The initial XML-editors are already arriving. It will be very exciting to see what they can do in 5 years.

5.1 "HTML is a SGML DTD" - the concepts

Table 5.1 on the facing page shows how the most frequently used acronyms relate when mapped to the programming domain. See section 2 on page 5 for the full descriptions.

Basically SGML (Standard General Markup Language) covers a large family of document markup languages, which all share that a given SGML-document *must* comply to a DTD (Document Type Definition). DSSSL is used to convert a SGML document into something else. DSSSL's are written in a Scheme like language (see [Dybvig(1996)] for a reference manual).

XML is a light-weight version of SGML designed to be embeddable in a browser. XML-documents *may* comply to a DTD, or be stand-alone ("DTD-less"). XSL is used to convert a XML document into another XML document (XSLT) or into the generic page description language XSL-Formatting Objects (XSL-FO).

XHTML is a XML-version designed to be presentable by HTML-4.0 browsers.

5.2 The author should provide content, not do layout

SGML came originally to because the authors at IBM were not consistently marking their documents up, resulting in suboptimal documents. This was because they were doing physical markup instead of logical markup.

Documents	**Programs**
SGML	Family of programming languages (C/Pascal/Lisp...)
XML	Another family of programming languages. (XML is a subset of SGML)
DTD	*Backus* -Naur notation of a given language, defining how it looks
HTML	A certain programming language like C
XHMTL	Another programing language like Java Well written programs can run both in C and Java
DSSSL	A report generator which takes a XML program as input and produce a report, another program or something else
XSL	A report generator which takes a XML program as input and produce a report, another program or something else

Table 5.1: The analogy between programming languages and SGML

> Modern word processors have been constructed around the "What you see is what you get" paradigm, where a great deal of effort has been spent on allowing the user to see exactly on screen what would be printed on paper. Unfortunately the limited resolution and size of a modern computer monitor cannot show the whole page in high resolution, and the user must therefore work in small "regions" without being able to see the whole page, or perhaps even several pages *A tufte reference here?*. Microsoft Word provides a draft mode where just the text is shown without markup; WordPerfect provides a "Reveal Codes" mode where the user can edit the text along with the tags.

The most recent versions of Word and WordPerfect *wp9 should have sgml builtin* have had styles built in, where the user could specify "heading" and similar categories for selected text, but have been very passive in helping users to employ these categories, meaning that most of these document have had almost no semantic markup, but only visual markup.

In combination with this the web publishing tools by Microsoft and others have blatantly ignored web standards, and used all kinds of font modifying tags instead of the HTML-tags conveying *meaning* about the text. These tags often implement the particular settings that the author had on her word processor, so that the author is effectively doing layout on the web.

This is not a new phenonomen, however. The best way to typeset mathematics have been the LaTeX system[2], (see section *refsec:tex-and-latex*) where the author basically is told to write what she wants to and let TeX do the rendering. Even so, it is often very tempting even for the most dedicated purist to want to change the layout to make a small visual change, simply because they *can*. Unfortunately very few have the expertise to do this right, so the changes usually make it worse. I have seen a book where the author had prepared camera-ready copy for his LaTeX-document. The book was printed in A5, but he had apparently handled this merely by increasing

[2]...LaTeX system...`http://www.tug.org`

the margins of a normal page, and nothing else. The result was that the layout looked awkward - the line spacing seemed too large for the page. This would not have happened if a professional layouter had produced the book!

A layouter may also want to change the whole layout of a book, for example if a new edition is to be part of a series with another distinct style. The publisher may want to turn the book into webpages or a CD-ROM, or into a Braille version for blind persons. It might even be read aloud by a computer.

HER SKAL O'REILLY DIMSEN IND!

It should not bother the author during the writing process how the finished document will look, as long as she marks up the document appropriately, providing suffiecent meta-data about the content.

As the possible annotations are decided by the DTD, it is important to choose a DTD carefully. A well annotated document can be used for many purposes, not all of which was envisioned when the document was written. Ten years ago no one had heard about the world wide web - within the next few years, it will probably be the main use of SGML in the world.

5.3 The *creation* of HTML

Look up creation of HTML on the net. Look for reasons why HTML was chosen to be a SGML DTD instead of

The w3c have notes and DTD's describing the first versions of HTML[3].

HTML: 1.0 (original, simple, <hr> tag was added due to popular request), 2.0, 3.0 never made it, 3.2 tables, 4.0 w3c straightens up things. Netscape added new tags *ad hoc* often. In the mean time HTML has been made to do things it was never meant to do originally, for presentation purposes (large imagemaps in tables without any textual information), and there is still a need for new features that designers want. *W3C saw that blindly adding new features to HTML would result in an even more bloated standard* with a lot of backward compatability to maintain - a modern browser must still be compatible with the bugs in Netscape versions 2 and 3. These were renowned for being very lenient towards users providing erroneous HTML, and have started a tradition of browsers doing their best to interpret what the users *might* have meant. This makes it hard to use HTML as a generic information data format, since parsers are complex due to the many exceptions.

The W3C took a bold step in saying: We need a new format - the eXtensible Markup Language. XML!

5.4 The *creation* of XML

XML was as HTML built on SGML, but with a much stricter syntax than HTML and a lot less facilities than in SGML. XML is a new standard - it was made a W3C recommandation in 1998, and

[3]...The w3c have notes and DTD's describing the first versions of HTML...
`http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/MarkUp.html`

the accompanying XSLT and XPath in November 1999.

What was the design goals

The resulting language for designing languages has these features:

- A very strict syntax which every XML-document must comply to (which is what a *well-formed* document does). All open tags must be closed explicitely. Tag attribute values must be in quotes.
- Easy to parse and generate. The strict syntax makes the parser much simpler than a SGML parser.
- The DTD is not mandatory. DTD-less documents does not have a DTD to conform to, and must only be well-formed. *Unicode?*

TRANSFORMATON

FORMATTING

5.5 XHTML - XML compliant HTML

[The XHTML W3C recommandation[4].]

A big problem with XML in general is that it cannot be viewed by anything the majority of users have today. It is possible, however, to specify a form of XML called *XHTML* which bridges the gap between the two worlds of XML and HTML. Documents corresponding to a XHTML DTD are parsable by HTML-4.0 compliant browsers if a few, simple guidelines are followed:

- Elements (the HTML tags) must be closed correctly.
- Attribute values must be quoted
- Element names and attributes must be given in lower case
- Empty elements must either have an end tag, or the start tag must end with "/>". I.e. a horizontal line which is written as <hr> in HTML, must be written as "<hr />", where the space is important to make this acceptable to HTML-4.0 parsers.

The well-formedness of an XHTML document is achieved*...*

```
HTML: <hr noshade>
XML:  <hr noshade="" />
```

Why is this smart??? Not only pure output but also input for another XML transformation

XHTML is an XML-variant designed to be viewable in HTML-4.0 compatible browsers (without XML support). XHTML-documents are well-formed XML documents, while at the same time being valid HTML-4.0, so that XHTML can be the target of a standard XML-transformation as well as a source for initial XML-processing.

[4]...The XHTML W3C recommandation...http://www.w3.org/TR/xhtml1/

This is very different from the usual situation, namely that HTML is the result from *formatting* a document, where the result is not XML any more.

Note: XHTML style sheet generated files does not trigger correct change to UTF-7 characterset.

WHICH CONVERSION UTILITIES ARE THERE? TIDY? READ XHTML? WRITE XHTML? WHAT DO W3C SUG

5.6 Which dialect of SGML should be used?

The Document Type Definition (*DTD*) defines exactly the way a SGML/XML document can look, when conforming to the DTD.

- which tags are valid
- which tags can appear at any given part of a document
- which attributes are valid for a given tag
- which *entities* (macros) can appear in a document, both for expansion strings but also for Unicode characters unavailable to the document author
- *Others?*

There is a lot of different DTD's available to many different purposes, since basically everywhere a datafile is needed XML can be used with a suitable DTD. *ChemML*, *SVG?*. If a need arises it is just a matter of creating a suitable DTD to work with it.

Unfortunately, for each and every pair of DTD and output format an XSL-style sheet must be written, tested and maintained. Therefore it is a very good idea to use a commonly used, documented, and well-thoughtout DTD for the documents, and several other groups have already done such work.

The Cactus system uses the DocBook XML V3.1.7 DTD with the DocBook XSL

5.7 How can a *ML document be viewed?

In order to be rendered into a view, a *style sheet* must be applied to the document. Style sheets convert the tags in the document to a given output format, and depends on (DTD, output format pairs).

GRAPH DEPICTING STYLE SHEETS BEING APPLIED FOR DIFFERENT OUTPUT FORMATS

Commecial products usually have a viewer which can apply the style sheet directly `examples? Panorama? Framemaker+SGML`, but are due to licensing costs and supported user base rarely an option for general for documents targeting end-users, and I have not tried any of them. Table 5.2 on the facing page discusses the current *de facto* document formats on the Internet.

The Microsoft Internet Explorer 5.5 (not yet available at the time of writing) will be able to transform XML documents with XSL-style sheets internally complying with the W3C recommendation. The previous version of IE was available on Windows, Solaris and HPUX.

Format	
HTML	Browsers are available for any modern platform. HTML-4.0 compliant browsers are Netscape Navigator 4, Internet Explorer 4... *More?*
PDF	Adobe Acrobat Reader is available for most mainstream platforms *URL*. The Adobe Acrobat Viewer is available for any platform with Java *URL*. The Open Source project Ghostscript *URL* is available for these and other platforms.
Microsoft Word	The `doc` and `rtf` file formats are widely used, but needs a full word processor to format properly. Anything else but Microsoft Word gives inferior results, and Word only runs on Windows.
PS	
PDF	

Table 5.2: Frequently encountered document distribution formats

The Mozilla browser in its pending release as Netscape Navigator 5.0 will not be able to apply XSL-style sheets, but only CSS cascading style sheets. This *decision is rather old and may have changed*.

For the immediate future web publishing in XML implies a need for server side conversion to other formats. The available Open Source software for this is surprisingly small, but that is rapidly changing. Please note that XML style sheets still implies SGML style sheets, which is interesting because these are more mature due to longer use (see sectionṽrefsec:docbook-xml-to-rtf for more details).

5.7.1 SGML style sheets

CONFIRM THE BEHAVIOUR WRITTEN BELOW IS EXCACT

SGML style sheets are *written in DSSSL* (others?), which is a Lisp-dialect, and the usual Lisp-conventions apply. Several

several? *what can this do?*

The DocBook reference recommends using `jade` written by James Clark, to do DSSSL conversions. *long processing times later implemented in XT with threads to allow faster output*.

