Database backed Websites
D R A F T – Revision: 1.14

Thorbjørn Ravn Andersen

March 16, 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 vurderes derefter.

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*.

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

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 or similar.
- Virtual stores - the user may browse and order from a virtual inventory. The order is then dissected 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 dataloss.

- Program development - The Mozilla project[1] has shown the benefit of up-to-date information regarding just about anything related to the source code. write about it

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.

[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??. 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"

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

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

FO Formatting objects. A generic description of the physical layout of a XML-document on paper. FOP converts this to PDF. Several users have expressed doubt about this XSLT transformation

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 A family of languages. The *DTD* specifies which language it is. XML-documents can be transformed with *XSL*-style sheets to another XML-document (this process is called *XSLT*). XML is a subset of *SGML*

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

DTD The Document T? 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 W3C's initial model for representing an XML tree internally. Superceeded by SAX (for java?)

SAX Simple Application interface 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. (URL?). Parsers and processors? which implement SAX can be selected freely, currently allowing for 20 different combinations of parsers and whashallicallem.

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 style sheets something. Is used to render an SGML document to another format. "`jade`" can render to FOT, RTF, TeX (must be post-processed with "`jadetex`"), SGML and XML. Is this a standard?

FOSI Another style sheet for SGML, which I do not know anything about. Apparently the US Navy uses it a lot. Check in DocBook.

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 parts of the visual information which is hardest for the human eye to see.

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.

Explain XML, XSL, XSLT, HTML (yes!), SQL, SAX, DOM, XSP and whatever fancy terms come to mind. Draw graph of what happens with an XML document.

Chapter 3

Dynamically generated documents

3.1 The original vision: A Web-server with a file system

Badly phrased ... The original capabilities of a web server was : Nice figure... Mention URL

- A underlying file system containing a lot of *static* files. These files were either pages written in HTML, or images in GIF or JPEG formats.
- The *Common Gateway Interface*. If a given script complied to this, the web-server could start it with user-provided parameters, and return the result as any kind of file.

That's 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 NSCA 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.

A well-tuned server which serves documents directly from the filesystem can reach impressive numbers. Where was this - look for statistics. On a modern PC Apache has no problem saturating a 10Mbps Ethernet connection (what about a 100Mbps connection), meaning that even for very large and demanding sites the bottleneck will not be within the webserver software.

On a related note: The ftp-server `ftp.cdrom.com` serves a lot of files everyday. On March 14'th 2000 the transfer statistics[4] said that the average output 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 lone machine runs FreeBSD[5] (see www.freebsd.org[6] for details). A ftp-session have a notably larger overhead than a http-session, so it is not unreasonable to expect similar performance from a suitably tuned web-server for static files.

[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 NSCA was 10835. . . – `http://www.netcraft.com/Survey/Reports/9508/ALL/`
[4]. . . the transfer statistics. . . – `http://www.emsphone.com/stats/cdrom.html`
[5]. . . This lone machine runs FreeBSD. . . – `ftp://ftp.cdrom.com/.message`
[6]. . . www.freebsd.org. . . – `http://www.freebsd.org`

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".

One of the first demonstrations of dynamically generated images was the map generator somewhere on the web (this ??? quick hack has since been superceeded by professional map companies which produce high quality maps on demand).

a reference to the problems with CGI? From CGI.pm perhaps

Unfortunately, there are a few short-comings with CGI-scripts:

- **Runs as a subprocess** – 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 ISP's - 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 webserver.

  This has caused interpreted programs like Perl to be very popular. These do not require recompilation for a given platform, and are for most purposes as fast as the equivalent programs written in C. Recently Java servlets has shown as a popular and well-supported alternative to CGI-scripts.

- **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.

  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. Java Servlets are discussed in section **??** on page ??, PHP3 scripts in section **??** on page ??, and Perl (with the mod_perl accelleration module) in section **??** on page ??).

3.2 Navigational structure should not be maintained by the author

reference? Almost all dynamic web-sites which add and delete web pages run into the problem of maintaining site-integrity, regarding ensuring that all the "links" point to the correct document. For external document all you can do is to check occasionally that the page is still there, but for internal documents the webmaster has to do the updates manually. Updating HTML-documents manually is at best a tedious pain, because it is hard, repetitive, mindless labour, which should be left to a computer.

See figure?? for a navigational framework on the Fyns GNU/Linux User Group home page.

Reference to FLUG.DK

The most simple version of this, is to say that every web page on the site must have a static header usually containing at least a link to the entry page and a search engine, but where the header is the same for every page.

In the original webservers this required that every webpage 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. (Modern webservers allow a webpage to include other files with Server Side Includes, 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).

Many modern websites put a given page in a *context* where the navigational framework reflect this context. See figurevrefintel-1 for a sample from Intel regarding the VS440FX motherboard[7]. This requires more discipline for the web author writing each page, but can still be implemented by including the appropriate HTML-snippet, since these are the same for all pages regarding this motherboard.

Reference to INTEL.COM with navigational fraework for motherboard vs440fx.

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

3.3 A good website needs meta-information

Creating In order to generate a good navigational framework automatically

YAhoo - is generated from a list with lots and lots of links with attributes

3.4 The consequence - multiple views of a document

Introduce publishing where HTML is just a backend amongst many (PDF, Word, ASCII).
Describe why it is important to be able to render finished versions of documents fully automatically from a single *annotated* source (www/print/cdrom/Palm Pilot/braille/handicapped persons,whatever). The better the annotation, the better the output (ref: Stibo). Describe the need for SGML (history/usage) and XML (why/browser support/on-the-fly publishing/bleeding edge - being standardized).
Donald Knuth - TEX/Web/Literate Programming - advantages (high quality, excellent math, superb algorithms, can be tailored to needs (basic interpreter written in TeX) and disadvantages (programming language, not abstract) (designed 20 years ago). Ask Steffen Enni about his thoughts. Javadoc. Perl POD. DocBook projects (FreeBSD, LDP).

[7]... Intel regarding the VS440FX motherboard... – `http://www.intel.com/...`

3.5 The user should use current tools to publish documents

Publishing a document to the web should be as easy as printing a document. The basic principles are the same - you can do with a "Publish"-button and a dialogue box where the essential information is provided. It is not so today - discuss reasons . Use "print to fax" software as horrible example of this idea gone wrong.
When the webserver is dumb, you cannot do much. If there is a database underneath, much more is possible. Discuss the idea of having several ways of entering documents in the database (upload via form, send as email (fax software can do this too), print to virtual printer, scan, fax, voice) and letting the software do the conversions.
Use example with print PostScript to file and upload via ftp to printer (LexMark).
Writing XML directly is much harder to do than writing HTML (stricter syntax, more options, plainly just more to type), and should be aided by a good tool. Alternatively, the user should use well-known tools (Word) and mark up according to very strict rules, which is then automatically converted to the XML document. This does not give as rich documents, but allows users to publish existing documents with very little trouble.

3.6 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 (A.2.3 on page 62) in http-accellerator mode, where it transparently adds cache facilities to a webserver, respecting the "`Expires`" header.

3.7 The need for well-defined standards

The need for open, well-defined standards. HTML, PDF, PNG (versus GIF) RFC-things (SMTP, POP3, IMAP, MIME) good examples. Word counterexample (look for Bill Gates saying people can buy an upgrade to read Word97 files in Word95). Adobe has good documentation on their PostScript and PDF document formats (doing theirs to have it well supp orted). HTML is widely documented in RFC.

3.8 Why is Open Source essential?

Independence from hardware and software vendors, allowing free choice of hardware platform and operating system, as well as the possibility to fix bugs and use other products. (Examples: XT versus Xerces, Java Interpreter for Linux, servlet technologies, cheap - use money for hardware instead of licenses).

Chapter 4

Sample websites

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.

4.1 slashdot.org - high volume information site for nerds

Describe setup. Describe flow, and the slashdot effect. What hardware/software.

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 stories 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 **??** on page ?? for the top of the start page at March 14 2000.

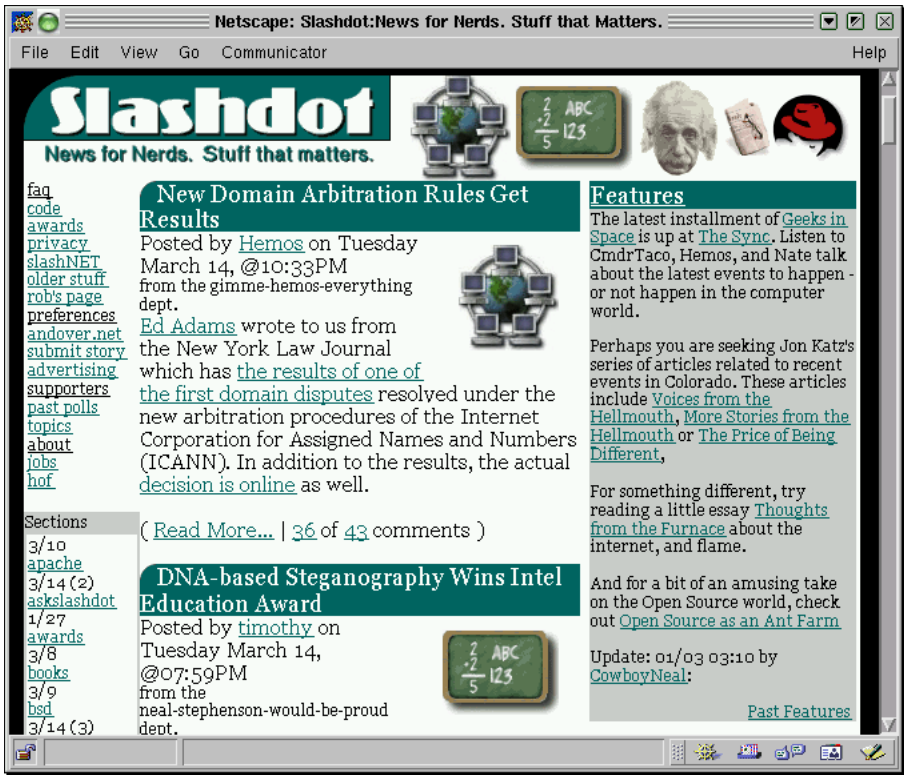
```
Slashdot Stats

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
```

4.2 Valueclick.com

Banner provider. Interview Ask for details, about how the NT-cluster was replaced with a small Linuxbox with MySQL and a squid in http-accelleration mode. Explain the problems with having fat processes serving slow modems.

[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

4.3 Politiken

Talk with Lars. Politiken genrates their pages on the fly.

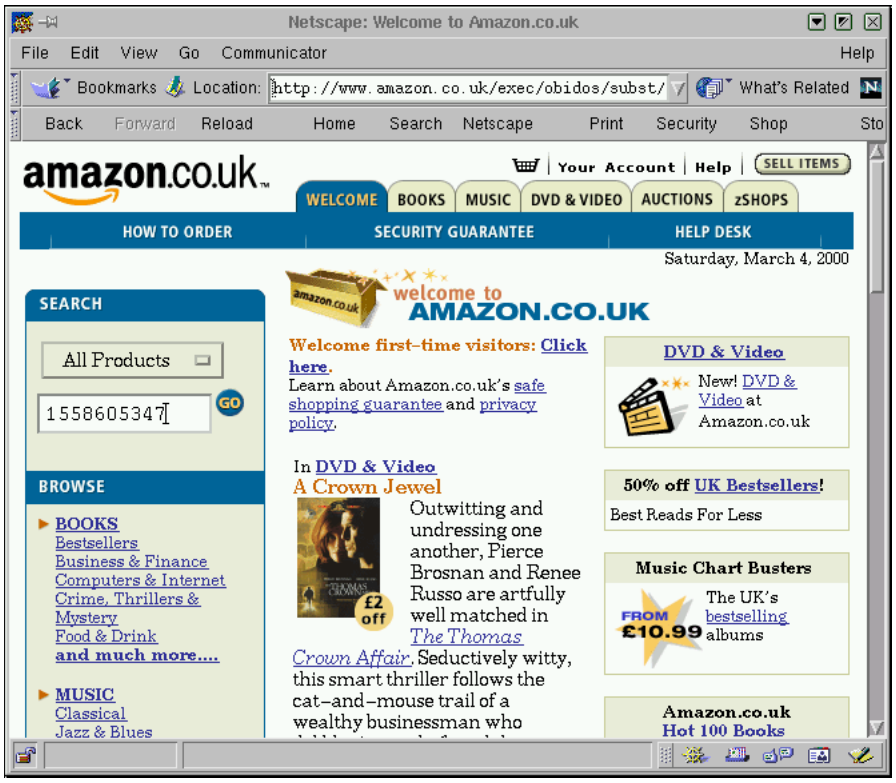
4.4 Amazon - an Internet bookstore

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 others.