Show a sample stylesheet

5.7.2 XML style sheets

XSL - the XML style sheets - are written in XML too. The tags in the name space “xsl:” describe what is to be done with the source XML tree, in terms of tag-remapping (<para> to <p>), sorting of subtrees, and several programming constructions like “if”, “while” and “foreach”. New tags can be constructed, existing tags can be altered, and *yes?*

a reference listing here

This style sheet - lifted from the DocBook distribution - creates an outline of a DocBook document by looking for headline tags, and make a tree of them *Eh?* Line breaks have been introduced for readability.

```
<!--
| Identity Transform Stylesheet
| ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
| $Author: ravn $
| $Date: 2000/03/02 21:55:31 $
| $Source: /home/ravn/CVS/SPECIALE/SPECIALE/x-outline.xsl,v $
| $Revision: 1.1.1.1 $
+-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version='1.0'>

<xsl:output method="html"/>

<xsl:template match="/">
  <html>
  <body bgcolor="white">
  <xsl:apply-templates/>
  </body>
  </html>
</xsl:template>

<xsl:template
match="set|book|part|reference|article|sect1|sect2|sect3|sect4|sect5|section|
       simplesect|preface|chapter|appendix|example|figure|table|informaltable">
  <xsl:param name="treeicon">n</xsl:param>
  <xsl:variable name="level" select="count(ancestor-or-self::*)"/>
  <xsl:variable name="title" select="title"/>
  <xsl:variable name="this"  select="local-name(.)"/>
  <img src="s.gif" border="0" height="1" width="{22*$level}"/>
  <img border="0" src="{concat($treeicon,'.gif')}"/>
  <img border="0" src="{concat($this,'.gif')}"/>
  <xsl:variable name="color">
    <xsl:choose>
      <xsl:when test="$level = 1">navy</xsl:when>
      <xsl:when test="$level = 2">red</xsl:when>
      <xsl:when test="$level = 3">blue</xsl:when>
      <xsl:when test="$level = 4">black</xsl:when>
    </xsl:choose>
  </xsl:variable>
  <span style="color:{$color}"><xsl:value-of select="$title"/></span>
  <xsl:if test="local-name(.)='example' and programlisting/@role">
   <small style="font-family:courier">(<xsl:value-of
       select="substring-after(programlisting/@role,'-')"/>)</small>
  </xsl:if>
  <br />
  <xsl:apply-templates/>
</xsl:template>
```

```
<xsl:template match="*|@*|comment()|processing-instruction()|text()"/>

</xsl:stylesheet>
```

Since the DocBook reference is very vague on this matter (the book went to press before the XSLT recommendation was finalized) the Internet has been a great help. Table 5.3 lists the tested XSLT processors which conforms to the W3C XSLT-19991102 recommendation.

XT	XSLT processor based on the XP parser. Both written in Java *by James Clark*.
Xalan	XSLT processor written in Java originally based on the LotusXSL processor by *alphaworks?*. Currenly under heavy development by the Cocoon team (`xml.apache.org`).

Table 5.3: Leading XSLT processors implementing *XSLT-19991102* in February 2000

During testing it showed very quickly that the Java tools are quick enough to use for casual use (*do a few benchmarks*), even though they are not yet tuned for optimal performance. Additionally

Even so, implementations in C should only be considered by designers if Java clearly cannot do the job. The advantages of Java over C (*reference big list, probably at JavaSoft?*) in program development

- Very hard to do pointer access
- A large set of tested libries

Xalan is at the time of this writing in a development state, and contains bugs which causes it to crash on some of my sample DocBook XML document. After that I turned to XT which is extremely stable and reliable, and that I have used for the rest of the project - my only complaint is that it stops after reporting the first error instead of parsing the whole document.

When the Cocoon project stabilizes they will have a high end XSLT-engine embedded as a servlet in Apache, which makes this a project to watch. The technology is at the time of this writing still immature, but very promising.

The platform independance of Java indirectly prompted the implementation of remote filters in Cactus. Very early in the evaluation Xalan threw the above mentioned exception when processing a medium sized DocBook document. In order to rule out errors in the underlying Java environment on Asserballe (see section A.3.1 on page 114 for technical details) an identical run was made with JDK 1.2 on Aalborg. The same error occured, but in less than one second instead of ten. Experiments showed that Java programs consistently runs 15 times faster on Aalborg than on Asserballe. *and so what?*

`incoming.xml`

```
<?xml version="1.0"?>
<?xml-stylesheet href="incoming.xsl" type="text/xsl"?>
```

```
<?cocoon-process type="sql"?>
<?cocoon-process type="xslt"?>

<page>

 <connectiondefs>
  <connection name="foo_connection">
   <driver>org.gjt.mm.mysql.Driver</driver>
   <dburl>jdbc:mysql://localhost/Cactus</dburl>
   <username>XXXXXXXXX</username>
   <password>XXXXXXXXX</password>
  </connection>
 </connectiondefs>

 <query connection="foo_connection" tag-case="lower">
  select when, mime, user, how, filename, comment,length(item) as l  from incoming
 </query>

Testing
</page>
```

incoming.xsl

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<?cocoon-process type="xslt"?>

<xsl:template match="page">
<html>
<head>
<title><xsl:value-of select="title"/></title>
</head>
<body bgcolor="#ffffff">
<img src="gr/organ_pipe_cactus-single.jpg" align="left" />
<xsl:apply-templates/>
</body>
</html>
</xsl:template>

<xsl:template match="ROWSET">
<table border="0" bgcolor="#F0F0F0">
<tr><th>when</th><th>mime</th><th>user</th><th>how</th><th>filename</th><th>comment</th><th
<xsl:apply-templates/>
</table>
</xsl:template>


<xsl:template match="ROW">
```

```
<tr>
<td><xsl:value-of select="when"/></td>
<td><xsl:value-of select="mime"/></td>
<td><xsl:value-of select="user"/></td>
<td><xsl:value-of select="how"/></td>
<td><xsl:value-of select="filename"/></td>
<td><xsl:value-of select="comment"/></td>
<td><xsl:value-of select="l"/></td>
</tr>
</xsl:template>
</xsl:stylesheet>
```

5.8 Converting documents to XML

XML is currently having the same problems that HTML had in its initial years, namely lack of software support. The difference is that HTML is so lenient that it is easy to write HTML by hand in a text editor. Not so with XML.

The difference this time is that Microsoft has been an active participant in the *definition of XML*, and has made *???* about Windows 2000 supporting XML directly *in what precisely?*

Even so there is still a severe lack of conversion tools for the many files out there in non-SGML based formats like Word, Excel, LaTeX, TeX, Lotus Notes, etc., which must be available before the documents can be converted.

Majix	Java based RTF to XML, which I have adapted to DocBook
pod2docbook	Converts Perl documentation to XML
wordview *others* *sp?*	Converts *Word 6-95-97* to HTML. I have adapted them to produce DocBook output *better do it too*
tex4h	package for LaTeX which can *what was it now?*
others	??

Table 5.4: Conversion tools to XML [February 2000]

Due to the rather inflexible DTD-requirements in XML, it is most convenient that such a tool generates stand

Move filter descriptions to Cactus chapter discussing filters? ... wordperf 9 should be ablt ot do sgml

5.8.1 Majix - RTF/(DOC) to XML

Majix *URL, company description* is an RTF (for Word 97 documents) to XML converter. I have created a configuration and a small style-sheet which can render a RTF file to DocBook XML.

If run with the Microsoft Java Machine ("`jview`") while logged into a Windows machine with access to the GUI, it can launch Word to convert a given DOC file to RTF and process it, with very reasonable results. I have experimented with making this accessible to Linux by installing Microsoft

Services for Unix on a NT machine with Word installed, and using telnet to invoke Majix. Unfortunately this doesn't work. Another approaches (*which will be tested if time allows*) includes having a pseudo-user watching a directory on a NT-server and launching Majix whenever DOC or RTF documents arive. These can at best be described as kludges.

Even though it shows great promise, it has been a year without new releases (including the promised Pro version), and the source is not available.

Conclusion: The Majix software is currently best for personal end-user conversions.

5.8.2 pod2docbook - POD to XML

The POD format is created for writing documentation for the Perl programming language, and was created to be "...an idiot- proof common source for nroff, TeX, and other markup languages, as used for online documentation" by Larry Wall (*the perlpod manual page*).

The original release produced SGML DocBook. I have submitted patches to the author which allows the user to choose between XML and SGML DocBook.

Conclusion: This do a good job for documents written in POD, and will probably with time take over as the default formatter for printed versions of Perl documentation.

5.8.3 tex4h - convert TEX to XML

What daelen does it do

Conclusion:

5.8.4 wordview with friends

Parses binary stream and produces HTML, WML plus more.

5.9 Rendering HTML on the fly

look at docs.sun.com - what daelen do they do? Render SGML to HTML on the fly?

When the decision has been made to provide information in *ML on the web server, another decision must be made. Should the various renderings to PDF and HTML be created before they are requested, or when the user asks for a certain rendering?

Look for an article on static versus dynamic

As always it depends on the nature of the data.

Static renderings is probably suitable for documents which rarely change their rendering or take long time to render, and which just must be served as fast as possible. Often there is a desire for an inexpensive solution[5], and in that case a standard Apache with a rich set of prerendered documents in a flat file system may very well be optimal. Personal experience has shown this to be a very robust solution which very rarely requires human intervention. esr *Eric S. Raymond reports that a modern PC with A*

Dynamic rendering is suitable whenever the data changes very often, or there is a wish for the user to be able

The history and usage of SGML. Creation of XML. Describe document validation, and conversion (XSLT) to XML and other formats. Freedom from restrictions of HTML. Problems with tons and tons of DTD's. The need for well-documented standard, robust, supported DTD's (current ones: TEI, ebook [buh], DocBook). HTML conversion utilities can be tailored to generated DocBook XML for SSP (currently pod2docbook, Excel xls2xml). Found that SGML (jade) is powerful but too slow for on-the-fly stuff, as opposed to XML rendererer. Several to choose from if they conform to the w3c standards (DOM and SAX). Who uses DocBook at the moment? Man pages in DocBook on Solaris (look on machine). IE50 cannot show "simple" DocBook XML yet but it is the goal of that project.
the xsl/docbook/contrib/outline/outline.xsl is an excellent sample of a small, powerful style sheet.

[5]"Inexpensive" here is defined in terms of man hours needed to learn and maintain the web server, as well as the hardware needed

Chapter 6

The DocBook and TEI DTD's

6.1 DocBook considerations

DocBook[1] is a DTD designed for documenting software. Used by Sun, FreeBSD, O'Reilly and others.

The SGML is converted to HTML by applying a HTML style sheet for DocBook, and sending it through jade w

screen shot The generated navigation tags are a little troublesome in Lynx, but nice in Netscape. A "up", "next" and "previous" are available combined with a "Top title"->"Chapter title" -> "Section title" navigation bar at the top.

The localization code is manually maintained.

It is possible to have a DocBook XML-file which can be processed both with XSL-stylesheets *and* DSSSL-stylesheets, by letting the !DOCTYPE header point to the DocBook XML DTD along with system-dependent path (which is required by XML). A sample header for DocBook is listed below:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE article
  PUBLIC "-//Norman Walsh//DTD Simplified DocBk XML V3.1.3.6//EN"
  "/home/ravn/sgml/docbk/db315/docbookx.dtd">
```

If the `standalone="yes"` field is set in the <?xml?>-tag instead, it means that the XML-tools will not validate the contents against the DTD *does this increase performance?*, but still apply the style sheets.

Generally XSL conversions (commands starting with java) are fast enough to be done "on-the-fly", where DSSSL conversions (commands starting with jade) are *too slow for this (timing?)*.

[1]...DocBook...http://www.docbook.org

Format	Command
XHTML-1.0	java com.jclark.xsl.sax.Driver $< ../docbook/xhtml/docbook.xsl $@
HTML-$\boxed{4.0}$	java com.jclark.xsl.sax.Driver $< ../docbook/html/docbook.xsl $@
XSL-FO	java com.jclark.xsl.sax.Driver $< ../docbook/fo/docbook.xsl $@
RTF	jade -t rtf -d $(HOME)/sgml/dsssl/docbook/print/docbook.dsl $(HOME)/jade/pubtext/xml.dcl $<
MIF (Frame)	
TEX	
PDF (from FO)	java org.apache.fop.apps.CommandLine <@
PDF (from FO)	tex '&pdffotex' document.fo
PDF (via TEX	
OThers?	