The Amazon frontpage is shown in figure 4.2 on the next 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 4.3 on page 16).

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. These features are important for Amazon:

[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
GO
BROWSE

Saturday, March 4, 2000

1558605347

welcome to AMAZON.CO.UK

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

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

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

£2 off

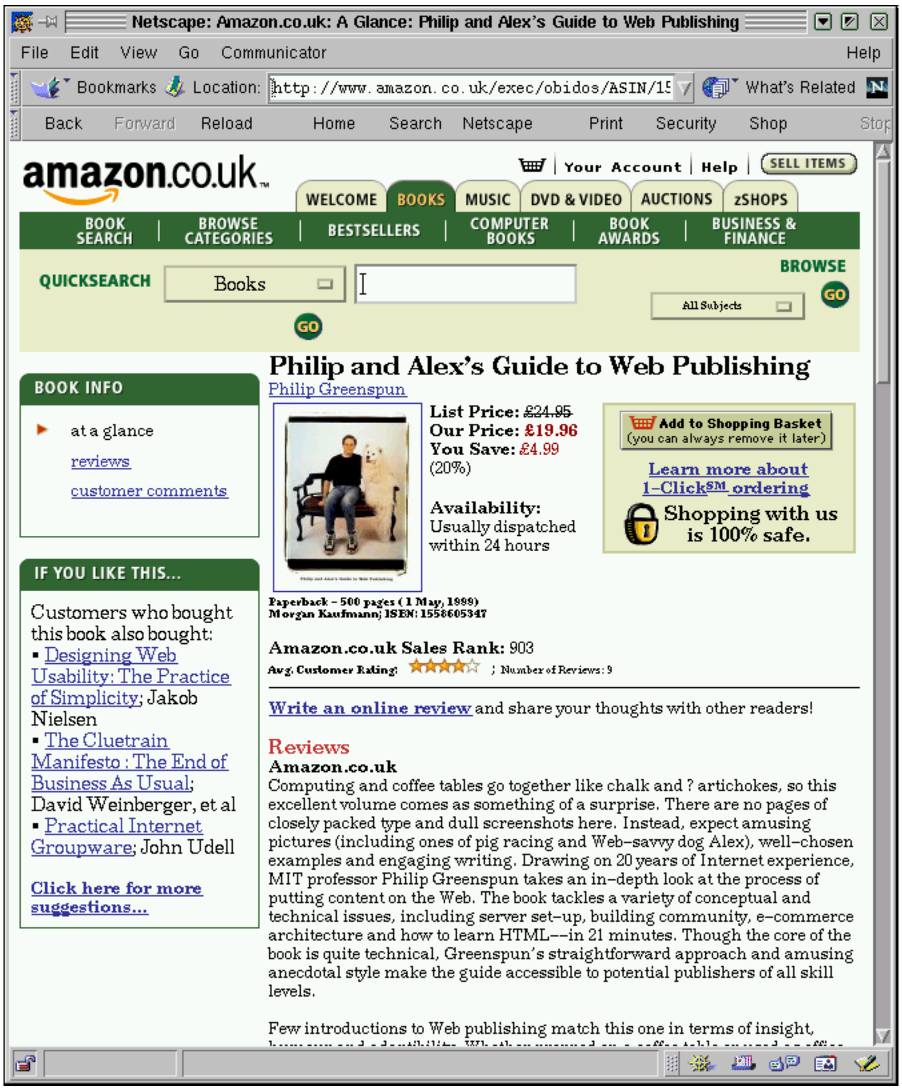
50% off UK Bestsellers!
Best Reads For Less

Music Chart Busters
FROM £10.99
The UK's bestselling albums

Amazon.co.uk Hot 100 Books

- **BOOKS**
  - Bestsellers
  - Business & Finance
  - Computers & Internet
  - Crime, Thrillers & Mystery
  - Food & Drink
  - **and much more....**
- **MUSIC**
  - Classical
  - Jazz & Blues

Figure 4.2: 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
BOOK INFO
at a glance
reviews
customer comments
IF YOU LIKE THIS...
Add to Shopping Basket (you can always remove it later)
Learn more about 1-Click℠ ordering
Shopping with us is 100% safe.

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

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!

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,

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...

Figure 4.3: The result of the search in figure 4.2 on the page before

- **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 each poll is shown along with the review, giving the content more value.
- **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 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.

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 provide excellent

4.5 www.krak.dk

Map making on demand, was one of the earliest applications for the web. As mentioned earlier the web map application was started in 199?, and has since been followed by yahoo map, whoelse.

In Denmark, the well renowned Danish map manufacturer Krak A/S[3] opened their web map in 199? in direct competition with 909, 118 and who else? since they provided a person/number lookup facility. Their advantage was up-to-date data from the Telephone companies combined with an ability to generate a map for an address, whether it was entered directly or being a byproduct of a search for something else.

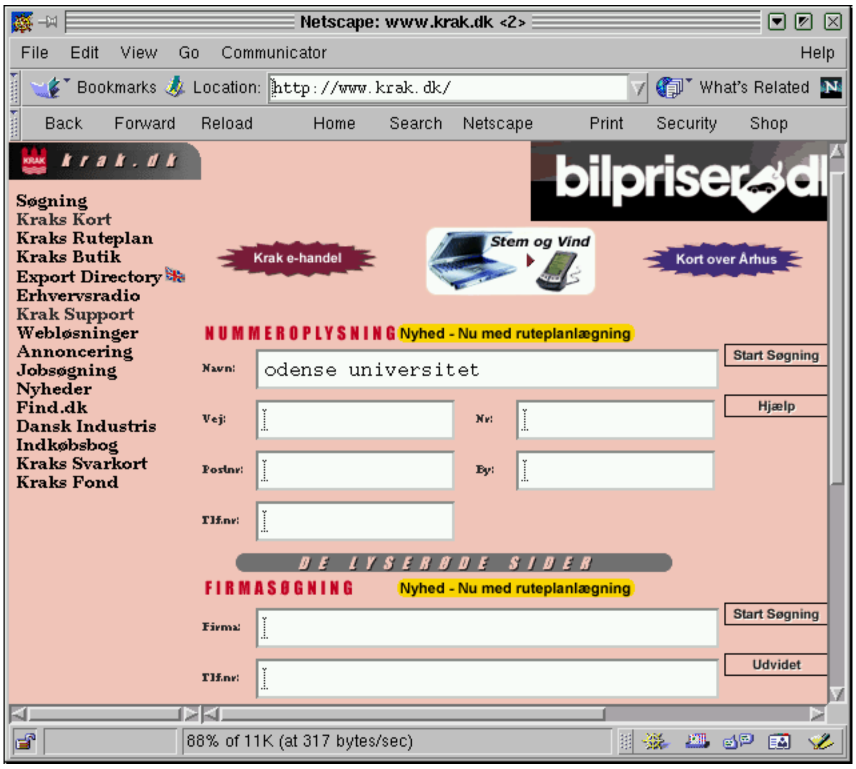
The website has been steadily improved with new facilities and cross-references, as well as more precise data (In ?? 1999 it could not locate "Herlev Hovedgade 205" on the map, so the whole of the road was inked on the map. This is corrected today). The start page is shown in figure 4.4 on the following 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 Phonenumber. The fields in the pink page section are Company and Phonenumber. The form has been filled out with a search request for "Odense Universitet", and the results are shown in figure 4.5 on the next page.

Each line containing an answer may have icons referencing to facilities regarding the location of 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 4.6 on page 19.

The blue dot usually indicates the location of the address in question, but in this case it is a bit off (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" a new map is shown (see figure 4.7 on page 19), 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.

[3]. . . Krak A/S. . . – http://www.krak.dk


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
krak.dk
bilpriser.dk
Krak e-handel
Stem og Vind
Kort over Århus
Start Søgning
Hjælp
Start Søgning
Udvidet
DE LYSERØDE SIDER
88% of 11K (at 317 bytes/sec)

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

NUMMEROPLYSNING Nyhed - Nu med ruteplanlægning

Navn: odense universitet

Vej:

Nr:

Postnr:

By:

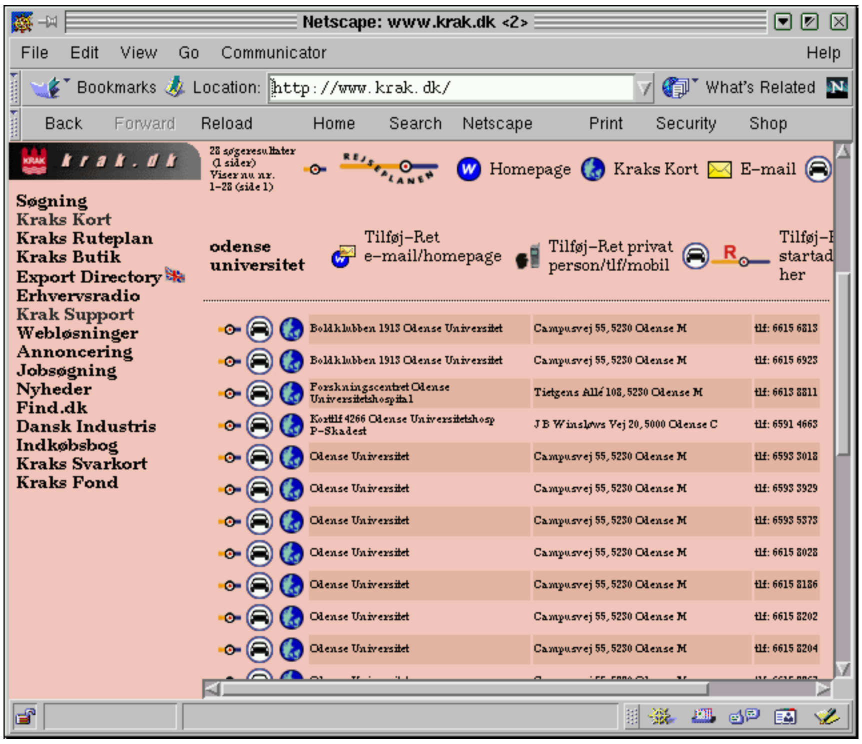
Tlf.nr:

FIRMASØGNING Nyhed - Nu med ruteplanlægning

Firma:

Tlf.nr:

Figure 4.4: Welcome page (and search form) for www.krak.dk


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
krak.dk
REJSEPLANEN
Homepage
Kraks Kort
E-mail
Tilføj–Ret e-mail/homepage
Tilføj–Ret privat person/tlf/mobil
Tilføj–I startad her

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

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 4.5: Result of searching for "Odense University"