Table 6.1: Output formats possible for a DocBook XML document

6.2 TEI – Text Encoding Initiative

6.3 Converting documents to other formats

6.3.1 SGML to XML

```
sx -biso-8859-1 -xlower input.sgml $>$ output.xml
```

The DTD for the document *must* be registered in the $\boxed{\textit{catalog}}$. Even so a number of warnings and errors may be printed, since some SGML-constructions cannot be represented in XML, and some entities are unknown. This must be handled manually - SX is not a tool do this perfectly.

6.3.2 XML to RTF

Use `jade` with the appropriate stylesheet and $\boxed{....}$

```
        jade -t rtf -d docbook.dsl xml.dcl inputfile.xml
```

where `docbook.dsl` is one of the stylesheets from the $\boxed{\textit{DocBook Modular StyleSheets}}$ (`dsssl/docbook/print/`
and `xml.dcl` comes from the Jade distribution (`pubtext/xml.dcl`)

The resulting RTF-file is compatible with Word-97, Word-95 (claimed by author), StarOffice, $\boxed{\textit{other texteditors}}$

$\boxed{\textit{other backends}}$

SQL to XML :: Coocncococns sql processsor

O'Reilly – The publisher of well-renowned reference books – have for quite some time had their authors write in SGML for easy rendering to PostScript and other versions (like their CD-ROM editions where several books had been made available as indexed HTML-pages), and they have made their DSSSL scripts for the purpose available on the net[2].

[2]. . . their DSSSL scripts for the purpose available on the net. . .
`http://www.oreilly.com/people/staff/crism/dsssl/orastyle/`

Chapter 7

Standard Query Language, Databases and Webservers

"You've got to be carefully taught"
South Pacific – Rogers and Hammerstein

7.1 SQL

SQL (the Standard Query Language) was developed in the early 1970'es by Don Chamberlin and Ray Boyce at IBM Research, to implement the mathematical notation in which E. F. Codd had defined relational databases [Codd(1970)]. Today SQL has grown into the *de-facto* standard for communicating with a relational database, which is being enforced with the modern ODBC and JDBC drivers which provides a "tunnel" between the client program and the database where the client can send a SQL-command and get the result back without caring about *how* the data are transmitted, much in the same way that the TCP/IP protocol allow one computer to send a data-stream to another computer without concering about how the data is sent.

Unfortunately this is as much abstraction as you will get from these "SQL drivers" – the ability to execute a SQL-command and retrieve the result. Any further abstraction from plain SQL is still left to the application programmer which means that even the state-of-the-art applications talk to the database in a command-line fashion. The SQL looks like this (lifted from the MySQL tutorial):

```
CREATE TABLE shop (
 article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT NULL,
 dealer  CHAR(20)                 DEFAULT ''     NOT NULL,
 price   DOUBLE(16,2)             DEFAULT '0.00' NOT NULL,
 PRIMARY KEY(article, dealer));

INSERT INTO shop VALUES
(1,'A',3.45),(1,'B',3.99),(2,'A',10.99),(3,'B',1.45),(3,'C',1.69),
(3,'D',1.25),(4,'D',19.95);
```

```
SELECT * FROM shop;
```

article	dealer	price
0001	A	3.45
0001	B	3.99
0002	A	10.99
0003	B	1.45
0003	C	1.69
0003	D	1.25
0004	D	19.95

Figure 7.1: The output from the SQL-example

which then returns the table shown in figure 7.1:

The table shown here was originally in ASCII (manually put into a table for presentation purposes); inside an application program it is common forthe application to work with a row at a time.

7.1.1 SQL - really a standard?

SQL has evolved over the years. The latest ANSI standard is SQL92, which is discussed in [Melton and Simon(19 *as well as the standard bodies online*, and it provides several levels of compliancy a given SQL-database can adhere to. Even so, there are several facilites which are still not covered by the standard:

- **unique numbers** – having a row or function which generates a unique number every time it is called. This is called *sequence numbers* in Oracle, and *?* in MySQL.
- **date and time functions** – querying for the current date and handling date/time fields in the database is not standard. Additionally it is common for a vendor to have additional formats for timestamps which is incompatible with other vendors.
- *others?*

These vendor specific extensions from SQL92 are normally done in order to allow faster database queries, and are therefore interesting for the application programmer who wants the fastest possible implementation. Unfortunately this usually ties the application dependent to a specific database server, which rarely is in the interest of customers (who might have a database already they wish to use), meaning that the application programmer must be aware which facilities are acceptable to use. I have not found application frameworks which provide a generic SQL interface and parse it to the underlying database.

7.2 SQL – getting started

I was reintroduced to SQL rather abruptly in October 1999, where I needed one or more good introductory texts for ASP-programming with MS-Access as the database backend[1], and found almost none on the Internet. Martin Damsbo, who had previously written the PHP3+MySQL based WACO system [Damsbo(1999)], pointed me towards the tutorial in the MySQL reference manual[2], which was what he had used.

Even though several databases were availabe, I decided to dvelwe into MySQL since it had excellent Perl support which I had chosen for developing the Cactus system (see 12.1.4 on page 103 for reasons), and was OpenSource software available on Linux.

The MySQL tutorial is reasonable for a person knowledgeable in SQL to get acquainted with the MySQL database, but it falls a bit short of being a generic introduction.

A *much* better choice for me was Phil Greenspuns SQL-introduction[3] (see [Greenspun(1998)]) which was specifically written as introductory material for college students, based on his long-term real-life experience with the heavy duty web-service at ArsDigita[4], which also serves photo.net[5] where this SQL-introduction resides..

Greenspun start with the basics and moves to advanced topics in a steady pace, and is an excellent read. There is a strong emphasis on Oracle 8i, but is easily adapted to other databases like MySQL. He has numerous examples, and several references and evaluations of other texts. Additionally the text is sprinkled with thumbnail photos, which links to a large version along with a link to another part of his site.

a swkveen shot?

This in addition with the comments from the readers to the text, makes this a very valuable resource for users. The SQL-introduction is available both online and as a book.

7.3 Some of the possible ways of accessing a database from a web-server

Armed with the necessary, solid SQL knowledge, it is reasonable to consider the current approaches of interfacing a Web server with a SQL-database. At least the following options exist:

ODBC – The Microsoft "Open Database Connectivity[6]" which provided for the first database access standard on Windows. Since superceeded by OLE DB which does basically the same thing on Windows.

[1] I learned quite a lot in those few days. Basically ASP is a very good idea, and Frontpage is extremely unsuited for working with it

[2] ...the tutorial in the MySQL reference manual...
`http://sunsite.auc.dk/mysql/Manual_chapter/manual_Tutorial.html#Tutorial`

[3] ...Phil Greenspuns SQL-introduction...`http://photo.net/sql`

[4] ...ArsDigita...`http://www.arsdigita.com`

[5] ...photo.net...`http://photo.net`

[6] ...Open Database Connectivity...`http://www.microsoft.com/odbc`

JDBC – The “JDBC Data Access API[7]” does the same thing for Java-programs as ODBC do for Windows-programs.

P[8]erl DBI – The “Perl Database Interface” allows Perl programs to speak to Adabas, DBMaker, Fulcrum, Informix, Interbase, Oracle, Solid, Empress, Sybase, Illustra, Ovrimos, PostgreSQL, QBase.

Python DB-API – A rapidly growing set of database interface modules for Python[9]. Currently are Informix, Interbase, MySQL, Oracle, Sybase and the generic ODBC interface supported.

Vendor dependant – Each vendor have their own way of moving HTML response (and some XML) into the database itself, basically allowing the database server to respond to webrequests, but these are more often than not very slow.

7.4 What databases are available?

7.5 What database should we use?

Recommendations:

JDBC support is gaining widespread use in general, meaning that java programs can interface very easily.

If the implementation language and platform is fixed (like perl on solaris, or visual C on Windows) look at the driver support available and choose the DB from that. For development purposes the Java + MySQL comination has worked well for me.

[7]. . . JDBC Data Access API. . . `http://www.javasoft.com/jdbc`
[9]. . . database interface modules for Python. . . `http://www.python.org/topics/database/modules.html`

7.6 Can users transparently harness the power of XML for webpublishing?

Hmmm looks familiar. Integeration wity DB instaed

Discuss how users can use their current tools to publish information to the web (Cactus principles: data capture and automatic conversion) by automatic conversion to XML and on-the-fly conversion on the way out of the webserver (while still having access to the original file provided by the user). This also allows people to view files made with software products not available to them.

7.7 Active and passive data acquisition for a database backed webserver

[just thoughts] User can “push” data to the system, or the system can actively watch a resource (directory, webpage, usenet server, “channels”) and update whenever new data is available. www.mind-it.com. RDF-format from netscape to push headlines.

7.8 Search engines and data mining

htdig (process website) swish (process files for website) are ok. With database you can ask the database directly (currently in blobs) but with more analysis of the files you can do better. “Which author wrote this back in 1948”.

Chapter 8

Overview of technology available for Linux February 2000

Describe what products there are currently available for these categories. Evaluate as much as possible. Conclude that currently this is an area in great development.

8.1 Webservers

Apache, misc Servlet Javaservers (jetty, thttpd, java webserver, etc). Look for comparisons. Roxen (graphs). Discuss whether several webservers should be run on the same machine to provide better services.
Look at the netcraft search - differentiate between Internet server and intranet server, and internal product stuff.

8.2 Search engines

8.3 Database engines

The number of database vendors which support Linux (which as for now is *the* Open Source operating system which people know about) is steadily increasing. This webpage try to keep track of them all[1], but I only list the major commercial vendors and Open Source projects, where the drivers should be of production quality. The following overview was created March 2000.

PostgresSQL (discuss features: transactions, unique numbers, in-kernel code, speed, availablity, multi-host capabilities?). Any flat file RDBS?
Greenspuns evaluation of embedded webservers in databases

[1]... This webpage try to keep track of them all... `http://linas.org/linux/db.html`

8.3.1 Cloudscape - Informix

The Cloudscape Database[2] is written in Java, with a JDBC and a HTML interface, has many core SQL and Java extensions implemented, and allow distribution amongst several Java-machines (using RMI). Informix bought Cloudscape Inc. in October 1999 to get this database system, so it is still very new. They offer a 60 day trial version from their website[3]. Pricing is approximately $900.

8.3.2 DB2 - IBM

DB2 for Linux[4] is available - the http://www6.software.ibm.com/dl/db2pde/db2pde-p[5] is available for free for non-commercial purposes.

DB2 XML Extender[6] is an interesting addition to DB2 which "let you store XML documents in DB2 databases and new functions that assist you in working with these structured documents.". It is available for AIX, Solaris and Windows NT.

8.3.3 Informix - Informix

http://www.informix.com/datablades/dbmodule/informix1.htm

8.3.4 Ingres - Ingres

The Ingress II database is in a beta stage for Linux[7], and can be downloaded for free. It uses Perl, gcc and Apache to provide webserver facilities.

8.3.5 mSQL - ?

Yes? Looook for stuff - rememebr to get the mSQL/MySQL book in the databse

8.3.6 MySQL - TCX

MySQL is an Open Source database[8] which is very popular amongst Perl programmers due to the good performance[9] combined with the free availability of source code and high quality drivers.

MySQL only costs money if you run it on a Microsoft operating system, or commercially on a server.

[2]. . . Cloudscape Database. . . `http://www.informix.com/informix/press/1999/dec99/cloudscape.htm`
[3]. . . a 60 day trial version from their website. . . `http://www.cloudscape.com/Evaluations/index.html`
[4]. . . DB2 for Linux. . . `http://www-4.ibm.com/software/data/db2/linux/`
[5]. . . http://www6.software.ibm.com/dl/db2pde/db2pde-p. . . `DB2 Personal Developer's Edition V6.1`
[6]. . . DB2 XML Extender. . . `http://www-4.ibm.com/software/data/db2/extenders/xmlext/`
[7]. . . Ingress II database is in a beta stage for Linux. . .
`http://www.cai.com/products/betas/ingres_linux/ingres_linux.htm`
[8]. . . MySQL is an Open Source database. . . `http://www.tcx.se`
[9]. . . the good performance. . . `http://www.tcx.se/benchmark.html`

8.3.7 Oracle

The complete Oracle 8i product range is available for Linux, with a low-end version available for download. The WebDB tool functions as a webserver-interface to any Oracle database.