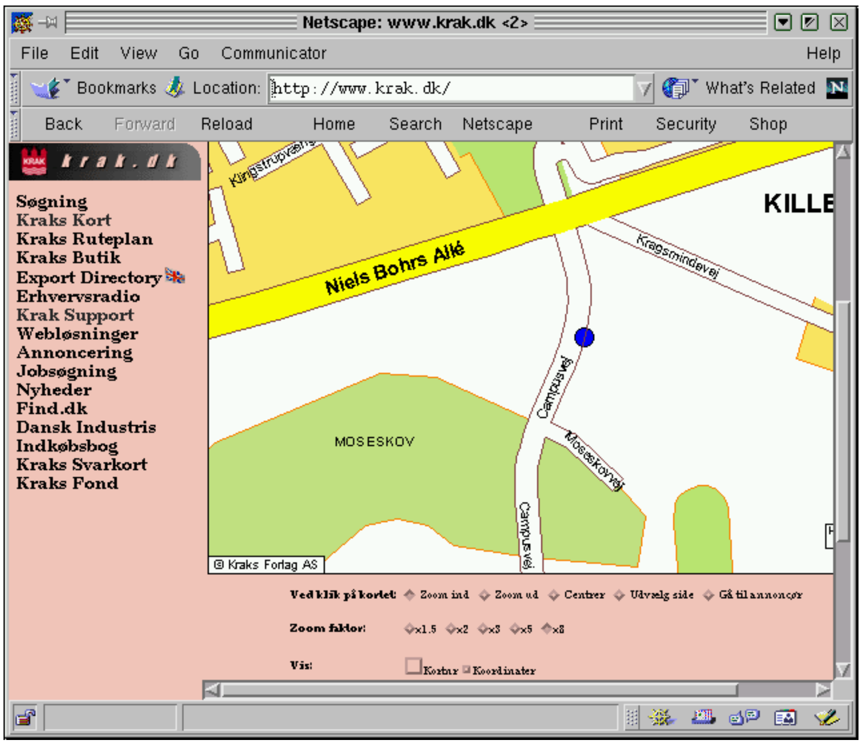


Figure 4.6: Following the map icon for the first "Odense University" reference

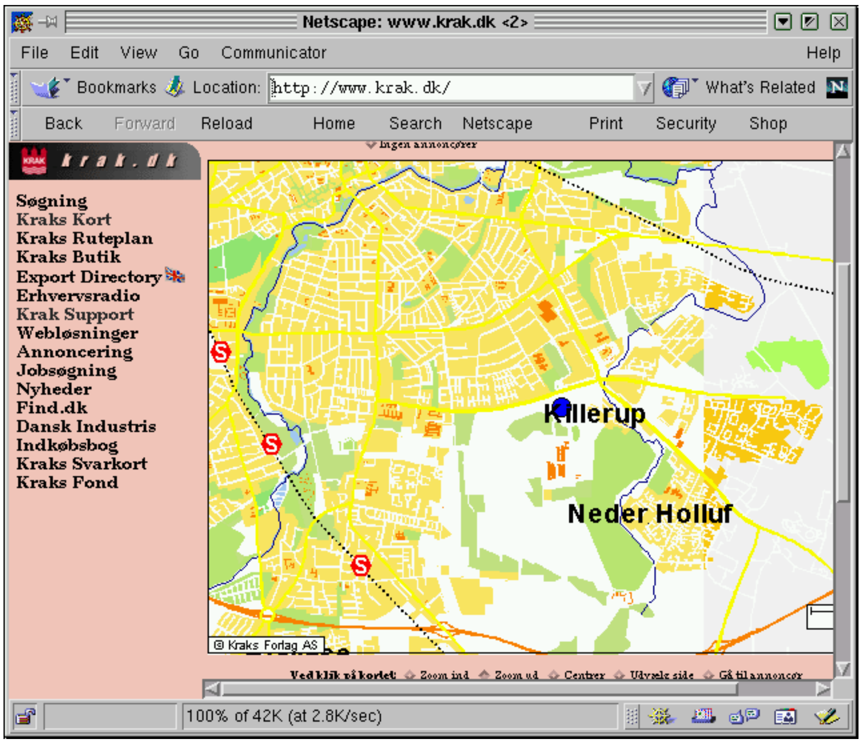
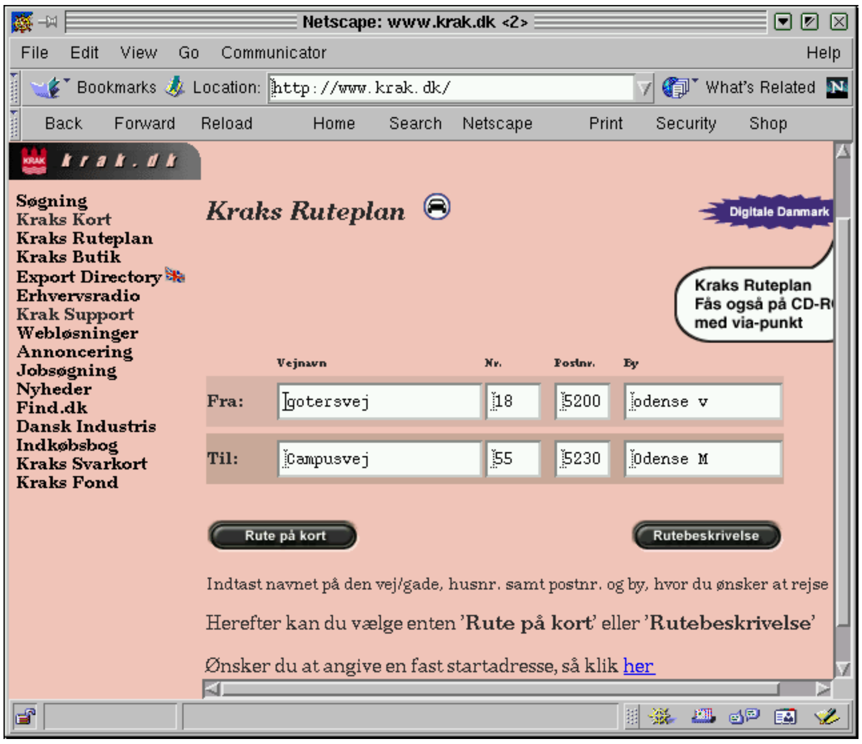

Netscape: www.krak.dk <2>
File Edit View Go Communicator Help
Bookmarks Location: http://www.krak.dk/ What's Related
Back Forward Reload Home Search Netscape Print Security Shop
Ingen annoncører
krak.dk
Søgning
Kraks Kort
Kraks Ruteplan
Kraks Butik
Export Directory
Erhvervsradio
Krak Support
Webløsninger
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 4.7: The map in figure 4.6 zoomed out with a factor 8

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 4.8 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.



	Vejnavn	Nr.	Postnr.	By
Fra:	gotersvej	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 4.8: ?????????????????????? University” reference

This map is shown in figure 4.9 on the facing page, 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 “???” button and print out the directions.