When Oracle visited FLUG[10] we were promised that we could have all the Oracle software for Linux we needed (without official licenses), and that Oracle Denmark would provide the necessary support. It is interesting that Oracle Denmark unofficially seed the Linux communities with pirate copies of their software.

8.3.8 SyBase - Sybase

Sybase has Sybase Adaptive Server Enterprise and SQL Anywhere Studio[11] available for Linux for free as long as they are used for development. For production installations a license must be bought. An earlier version is unsupported but may be used without restrictions for production enviroments.

Their web integration tool does not appear to have been ported to Linux yet. The Linux database page[12] lists several third-party tools which provide web functionality.

Even though Microsoft does not provide Linux drivers for the Microsoft SQL-server, the freely available SyBase drivers for Linux appear to be usable.

8.3.9 Other approaches to database storage

Landbrugets rådgivningscenter[13] uses a Lightweight Directory Access Protocol[14] serverto provide easy access to commonly used web elements.

8.4 Browsers

MSIE5 (XML support, Solaris), Mozilla (alpha), Netscape (too old), HotJava 3.0 (no XML, can be expanded to do XML conversion internally?), Opera (?), Amaya (other W3C browsers?), AWTviewer in fop (can it do xml directly?),

Consider rewriting this chapter!

8.4.1 Netscape Communicator

The Netscape Communicator browser is available and fully supported for Linux. The version available at March 2000 was 4.72 in English only.

[10]...Oracle visited FLUG... `http://www.flug.dk/pages/foredrag/tidligere_foredrag/tidligere_foredrag.html`

[11]...has Sybase Adaptive Server Enterprise and SQL Anywhere Studio... `http://www.sybase.com/products/linux/`

[12]...Linux database page... `http://linas.org/linux/db.html`

[13]...Landbrugets rådgivningscenter... `http://www.lr.dk`

[14]...Lightweight Directory Access Protocol... `http://www3.innosoft.com/ldapworld/ldapfaq.html`

This browser have a reputation of being rather picky of its environment. Especially the Java-subsystem is prone to crashing the server. If Java is important, consider HotJava.

8.4.2 Mozilla - Netscape 6.0

The Mozilla browser[15] has been under development since 1998, where Netscape – inspired by the *the Cathedral and the Bazaar* paper – decided to switch development strategy for its incomplete source for Netscape Communicator 5.0 from an in-house system to a full-scale OpenSource model housed at http://www.mozilla.org[16].

In the past two years, the OpenSource developers have basically rewritten the Communicator from scratch into a truly impressive product (this is based on the M14 milestone release). A seperate beta-branch of the source code designated for the Netscape release was opened March 2000. A tentative release is midsummer 2000.

Mozilla is not capable of rendering XML in the client.

8.4.3 Opera

This Norwegian browser[17] has been highly successful on Windows for providing a small and quick browser which goes well with users who think that Netscape and Internet Explorer are too big and slow for their machines. Their Linux port is still in Alpha-testing, and unsuited for mainstream use.

8.4.4 HotJava 3.0

HotJava is a Netscape 3.0 compatible browser written in Java.

It is a bit slow, especially in screen updates, but has its force in its ability to execute Java applets (which is where Netscape is slow, and Microsoft does not comply with the standards).

HotJava requires a Sun JDK-based Java implementation (clones do not provide all the functions they need), and I have tested version 1.1.2, 1.1.4, and 3.0 under Linux, OS/2, Windows, Irix and Solaris, where it worked satisfactory.

There is currently no information regarding the status of parsing XML internally. Full access to the source is probably necessary before any third-party implementations come. In a perfect world, Sun would release the source, but that is probably very unlikely.

Rumours has it that development on this browser has ceased. I tried to extend version 1.1.4 to show TIFF files, but had to give up due to the miserable documentation for HotJava combined with lack of the source.

Other rumours has it that Sun will release the source for HotJava under their Community License (which basically makes the developers unpaid employees of Sun) like it has been done previously. On March 13, Sun released their first product under a true OpenSource license (the Forte product -

[15]...The Mozilla browser...`http://www.mozilla.org`
[16]...http://www.mozilla.org...`http://www.mozilla.org`
[17]...Norwegian browser...`http://www.opera.com`

previously Netbeans), which could indicate that they had evaluated their experience with previous releases, and found that nothing short of full OpenSource will do.

Remark: Today there are so many OpenSource projects that the truly talented coders can choose freely what they want to develop on, and they choose projects where they can get return in recognition from their equals, and where they don't feel that they slave for a company.

Sun has a "vote for the bug you want fixed"[18] system for Java, and the highest rated bug was "Port Java to Linux" for a *very* long time, and with four times the number of votes for number 2. The Blackdown Team[19] ported Java 1.1 to Linux, and collaborated with Sun on porting Java 1.2 to Linux. It took very long time, and when they finally had it officially published on the Sun Java webpages, Sun did not give credit to the Blackdown Team - not on the web pages, not in the code [rumours have it they even removed comments crediting Blackdown]. The Blackdown Team was highly offended, said publicly so, and stopped working on the port!

My personal guess is that this incident taught Sun that they must be very observant to the OpenSource culture they want to attract. It must be nurtured and credited in order to create the symbiosis that the Mozilla project has demonstrated to be possible.

In the same spirit SourceForge[20] are providing free hosting for OpenSource projects, and they have reached the point where people ask "why don't we put this on SourceForge to get CVS".

8.4.5 Microsoft Internet Explorer 5.0

Microsoft has created a very nice browser which works well on Windows platforms, and which supports a recent version of XML with the XSL-stylesheets (*reference - dette afsnit er simpelthen ikke korrekt)* . Since the XSLT standard was released January 2000, a release of MSIE50 can be expected "rather soon" which implements the XSLT standard, allowing the browser to render XML files on the client side. For now, server side processing is still the only option.

Microsoft Internet Explorer 5.0 is *not* available for Linux. It is, however, available for Solaris and HP-UX so the codebase is ported to Unix, and from there it is usually comparatively easy to get software to run under Linux. The Solaris version was incompatible with the window manager I was running under Irix, so I have not tested it much, but it was very, very similar to the Windows version.

My personal guess is that Microsoft holds back Linux versions of the Internet Explorer with Outlook Express until they see a fortunate opportunity to release it. A certain lawsuit springs to mind[21] - I do not expect Microsoft want to be accused of monopolizing the Linux market too.

To summarize: Microsoft Internet Explorer is not available for Linux at the time of writing.

[18]... "vote for the bug you want fixed"... `http://developer.java.sun.com/developer/bugParade/top25bugs.html`
[19]... Blackdown Team... `http://www.blackdown.org`
[20]... SourceForge... `http://www.sourceforge.net`
[21]... A certain lawsuit springs to mind... `http://www.idg.net/microsoft`

8.4.6 Amaya

Amaya is the W3C testbed for HTML-development. Is a nice browser combined with a reasonable editor. Version 2.4 do XHTML and HTML-4.0 with CSS but not XML yet.

8.4.7 StarOffice

$\boxed{\textit{What can we do here?}}$ - try it out.

StarOffice include a Netscape $\boxed{\textit{3.0}}$ compatible browser, with Java applet support. It is an integrated part of the StarOffice program, which besides Linux is available for Windows, OS/2 and Solaris.

As Sun recently bought StarDivision who had written StarOffice, the software currently being retargetet to Sun interests. It is not evident yet what that might be, but it is not unreasonable that the thin Java client (connecting to a StarOffice server on a Solaris machine) is what Sun bought it for.

Currently there is no sign for integrating XML support in StarOffice.

8.4.8 W3 - browsing inside Emacs

The W3-browser for Emacs[22] (needs XEmacs to show images) is a reasonably good browser, which has its force in its tight integration with Emacs, and is an interesting alternative for those who live in Emacs anyway. W3 is written in Lisp. The next major version is expected to be rewritten around a generic XML parsing engine.

8.4.9 Lynx - browsing in textmode

Lynx is one of the oldest browsers around still in widespread use, and does text-mode only. It is being steadily improved and hacked on, both because it is a nice browser for those who may not have a graphical login available, but also because it has a lot of options to help the programmer debug webservers and scripts. Has only recently been superseded in usefulness by the various tools in the Perl community.

$\boxed{\textit{Check if XML is being integrated in it? UNlikelyt}}$

8.4.10 Perltools?

$\boxed{\textit{?}}$

[22]...W3-browser for Emacs... `ftp://ftp.cs.indiana.edu/pub/elisp/w3/`

8.4.11 *OTHER BROWSERS?*

8.5 XML utilities

XT, Xerces, Cocoon, SQL2xml, validators (error recovering?), converters to XML (pod2docbook, tex4h, Perl with appropriate modules?, wordview modified)?
Show the XML-¿SQL-¿XML-¿HTML on the fly conversion possible with Cocoon .... Without auxillary code. All done in W3C style.
WordPerfect 9 should be able to edit SGML directly.

8.6 PDF utilities

PDF generated by pdflatex is good. PDF generated by fop isn't. createpdf.adobe.com pdfzone. acroread (ok, heavy), xpdf (font problems with bitmaps, ugly but fast, gives best result with usepackagetimes, no anti-alias), gv (pretty good). Distiller (check what it can do). Use pdfinfo to get the number of pages in the pdffile.
Conclude that it is possible to have a 100% Java solution for on-the-fly publishing XML to HTML and PDF through a webserver. Discuss why developers do it in Java, as opposed to any other language.
Very few utilites support creating PDF. Discuss how to create PDF.

Chapter 9

Konsensus – an unimplemented collaborative OpenFAQ system

A few years back I authored and maintained a Unix FAQ in the Danish Fidonet[1], where I collected a lot of information and wrote a lot too myself, which I then formatted and made available to the community for the common benefit.

The typical problem with a FAQ is that the maintainer must be quick to update with corrections when users take the time to submit them, and be certain to credit the users (to get the incentive for them to see their names listed in a community resource). The maintainer must also be sure to review the contents often to ensure that the information is up-to-date and correct.

This is a non-trivial task, and after basically giving up on the third rewriting of the UNIX-FAQ about halfway through, I concluded that for a FAQ to be truly successful a number of criteria must be met:

- *The information must be current and correct.* This is what makes people use the resource in the first place.
- *The bottleneck in this system is the maintainer.* The system should allow users to help you with updating the information in the FAQ!
- *The person providing the information must be given credit.* Credit and reputation is the mechanism that gets readers to submit updates and new information. Eric S. Raymond writes about the importance of reputation in the hacker gift culture[2] in the paper "Homesteading the Noosphere".

I was not the only one to reach such conclusions. Thomas Boutell developed the The OpenFAQ system[3] which divided the FAQ into a page per question with the corresponding answer. If a reader was better informed regarding the answer, a link was provided to a fill-out form where the updated information could be entered. The finished form was then saved in the system, and an email sent

[1]. . . a Unix FAQ in the Danish Fidonet. . . `http://www.fido.dk/faq`

[2]. . . the importance of reputation in the hacker gift culture. . .
`http://www.tuxedo.org/ esr/writings/homesteading/homesteading-8.html`

[3]. . . The OpenFAQ system. . . `http://www.boutell.com/openfaq/source`

to the FAQ maintainer so that he could check that the new text was suitable for updating the old text. It was the maintainers responsibility to maintain the questions.