Necessary to do rendering according to requests. Cannot be predicted reasonably. What have been done? How does it work? What about

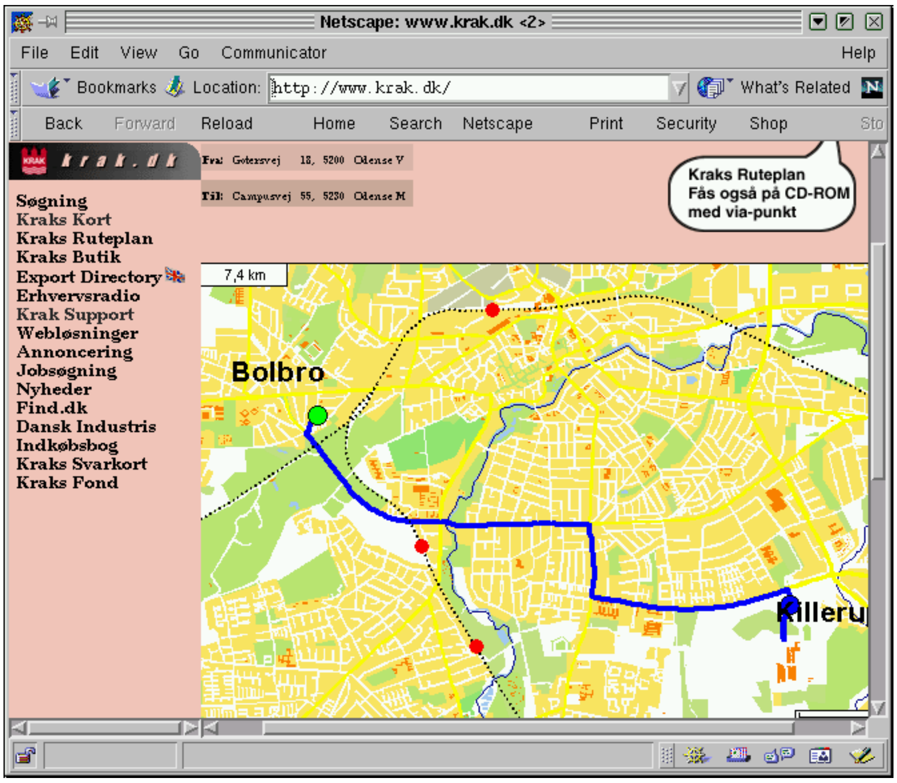
4.6 The Journay Planner - www.rejseplanen.dk

DSB (Danish Rail) have a journey planner which is based on train stations, and major bus stops. Figure 4.11 on page 22 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 70% of the area to the application itself.

Figure 4.12 on page 22 is the same as the previous figure???

The search returns a number of potential journeys, 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.

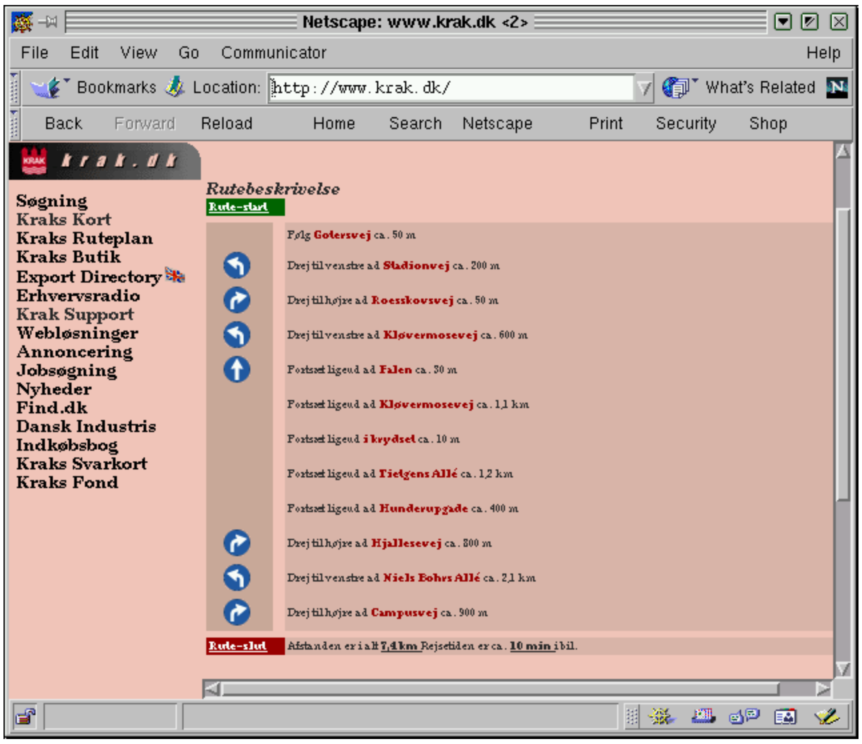
Figure 4.13 on page 23


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
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
Kraks Ruteplan Fås også på CD-ROM med via-punkt
7,4 km
Bolbro
Killeru