Even though this approach gives the maintainer an excellent method of organizing responses from users, it does not help much in the viewpoint of the submitting reader if the maintainer takes too long to accept the submission. It took three weeks for my personal submission to be accepted, which was plainly too long.

It is very understandable that the moderator wants to see the new submissions before activating them, in order to avoid electronic grafitti and vandalism, but that requires a high degree of visibility of the moderator. The site is not capable of running without supervision.

The answer to this is to give the users access to post (after they have identified themselves) *and* then deal with malicious posters when it happens. It appears that requiring the user to specify a valid email address, and get login information by email, is sufficient to weed out most of the vandals.

Recognizing this I outlined the "Konsensus" system which was intended to work in combination with a usenet autoposter for one or more groups. I have been involved in starting the autoposter system in the Danish newsgroup, where a given newsgroup is under surveillance by a daemon that looks at the email address in the "From"-field and checks with a database whether that address is "new" (has posted within the last 3 months or so). If not, an email is sent to that address with a canned message (like the "dk.edb.programmering.perl" message[4] which I wrote (the Perl group has many new Perl programmers since many Internet Service Providers provide Perl as their CGI-scripting language). This proved to be extremely efficient, since the regular posters *knew* that the newcomers would be greeted with a welcoming post containing the information they needed, and efforts could be saved for dealing with more complicated issues.

Such a welcoming message suffers from exactly the same problem as a FAQ, namely that it requires regular attention in order to be as useful as intended by the author, so I decided there was a need for a system which would allow the users of the newsgroup to collaborate on making the welcoming message as useful as possible until they reached a consensus (hence the name).

Basically this would involve the following:

- The maintainer were responsible for creating, modifying and deleting the questions – as these were intended to be rather static – after discussing this either in the news group or via email
- The users were responsible for changing the content of each page, by modifying the HTML-source of the answer part of each question.
- In addition to specifying the actual content of the individual answers, the online edition were to allow comments, where readers could provide additional information to elaborate on the various items if the reader needed additional information.
- Each modification would be checked in to RCS, allowing any reader to *go back in time* to see what earlier answers held instead of just having the latest edition which might not be 100 percent correct. These editions would be labeled with the email address of the latest editor.
- The web pages would be generated automatically whenever the underlying files were changed.

[4]... "dk.edb.programmering.perl" message...
`http://www.usenet.dk/oss/dk.edb.programmering.perl/intro.txt`

This were at the time where I was evaluating which language I should use for implementing Cactus (see section 10 on page 87) so I followed both the Java group and the Perl group carefully while doing so, which in particular prompted me to look for such software. The problem is that in the same way that the Perl groups are bothered by new programmers which ask questions about webservers (and not about perl) and therefore belongs in another group, the Java groups are bothered by new programmers which ask questions about JavaScript which is considered HTML-programming which also belongs in another group.

To make a long story short, I then discovered the FAQ-O-Matic[5] which does most of what I wanted the Konsensus system to do (except for the RCS storage with credit), and which runs as a set of CGI scripts. I installed it under Linux, played with it for a while, and found it to be quite suitable for the job. Combined with a commentary service as the one provided by ArsDigital, this would provide a complete implementation of the facilities I needed for Konsensus. I set up a FAQ-O-Matic for the Danish Perl group (based on the existing automailer in April 2000 as a test). If this any-body-can-update-information concept work well, I will write a script that updates the generated mail from the submitted entries, allowing this document to be entirely user-supported without the need of a maintainer.

[5]...FAQ-O-Matic...`http://www.dartmouth.edu/ jonh/ff-serve/cache/1.html`

Chapter 10

Cactus - a web based document publication and data-driven conversion framework

In the previous chapters I have talked a lot about web publishing and how the computers can help the authors by letting them work with tools they know at the writing tasks they know.

The *Cactus* system is my "proof-of-concept" implementation of a system which can do document conversion and publication without any intrinsic knowledge of the data in question, and in a way which is basically transparent to the users by letting them use it as a simple service where they use the tools they know.

This chapter explain the architecture of Cactus, chapter 11 on page 97 list a small set of external filters I have implemented to provide $\boxed{....}$, and chapter 12 on page 99 discuss a large number of possible filters I have been investigating.

The goals for Cactus were:

- Publishing documents must be as *easy as possible* for authors
- The publishing process should be *fully automatic*, without the need for human web masters
- The software base required from the audience should be *as small as possible*
- The system must run under *Linux*
- The amount of intrinsic knowledge hard-wired into the system should be as small as possible

The above mentioned goals have been fullfilled to my personal satisfaction even though the program turned out differently than I originally envisioned. Cactus is today a system which

- Accepts electronic documents from many sources
- Applies filters repeatedly to convert each document to as many other forms as possible
- Allows easy access to the documents through a web browser
- Provides a public service – all information is available to everyone

- Allows documents to expire if they are not intended for long term storage

Due to time constraints I have not implemented as much of the Cactus system as I would have liked to. ...

10.1 Philosophy

The Cactus philosophy is related to the pbmplus software package from Jef Poskanzer[1] which is a very powerful image manipulation package, based on three *very* simple image formats (namely PBM, PGM and PPM for binary, gray-scale and color images respectively). The formats are understood by a lot of filters manipulating these images (these may be piped together to combine effects), and a number of conversion programs to and from a large set of common graphics formats. This approach has proven to be very successful indeed, and Cactus functions as the arbiter for applying such *filters* to *items* in order to do the conversions and derivations to make the files that the users require.

Cactus works with items which are considered to be data streams with attributes owned by somebody. Cactus itself is only working with the data as a file containing raw data with a MIME-type, and calls externally defined filters (may be any kind of program that converts an input file to an output file) to interpret and manipulate the information inside as appropriate.

The database tables used by Cactus provide a number of attributes which may be interpreted and set by the various filters. This allows the *presenter* filters to provide better and more informative views of the items.

By using an underlying database for data storage it is possible to have several programs simultaneously do acquisition, conversion, derivation, compacting and archiving of data while presenting data as web pages responding to user needs. Experiments have shown that a modern home PC is sufficient to run Cactus for a small work group with a limited set of filters.

10.2 Cactus seen from the users perspective

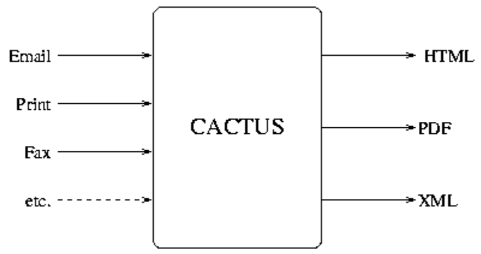


Figure 10.1: The basic idea behind Cactus

For a user, Cactus is not difficult, since she uses her usual tools to interface with Cactus. A normal document publication would be like this:

[1]...pbmplus software package from Jef Poskanzer... `http://www.acme.com/software/pbmplus/`

1. Submit the document to Cactus by one of many ways, like sending it as an attachment to an email or by printing it to a virtual printer
2. Wait a few minutes
3. Go to the Cactus home page and locate the upload. There is a list of recent uploads pr user, and a quick list listing the very most recent uploads. [An unimplemented option is to email the user when the requested document conversions have been completed]
4. Find the versions needed of each item. URLs to items can be mailed to others for them to view, and other versions of items can be downloaded if so needed.

The users view of Cactus is shown in figure 10.1 on the facing page.

10.3 Design

The database tables are the heart of the Cactus system. In addition to these, a large number of programs work on the data in the tables in order to do the various jobs needed when working with a potentially *very* large dataset. I have identified the following types of programs which should be running all the time (and sleeping when they have nothing to do):

Validators – validates whether the content of the item is conforming to the MIME-type provided and the data valid. If not, a MIME-type is synthesized according to filename and content.

Extractor – looks *inside* an item to look for references, embedded files and other kinds of extractable data. A reference may be an URL or an emailaddress in a signature. Embedded data could be an uuencoded image in a Usenet posting.

Converters – create another version of a given item *fast*. These are intended for conversions of data with users waiting, like `pnm` to *uncompressed* `png`. The generated versions are stored in the database and compressed when the system is otherwise idle.

Derivers – create another item from one or more originals irreversibly, like generate `dvi`-files from one or more `tex`-files.

Compressors – Reduce the size of a converted item if possible, by reencoding the data using any built-in compression schemes in the data format of the item. Sample formats are `png` and `tiff`.

Optimizers –

Archivers – moves compressed items which has not been used for a long time into long-term storage, leaving only the metadata. This keeps the working data set small.

These are complemented by

Acquirers – gathers items from ”outside” Cactus and enters it in the ”incoming” table with an appropriate MIME-type, an expiration date, and *misc user information*. An acquirer could accept email, emulate a printer, retrieve Usenet articles, etc.

Janitors - cleans up whenever the system is otherwise idle, or when the cache is full. Converted items can be emptied of content, originals can be archived. Expired items can be purged completely from the database, along with all their derived items.

Presenters - extracts data, and present them. This could be a CGI-script presenting a given item as a http-stream. A PDF item could be presented as a window with two frames, the leftmost containing a thumbnail pr page, and the rightmost a high resolution version of a given page.

In an ideal world we have infinite storage and infinite CPU-speed, meaning that everything would be done instantly when we need it. Since that is clearly not possible, a pricing system is implemented for derivers which are scheduled based on expected CPU-time used for this derivation. The "cheapest" derivations are then done first, leaving very expensive ones for times when the system is otherwise idle.

None of this information is hardwired into Cactus. All filters are stored in appropriate tables, as listed with the individual types. A full description is present in the implementation section.

10.3.1 Data acquisition

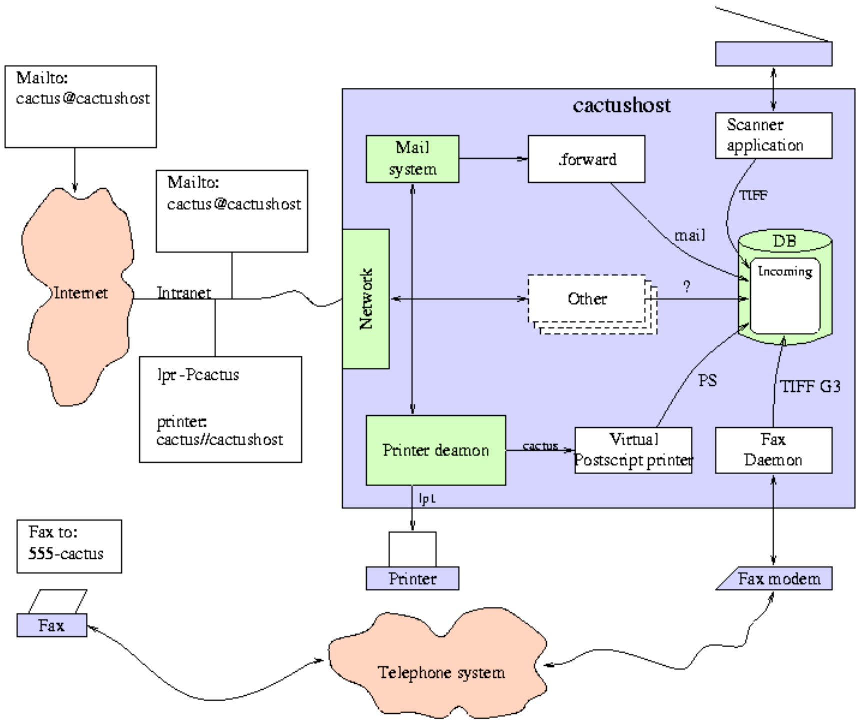


Figure 10.2: The data pathways into the `incoming` table in Cactus.

Figure 10.2 shows some of the possible ways to acquire data into Cactus. I have envisioned several ways to submit data to Cactus:

Email – a user may easily submit any file, by attaching it to an email which is sent to `cactus@asserballe`.

Virtual Printer – Cactus provides a virtual PostScript printer accessible from both Windows and Unix, which acceps printjobs that are inserted directly in the database instead of being printed out.

Scanner – A scanner may produce a set of images which belong together.

A drag-and-drop directory – A Windows *file share* is provided in which users may put any file that is then moved in the database by a periodic surveilling daemon.

Fax – A faxmodem may be connected and used to gather faxes. Information from the modem regarding phone number of the sender (as well as any other provided by the phone company) may be recorded along with the actual bitmap. The bitmap may be printed directly, in order to emulate a regular fax. If OCR-filters are installed, a text version may be generated which may be integrated in a full text search facility

Voice mail – The same faxmodem may be used as an answering machine, and the recorded messages stored in Cactus as a WAV file. These could with great benefit be converted to MP3 files and mailed to people.

Each of these must fill out as many fields in the `incoming` table as possible.

Note: Only the email and printer data acquirers have been implemented in the proof-of-concept implementation.

10.3.2 Examining an item

Cactus in itself have no notion of an item besides it being a raw bitstream. By providing an examiner for a given MIME-type, Cactus can be made aware of whatever is inside an item. An examiner must do two things, and print the findings back on `stdout` as lines of (MIME-type, space, value):

1. Provide information about the item in question and report it back to Cactus in order to have the attributes set in the database. Each attribute should be listed as the MIM "`x-cactus/`*attribute*" followed by a value

2. Provide a full list of contained items within. These should be listed with a value that uniquely allows the corresponding extractor to identify the item in question when given the MIME-type and the value.

Examiners are found in the `examiners` table (see table **??** on page ??. Figure 10.3 on the next page show how an examiner examins an item and return a list of the contents, Cactus updates the item in question and forward a list of any contained objects to the appropriate deriver which then in turn extracts the items the examiner found and inserts them in the database flagged for later examination.

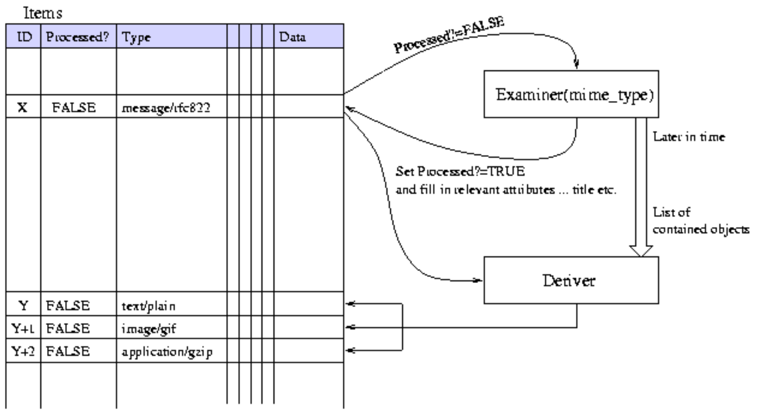


Figure 10.3: How the examiner update an item, and provide information to the deriver

10.3.3 Converters

10.3.4 Derivers

10.3.5 Compressors

10.3.6 Optimizers

10.3.7 Archivers

10.3.8 Janitors

10.3.9 Presenters

Currently Cactus does not provide presenting data based on MIME-type. There are the following ways to extract information and show it to the users through a webbrowser:

- An individual item. The URL `http://asserballe/ cactus/get.php?id=XX` returns the given item with the correct MIME-type. Additionally the URL `http://asserballe/ cactus/get.php?md5=XX` is recognized which is persistent across database rebuilds.
- Through JDBC to any Java program, like the SQL-processor in Cocoon which returns a complete XML-representation of query results, which can then be manipulated with XSLT operations. This is used to generate all XML-based information in the proof-of-concept implementation.
- Directly to PHP3, on a line-by-line basis.
- To the DBI::MySQL module in Perl, also on a line-by-line basis.

The easiest way to do this is as XML (with the addtional overhead of keeping the whole result in memory), since the programmer does not have to bother with iterating though the returned table.

Tjah, jeg er naaet hertil :-)

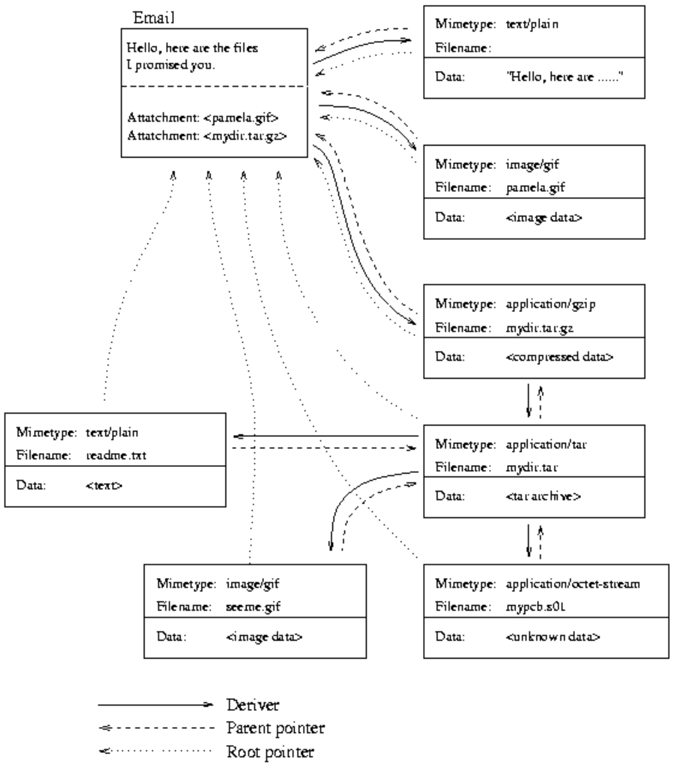


Figure 10.4: The processing of an email with a GIF and a compressed tar-archive attached

Hello done.

10.4 Overall view

10.5 Data flow

10.5.1 Input sources

10.5.2 Output formats

10.6 Daemons and SQL tables

10.7 Implementation

The `~cactus/.procmailrc` contains

```
MAILDIR=work
LOGFILE = _logfile
VERBOSE = yes
FROM=‘formail -rt -xTo:‘

:0 wc
| (cd /home/cactus/src/cactus; ./enter-document.pl message/rfc822 $FROM )

:0
incoming-mail
```

which means that the

10.7.1 Hardware/software considerations

10.7.2 Data acquisition

Email

Printing

Fax

Scanning

Voice mail

10.7.3 Derivers

10.7.4 Converters

10.7.5 Examiners

10.7.6 ...

10.7.7 Database

10.7.8 Web server

10.7.9 Making it all work under Linux

Source code

If I get the time I will make this a literate form. Perl modules with the literate programing extension?

10.8 Why the name “Cactus”?

10.9 The design of Cactus

I set the following goals for Cactus:

Cactus were two projects

Today there are very few working procedures which are potentially available to any computer user at a very low cost:

- Printing a document
- Email an attachment to another user
- Fax a document
- Scan an image
- Receive a voice mail

All of the above can easily be done with a computer as the recipient, allowing it to capture the data sent. The format preserving the most information is email with an attached file, as the others use a visual or audible representation of the original data.

10.10 Overview

My idea with Cactus was from the start to create a web-based facility which would ease the process of publishing and sharing information via the Internet.

[The short comings of Yggdrasil]

I set the following goals for Cactus: * Publishing documents must be as easy as possible for authors * The publishing process should be fully automatic, without the need for human web masters * The software requirements of the audience should be as small as possible * The system must run under Linux

These goals have been fullfilled to my personal satisfaction even though the program turned out differently than I originally envisioned. Cactus is today a system which * accepts electronic documents from many sources * applies filters repeatedly to convert each document to as many other forms as possible * allows easy access to the documents through a webbrowser * provides a public service - all information is available to everyone * allows documents to expire if they are not intended for long term storage

Chapter 11

Cactus - sample set of filters

Chapter 12

Cactus - possible candidates for filters

ypx is from comp.sources.misc volume 40 http://ftp.lth.se/archive/usenet/comp.sources.misc/volume40/ypx/

compiles on solaris with ”-lsocket -lnls”. does not compile on linux

12.1 Background

A major computer problem not solved satisfactory yet, is the ability for users to share information in spite of them using different hardware and software. Even though the modern use of the Internet allow any two users to exchange files as attachments to email without any furter ado, it has not helped much in actually *interpreting* the contents of these files.

If a given user needs to use a given document, she needs software specific to the document format to interpret it. If another user needs the document, she needs software too.

The most prominent word processor document format today is the Microsoft Word format, which is so complicated that even Microsoft has trouble with it (hence the quotation of this chapter). Basically you need Word to use this format – all other solutions provide inferiour results. If you only need to read and print documents Microsoft provides a Word Viewer program for this, which require Microsoft Windows.

Unfortunately, this is no help for those users who does not have Word, notably Linux users.

presentation and internet thingie

The Cactus system is an Open Source document conversion and presentation framework, which allows users to submit documents to a central server, which then uses a set of stored conversion filters to process the documents into other forms, notably HTML and other Internet formats.

The version of Cactus described herein, have the following features:

- Convert documents to HTML versions viewable in a browser
- Windows and Unix clients can use Cactus as a virtual PostScript printer with automatic PDF conversion, and may download the result directly or view the HTML-conversion (making this accessible to platforms without Acrobat Reader).

- Images can be resized and converted to a number of formats like JPEG, TIFF, and PNG.
- Extensible and configurabele. New filters can be installed to convert from one MIME-type to another.

This runs on a single Linux machine, but any number of Unix machines may requests jobs to do if more CPU-power is needed. *The necessary filters and a simple Java client must be installed on each. Will the*

ingen dokumentstandard. ingen wp-producenter der underst'tter det, ingen interessei at goere noget ved det,

12.1.1 System description

squid: http://www.squid-cache.org/

Cactus is a sample implementation of the following issues:

- **easy publishing** - users can publish via email/fax/print/watch usenet/www which is processed into the SQL database, and confirmed.
- **automatic conversion** - system will automatically convert a given document to what the user can see (or want). Browser sends a capability string with each request - use that to provide stuff directly, or generate a "this is available" summary.
- **automated navigational framework** - each document has an annotation which tells Cactus where to place it in the navigational hierachy. The corresponding navigational pages are automatically generated and updated when new documents arrive. This also ensures system integrity without "broken links".

Implementation languages - possibly Java or Perl. Perl chosen due to better library support (with source). Rex thingie in Java. How can MIME, TAR, etc be done in Perl (remember jubilations!). Using Apache with Cocoon (perhaps) and MySQL. Graph algoritms in [Sedgewick(1990)].
Good summary from 19991114. Speed of PNG gzipping? PNG uncompressed intiially and then later compressed by pngcrunch.

http://photo.net/wtr/word.html Utilities: mswordview, xls2xml

12.1.2 Background

Explain the history. Yggdrasil to address the need of an automatic webmaster → analysis (seperate file written earlier). Cactus to address the difficulties of publishing information to Yggdrasil.

Explain Cactus as a successor to Yggdrasil.

12.1.3 Installation

Perlmodules. MySQL server. SAMBA.

[Rephrase following paragraph]

I have concluded the following goals for CACTUS, in order to avoid repeating history:

1. ”Ease of publishing” is crucial.
2. Document preparations and transformations must be fully automatic.
3. The organisation using the system must be fully doing so.