Fra: Gotersvej 18, 5200 Odense V

Til: Campusvej 55, 5230 Odense M

Figure 4.9: ?????????????????????? University" reference



Rutebeskrivelse

Rute-start

Følg **Gotersvej** ca. 50 m

Drej til venstre ad **Stadionvej** ca. 200 m

Drej til højre ad **Roesskovsvej** ca. 50 m

Drej til venstre ad **Kløvermosevej** ca. 600 m

Fortsæt ligeud ad **Falen** ca. 30 m

Fortsæt ligeud ad **Kløvermosevej** ca. 1,1 km

Fortsæt ligeud **i krydset** ca. 10 m

Fortsæt ligeud ad **Tietgens Allé** ca. 1,2 km

Fortsæt ligeud ad **Hunderupgade** ca. 400 m

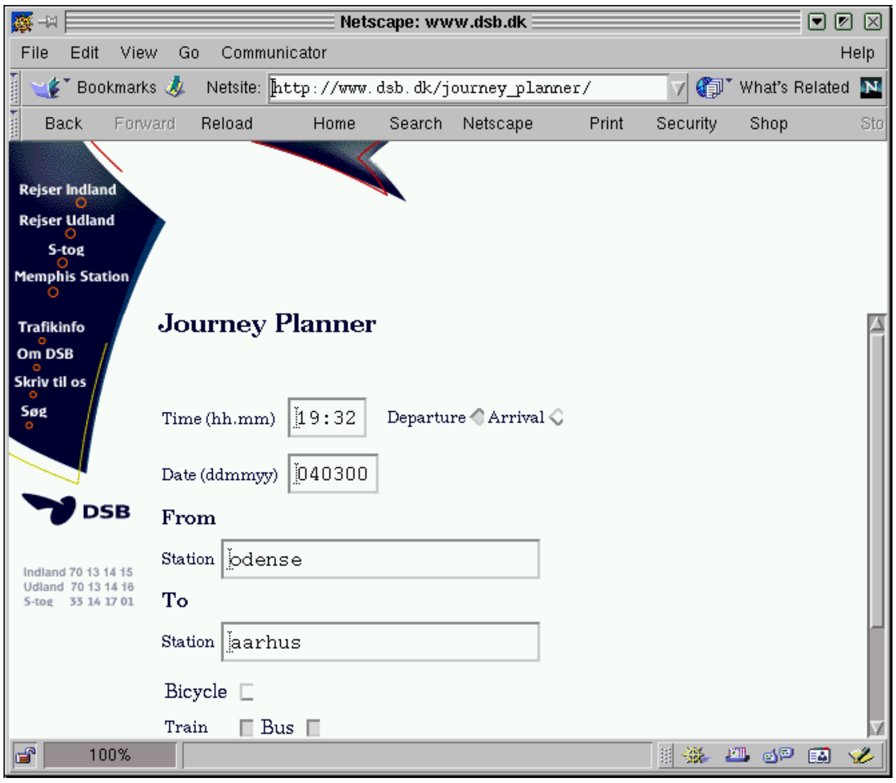
Drej til højre ad **Hjallesevej** ca. 800 m

Drej til venstre ad **Niels Bohrs Allé** ca. 2,1 km

Drej til højre ad **Campusvej** ca. 900 m

Rute-slut Afstanden er i alt 7,4 km Rejsetiden er ca. 10 min i bil.