”Ease of publishing” is crucial

CACTUS uses two approaches in order to make publishing as easy as possible for the users, namely

Documents are accepted in their native format, and in numerous ways.

Discussions with potential users showed [...] to be realistic ways in which CACTUS would be used:

As a document storage for various versions of a document, being developed and emailed back and forth between authors and peers. By allowing CACTUS to accept documents as email attachments, it would be very easy to enter each draft in CACTUS by including it on the list of authors or peers. In this way the archival of the document in CACTUS is completely transparent. As a fax machine. When replacing a fax machine with a computer, it is very easy to send a copy of the temporary image of the fax to CACTUS, before printing it. The fax system is then enhanced with the possibilities of the web, relieving the need for the physical copy. As a printer. Cactus provides a ”printer”, which makes a web-version out of any document the users can print. Users can then share final versions of documents without requiring the recipients to have the software in question. Either the Adobe Acrobat Reader can be used, or the primitive multiple image viewer in Cactus.

The full list of the publishing methods in Cactus is listed [....]

The users were primarily expecting to use these file formats:

- Word DOC and RTF files. These are the storage formats of the Microsoft word processor Word.
- HTML files. The common format for the web.
- LaTeX files. This typesetting program is very popular with mathematically oriented academics.
- GIF, JPEG, TIFF and PNG images. These and many more are in common use. Cactus use PNG internally [except possibly for JPEG]. The full list of supported formats is listed in the implementation section [see somewhere].
- PostScript and PDF files.
- Fax images.

Document preparation and conversion must be fully automatic

There was a strong consensus amongst the users, that it would be very nice not to have to convert the documents manually every time they were to publish a document. Therefore Cactus accepts several document formats as described above, and abandons the requirement that the users should convert their documents to HTML before publishing.

The system have some very different views on the documents depending on which representation the user needs - not all make sense for all documents:

1. The unaltered original. This file is always available, allowing users with the correct software to continue working with it.
2. A normalised version of a document. This could be a PNG version of a TIFF image or BMP image which can be viewed by all modern browsers, and an XML version of a document. If a suitable encoding can be found, even images and sound can be represented in XML so that this does not overlap the textual representation.
3. A visual representation of the original. This is the ”look of the file”, i.e. as it would look when printed to paper, and allows users without the corresponding software to view and print the contents.
4. A textual representation of the original. This is the ”meaning of the file”, which could say that the line ”foo bar” is a level 2 heading, providing search capability.
5. An audio representation of a ”document”. This is e.g. an message left on an answering machine.

Conversions between all these document representation must be automated as much as possible. [write about the Cactus framework for providing existing and future filter types]. The normalised document is the only one created when a document enters Cactus - the rest are created on demand to avoid overfilling the underlying database, but cached for a reasonable time to improve performance.

[....]

The organisation using the system must be doing fully so.[rephrase]

In order for such a system to be successfully used within an organisation, it is vital that the organisation is using it whole-heartedly. [and more of the same].

Note on derivers:

A deriver is a pipeline which derives another version of an item. E.g. PNG to GIF,

12.1.4 The implementation of Cactus.

Goals:

- Stable, well-known and remotely administrative platform- Linux/Solaris
- Platform independence - the resulting system must not be tied to Unix
- Modular - in order to minimise actual development, software libraries should be used as much as possible.
- Extensible - it should be easy for the system administrator to enhance functionality.

Choosing an implementation language:

Since platform independence was important for the system, it was quickly found that reasonable choices would include scripting languages plus Java. Scripting languages are fully interpreted allowing a script to run on numerous platforms without change. Java is the only compiled language with this property, due to the ”Write once, Run everywhere” philosophy from Sun, plus the tight integration with Internet technologies in Java.

Initially I wanted to write the core of Cactus in Java, and spent a couple of weeks evaluating the language but found that

The mentality of the Java-community on the Internet, is very influenced by the shareware philosophy typical for the PC-user. Everything useful cost money, source code is not revealed, and the general tendency is for large, stand-alone applications.

Nobody had written a publicly available, stand-alone MIME-parser in Java. Several RFC's should be implemented and tested, in order to get email-parsing in Cactus.

The general level of abstraction is - in my opinion - too low in Java, while the ”core language” is enormous.

[?]

Information extraction from items.

A given document contains one or more items, which again may contain further information. Cactus uses the MIME-type of a given item, to select the information scanning method with a fall back to the generic application/octet-stream examiner.

The application/octet-stream is parsed for:

The text is scanned for URL?s, either with the Tom Christensen urlify or the program posted or commented by Abigail. These include ftp, http, gopher, mailto and news references. These are stored as external references.

The text/plain is parsed for

uuencoded data in a text stream. These start with ?begin ### name?, contain lines matching ####, and ends with ?end?. Such a file is then assigned a validated MIME-type based on the name of the file, and entered in the system as a derived item.

An text/html item is parsed for:

normal references in <a>-anchors. While doing this, the text to be rendered is extraced and parsed as a text/plain stream, in order to get the sequence right (may be changed). The <meta> tags are examined in order to extract keywords for the label.

12.1.5 Converters

Microsoft Word

The Microsoft Word DOC-format is widely used, but apparently so hard to use that even Microsoft have trouble doing it correctly, and the reason why Cactus was started in the first place. I have spent a great deal of time looking for suitable software which could have been used with Cactus on the server side, and then testing it out. $\boxed{\textit{WINE}}$

Microsoft Word – The best solution would be able to actually run Word itself, but apparently this is integrated so well with Windows that it cannot run on other systems like WINE[1]. Since Microsoft previously have deliberately made the Windows 3.1 version of Internet Explorer 4 unable to run in the WinOS/2 subsystem for OS/2, I strongly suspect that this is also the case here.

Viewer for Microsoft Word – the Word Viewer is available for all Microsoft operating systems. The 16 bit version is reported by Usenet posters to behave reasonable in WINE, and would be a good candidate for producing PostScript printouts of Word documents. It would require a full Windows application to automate this, since it does not support command line arguments.

Corel WordPerfect 8 for Linux - This is generally a very nice word processor, but the import filter for Word crashed WP when I tried it with a large document. Filters should be better in WordPerfect 9, which is due for Linux medio 2000.

StarOffice 5.1 – the German office suite for several platforms have excellent filters for importing Office files in general, but cannot be automated from the command line. A StarBasic program must be written and invoked to do the job – a request on the StarOffice newsgroups for a tool to do this, did not get any response. Due to lack of time I did not pursue this further.

Since I did the testing, Sun have bought StarOffice - most likely since they need an network based office suite for Java, which StarOffice provide with thin clients and a Solaris based server - and promise to make the source generally available. This has not happened yet, but it is currently the most likely alternative to the Microsoft Office solution.

The StarOffice package is such a popular package that Sun has a download counter on their main home page (which said 1,962,277 downloads as of 2000-04-03).

AbiWord – An Open Source word processor which is part of the Gnome Office suite. It was not able to parse my test documents in Word.

wv (previously mswordview) – an Open Source Word to HTML converter which currently do text well, but have trouble with graphs which are converted to the WMF format.

Do-It-Yourself - If you ask Microsoft very nicely, you can get a copy of the Microsoft Office Binary File Format Specification, and write a personal parser of Word. It took Microsoft 6 months to answer my email request, and then they just sent me the license twice with no specification. Second time they got it right. Unfortunately, the license was so restrictive that I decided not even to *look* at the specification.

Majix - This small RTF to XML converter written in Java, can also parse DOC files when running in the Microsoft Java Machine by invoking Word to save the DOC file as RTF. (Unfortunately it had serious problems with RTF files generated by other programs). It was easy to customize it to generate DocBook XML.

If Majix could be called from Cactus with a DOC-file and return the corresponding XML file a major goal had been accomplished. A test scenario was therefore made with a remote telnet session to a NT-machine, where Majix was invoked from the command line. The test failed as

[1]...WINE...`http://www.winehq.org`

Word hung in the start up phase – my personal guess is that Word needs access to the GUI. An installation on a MIP server was not possible due to MIP system policy.

Additionally the company has not released a new version in a year, and the license explicitely prohibits disassembly.

My intermediate solution has been to adapt the mswordview to output DocBook XML instead of HTML, which provides a text-only display.

TEX and LATEX

tex4h

12.1.6 MySQL

On asserballe

```
create table incoming (when datetime, mime varchar(80), user
  varchar(80), how varchar(80), filename varchar(255), comment varchar(80),
  item longblob);

+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| when     | datetime     | YES  |     | NULL    |       |
| mime     | varchar(80)  | YES  |     | NULL    |       |
| user     | varchar(80)  | YES  |     | NULL    |       |
| how      | varchar(80)  | YES  |     | NULL    |       |
| filename | varchar(255) | YES  |     | NULL    |       |
| comment  | varchar(80)  | YES  |     | NULL    |       |
| item     | longblob     | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

- **wmf-anything**: libwmf (no fonts),

  KVEC[2] (could not render the wmf files from wv),

  WMF docs[3]

[2]...KVEC...http://ourworld.compuserve.com/homepages/kkuhl/

[3]...WMF docs...http://www.companionsoftware.com/PR/WMRC/WindowsMetafileFaq.html

12.2 Programs

12.2.1 Linux - an operating system

Linux was chosen as the development platform for these reasons:

- High degree of familiarity with the operating system
- Almost all Open Source software run "out-of-the-box" on Linux
- High performance Java Development Kit available from IBM
- Remotely maintainable via X/telnet/ssh.
- Freely available from the Internet.
- Root access possible without interfering with MIP security rules

These rules were also most likely fullfilled by any of FreeBSD/NetBSD/OpenBSD/Solaris x86, but I had already burned Redhat 6.1 on a CD for my portable, so there was no need to look any further. Linux has been a very satisfactory choice.

12.2.2 Apache - a high performance web server

There have probably been written more webservers than editors in the world of today. My requirements were simple:

- Run Perl programs efficiently
- Run Java servlets efficiently
- Have a large user base in order to ensure program availability

This basically left a single web-server, namely Apache[4], which is capable of running Perl CGI scripts very efficient with the `mod_perl` module[5] (which uses a resident Perl interpreter combined with caching of the bytecode of previously encountered programs).

If requirements were less, other webservers might have been applicable. The Jetty[6] (Java Webserver capable of running servlets), Acme.Server[7] and Roxen[8] (standard webserver with excellent graphics capabilities) webservers are just a few candidates for tasks with a less diverse need for scripting languages. Servlet support is getting very common place.

[4]...Apache...`http://www.apache.org`
[5]...the `mod_perl` module...`http://perl.apache.org/`
[6]...Jetty...`http://www.mortbay.com/software/Jetty.html`
[7]...Acme.Server...`http://www.acme.com/java/software/Acme.Serve.Serve.html`
[8]...Roxen...`http://www.roxen.com`

12.2.3 Squid - a high performance web cache

The http-accellerator mode[9]

12.2.4 Cocoon

12.3 Filters

12.3.1 ps to pdf

Adobe Distiller[10] – The reference PostScript to PDF converter.

Ghostscript[11] – This is a PostScript interpreter which has gained wide usage.

$\boxed{??}$ rtf2latex http://www.tex.ac.uk/tex-archive/support/rtf2latex2e/

papirbjergene ImageTag, 3m xerox

12.4 Scanning in mail

$\boxed{\textit{ELSAM}}$

[9]...The http-accellerator mode...
http://www.squid-cache.org/Doc/FAQ/FAQ-20.html#what-is-httpd-accelerator

Chapter 13

Conclusion

13.1 Conclusion

Databases are great, yea! Webservers are great, yea! Things must be powerful and easy for best results.

should this go away?

.1 TeX and LaTeX

which is well renowned for its excellence in general. For those who want a higher layer of abstraction from the raw, untamed TeX the LaTeX for

Appendix A

Report writing tools

When writing a large document like a thesis, you get to appreciate the power of the tools available to you in the Open Source community. This chapter outlines which tools I have used while writing and my experiences with them, with the thought that they might be useful to others later, and Open Source is all about not reinventing the wheel if it can be avoided.

All the software listed here, is available with a standard Linux Redhat 6.1 installation unless noted otherwise in the text.

A.1 LaTeX

Guide is [Kopka and Daly(1999)].

A.1.1 Creating PDF files

I have had good success with the following approach to creating PDF-files from a LaTeX-file.

- Use a PostScript font! Many LaTeX-installations use the standard TeX-font 'Computer Modern' as a bitmap font, which doesn't look good as a PDF-file at other resolutions. Try `\usepackage{times}` or `\usepackage{bookman}` to see if that helps.
- Use ""`pdflatex`"" to generate the PDF-file directly. Included graphics may have to be converted to png instead of eps. The traditional approach with "`dvips`" and "`pstopdf`" *from Ghostscript*
- Develop the document to work with both "`latex`" and "`pdflatex`". The edit-compile-view cycle is much faster with "`latex`" and "`xdvi`" than with "`pdflatex`" and "`acroread`".

I have had problems with printing the PostScript file from "`dvips`" directly to some HP-printers. Results have been much better with printing from Acrobat Reader[1] (can be done directly from the command-line with "`cat source.pdf | acroread -toPostScript | lpr`" – use the `-helpall` option to get a full overview).

[1]...Acrobat Reader... *www.adobe.com*

A.1.2 Including screen dumps

Screen dumps have been captured with xv[2] (which is now 6 years old and still unbeat) and saved as `tiff` files. These were then converted to `png` with "`pnmtopng`" and to `eps` with the NetPBM package[3], which has worked flawlessly. It is important to remember to use `pnmtops -noturn rle` to get the correct orientation and a reasonable compression (25 files shrank from 52 Mb to 9 Mb total).

The \includegraphic (from the `graphicx` package) can read `png`-files from within `pdflatex`, and `eps`-files from within `latex`

The process of embedding Postscript is fully described in [Goosens et al.(1997)Goosens, Rahtz, and Mittelbach

I have had "`xdvi`" active while writing, and if the command "`latex mydocument && killall -USR1 xdvi`" is used, xdvi will automatically update after a successful compilation of the TeX-document. That is very nice when your editing have happened within a single physical page.

For presentation purposes, *Acroread with packages found at the DK-TUG website at aucdk*

A.1.3 Finding back to the source document from xdvi

When reading the typeset document (either in "`xdvi`", "`acroread`" or on paper) it is often a problem to find the appropriate `tex`-file. I used this trick which was deduced from the *appropriate latex book here Daly*
which puts the filename to the left in the running header on each page.

```
\usepackage{fancyhdr}           % provide headers where I can put the filenames.
% Trick from page 16 in fancyhdr documentation
\newcommand{\currentinputfile}{}
\newcommand{\myinclude}[1]{%
  \renewcommand{\currentinputfile}{File:\texttt{#1}}\include{#1}%
  \renewcommand{\currentinputfile}{}}

\lhead{\currentinputfile}
\pagestyle{fancy}

...
\begin{document}
\myinclude{s-front-matter}
\myinclude{s-introduction}
...
```

Modified xdvik!

[2]...xv...`http://www.freshmeat.net/appindex/1998/07/28/901622941.html`
[3]...NetPBM package...`http://www.freshmeat.net/appindex/1999/06/24/930238715.html`

A.2 Emacs

Emacs is a very well known editor in the Unix-world where its extensability allows it to get new functionality by installing additional modules. The standard distribution of Emacs provides:

Ispell – The M-Tab command (bound to `TeX-complete-symbol`) runs `ispell-complete-word` if the current word is not a TEX-command, which suggest possible words starting with the word under point. The M-$ command (bound to `ispell-word`) is a great help when in doubt about a word. The *flyspell-mode* highlights words while you type, akin to the wavy underline used in MS-Word.

CVS – Emacs supports CVS[4] directly. I had my working directory on my portable, and my CVS repository on the MIP system, in order for my CVS files to be automatically backed up.

Compile – The `M-x compile` works well with Jade, nsgml, Java and other tools where you may encounter errors in your source. `Ctrl-x ‘` makes Emacs go to the line with the error.

Additionally I have installed:

AUC-TeX – provides a complete IDE for writing LATEX documents, including keyboard shortcuts, a debugger, and intelligent commands depending on the state of the files. *url? somewhere at sunsite*. Highly recommended.

psgmls – provides easy entering of tags, and parses DTD’s to verify tags, and describe which tags are open at a given cursor location. Good for writing XML and SGML directly. The psgml homepage[5] - *consider writing a set of AUC-TeX compatible keybindings for DocBook*

A.2.1 Making them all work together

Using these packages has basically been very easy, but occasionally they step on each others toes. That could normally be circumvented:

TEX and RCS – The RCS header fields, contain dollar-signs which causes TEX to enter/leave math-mode (like Id), causing it to be rendered in math italic $whichisnotsuitableforprose$. By typing it as `$ $Id$ $` it is possible to avoid this. The dollar-space-dollar renders the space in math mode and switches back to normal text.

[4]...CVS...???

[5]...The psgml homepage...`http://www.lysator.liu.se/projects/about_psgml.html`

A.3 Available hardware and software

This section lists the hardware I have used during development of Cactus.

A.3.1 Asserballe

Urls in long banes for this.

- Pentium-S 133 MHz PC with 128 Mb RAM and 10 Gb IDE disk
- Linux Redhat 6.1
- MySQL[6] 3.22.22
- `Apache/1.3.11 (Unix) PHP/3.0.15 ApacheJServ/1.1 mod_perl/1.21`
- Cocoon 1.7.1[7] for rendering XML to HTML on the fly. Can do SQL-requests, and render them easily to HTML.
- IBM JDK 1.1.8 for Linux[8]. This is currently the fastest and most reliable Java engine for Linux.
- PHP[9] for dynamic web-pages with none or some SQL-data.
- Perl 5.6.0[10] for doing complex background tasks with DBI::MySQL perl modules for accessing the MySQL databases.

The choice of JDK was made at a time when top bug at Javasoft Developer Connection[11] was “please provide a port of Java to Linux”, and where IBM beat Sun to it by porting their Java 1.1.6 implementation to Linux. The benchmarks showed it to be as fast as their Windows and OS/2 ports, which at that time was the fastest Just In Time compilers on the PC-platform. It performs very well on Linux, and the installed version is just a maintainance upgrade.

MySQL and the corresponding JDBC driver *must* be the latest stable versions due to security and bug fixes.

A.3.2 Aalborg

(4-CPU UltraSparc 440MHz with Sun JDK 1.2)

[6]...MySQL...`http://www.mysql.org`
[7]...Cocoon 1.7.1...`http://xml.apache.org`
[8]...IBM JDK 1.1.8 for Linux...`http://www.ibm.com/java/jdk/118/linux/`
[9]...PHP...`http://www.php.net`
[10]...Perl 5.6.0...`http://www.perl.com`
[11]... top bug at Javasoft Developer Connection...
`http://developer.java.sun.com/developer/bugParade/top25bugs.html`

A.4 Bibliography with Web references

Full reference with a lot of annotated URL's.

Bibliography

[Codd(1970)] E.F. Codd. A relational model of data for large shared data banks. http://www.acm.org/classics/nov95/, 1970. The paper that described relational databases mathematically. The history is explained in http://www-4.ibm.com/software/data/pubs/papers/bontempo/.

[Damsbo(1999)] Martin Damsbo. Web access course organizer. http://www.mip.sdu.dk/ mez/waco, 1999. MIP Intranet.

[Dybvig(1996)] R. Kent Dybvig. *The Scheme programming language: ANSI Scheme*. Prentice-Hall PTR, Upper Saddle River, NJ 07458, USA, second edition, 1996.

[Goosens et al.(1997)Goosens, Rahtz, and Mittelbach] Michel Goosens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion - Illustrating Documents with TeX and PostScript*. Addison-Wesley, 1997. Describes in great detail the intricacies of embedding graphics in LaTeX documents.

[Greenspun(1998)] Phil Greenspun. Sql for web nerds. http://photo.net/sql/, 1998. Excellent introductory text to SQL.

[Greenspun(1999)] Phillip Greenspun. *Phillip and Alex's Guide to Web Publishing*, volume ISBN: 1-55860-534-7. Morgan Kaufmann Publishers, Inc, 1999. An excellent guide to running websites seen from an OpenSource view.

[Hofstadter(1979)] Douglas Hofstadter. *Gödel, Escher, Bach - an Eternal Golden Braid*, volume ISBN:?? ??, 1979. A very deep book which has influenced my thinking in many ways, both regarding computers, molecular biology and Bach, as well as the depth of self-reference. For fellow readers only: I am positive I have found the end of the book.

[Knuth(1992)] Donald E. Knuth. *Literate Programming*, volume ISBN 0-937073-80-6. Stanford, California: Center for the Study of Language and Information, 1992. Knuths own page[12].

[Kopka and Daly(1999)] Helmut Kopka and Patrick W. Daly. *A guide to LaTeX*, volume ISBN: 0-201-39825-7. Addison-Wesley, 1999. An excellent book for the LaTeX author. I have used it a lot.

[Melton and Simon(1997)] Jim Melton and Alan R. Simon. *Understanding the New SQL: A Complete Guide*. Number ISBN 1-55860-245-3. Morgan Kaufmann Publishers, San Francisco, California, 1997. A good introduction and reference to SQL-92. Defines the requirements for the various levels of SQL 92 compliancy .

[12]...Knuths own page...`http://www-cs-faculty.stanford.edu/ knuth/lp.html`

[Sedgewick(1990)] Robert Sedgewick. *Algorithms in C*. Addison-Wesley, 1990. ISBN 0-201-51425-7.

[Sewel(1989)] Wayne Sewel. *Weaving a Program – Literate Programming in Web*, volume ISBN: 0-442-31946-0. Van Nostrand Reinholdt, New York, 1989. A good introduction to the original WEB system, as well as the CWEB dialect for C programming.

[Tanenbaum(1987)] Andrew S. Tanenbaum. *Operating systems - design and implementation*, volume ISBN: 0-13-637331-3. Prentice-Hall International, Inc., 1987. An excellent book in the inner workings of operating systems in general and Minix in particular. The complete source code is listed in the back of the book. *Minix was the direct inspiration of Linux.*

Contents

A.5 Cross references to this page have not been inserted at the correct page

A.6 The importance of a web cache

A database query is expensive, and it requires an expert to tune the database to run as fast as possible. It is not, however, always necessary to have the webserver do a database query to serve a page - often the generated page is valid for a short or long term period, and then it is relatively easy to cache the page for this period.

Here is one way to do it:

- Configure the script generating the page, to add an “`Expires''` header with a reasonable time of expiry
- Set up squid (12.2.3 on page 107) in http-accellerator mode, where it transparently adds cache facilities to a webserver, respecting the “`Expires`” header.

Mangler:

- myurl footsnotes - urls should be broken if too long, numbers should perhaps not be reset at new chapters, slightly smaller bottom margin
- myimage maa gerne hyperlinke til tiff eller pngfil i hyperudgaven? ER det muligt?
- tal med sm om internetbibliotekarer
- afsnit om filosofi/osv med brugernes egne vaerktoejer til at lave sgml dokumenter.

"The devil is real. He lives inside C programs."
Phillip Greenspun – [Greenspun(1999)] page 202