Figure 4.10: ?????????????????????? University" reference


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%

Journey Planner

Time (hh.mm) 19:32 Departure Arrival

Date (ddmmyy) 040300

From

Station odense

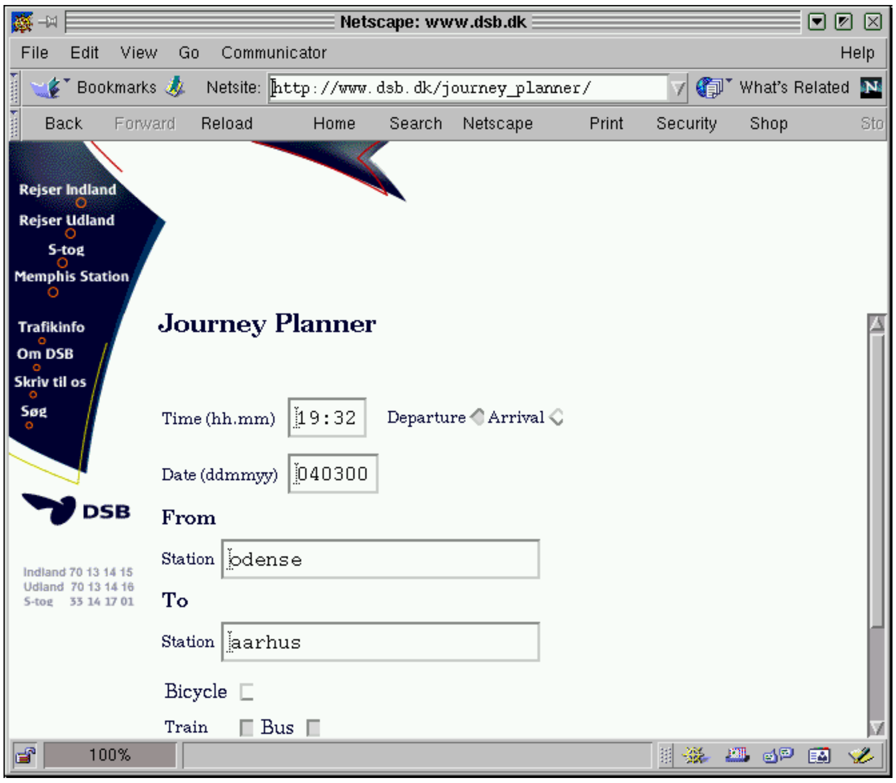
To

Station aarhus

Bicycle

Train Bus

Figure 4.11: Rejseplanen1


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%

Journey Planner

Time (hh.mm) 19:32 Departure Arrival

Date (ddmmyy) 040300

From

Station odense

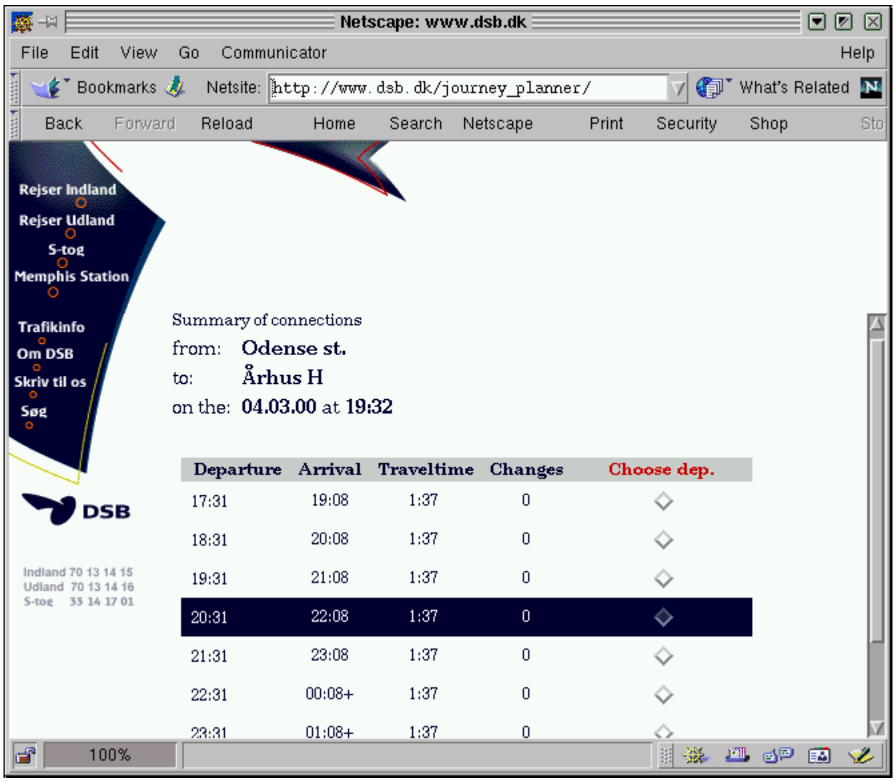
To

Station aarhus

Bicycle

Train Bus

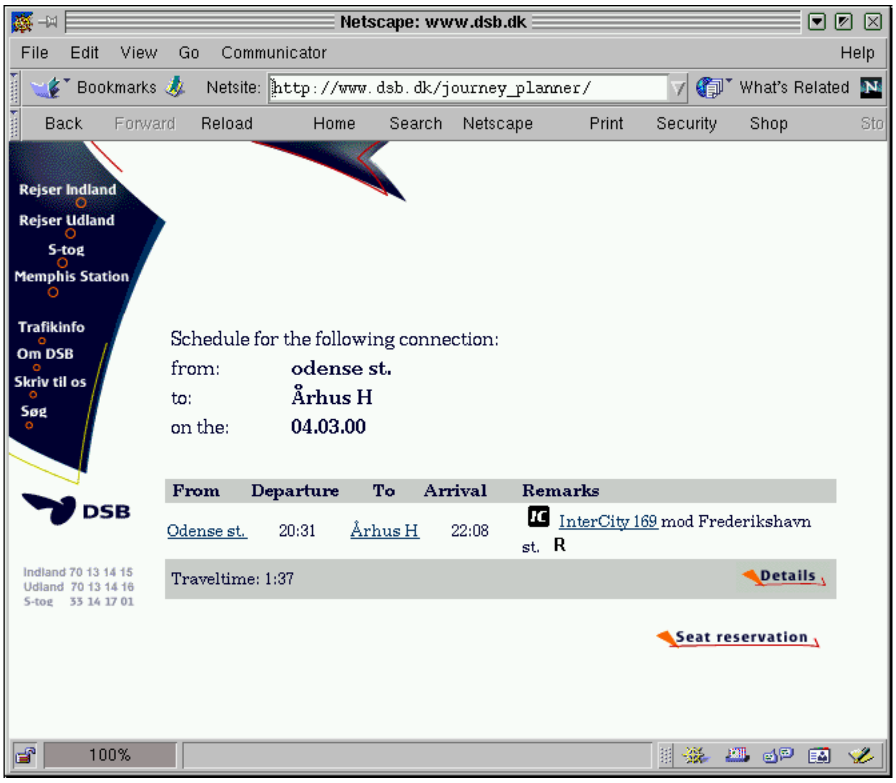
Figure 4.12: Rejseplanen2


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%

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 4.13: Rejseplanen3



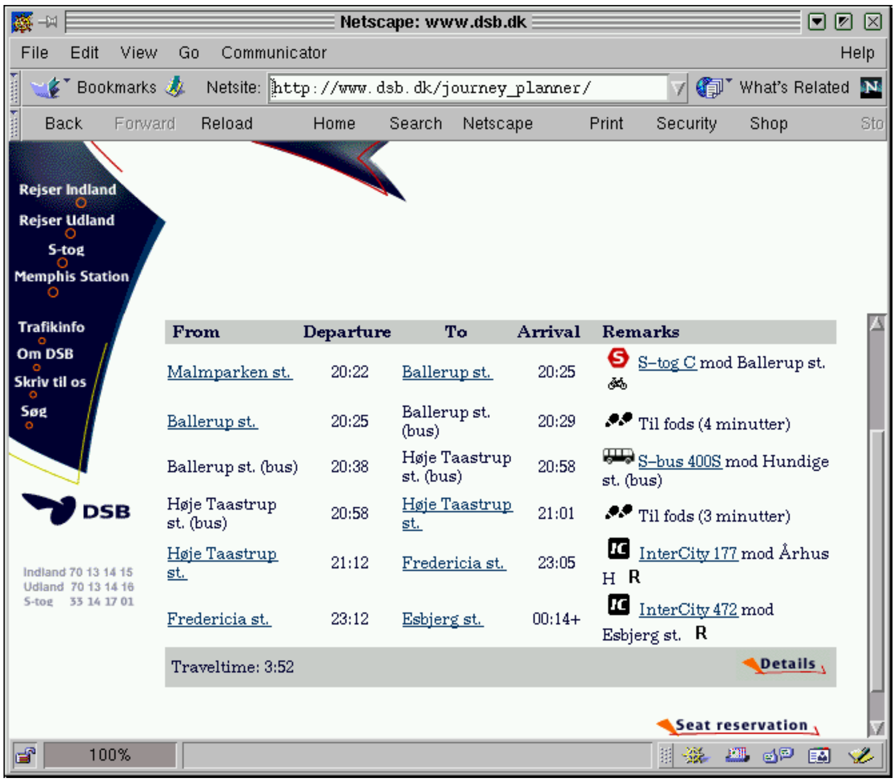
Schedule for the following connection:
from: odense st.
to: Århus H
on the: 04.03.00

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

Traveltime: 1:37

Figure 4.14: Rejseplanen4

Figure 4.14 on the preceding page Figure 4.15



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

Figure 4.15: Rejseplanen5

4.7 Freshmeat.net

4.8 Google/Altavista

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[4]

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

4.10 Jydske Bank

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

4.11 Cactus – document capture and conversion

Two out of four methods are implemented (email, printer). (Increase blob limit in DBI::MySQL).
Sample PDF viewer, perhaps simple XLS viewer.
Demonstrate that Cocoon can provide the navigational framework.

Chapter 5

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.

5.1 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.

5.2 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?

5.2.1 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.

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

5.2.2 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] 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. Evaluate?

5.2.3 Informix - Informix

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

5.2.4 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.

5.2.5 mSQL - ?

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

5.2.6 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.

5.2.7 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.

On the FLUG meeting ??[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.

[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`
[10]... the FLUG meeting ??... – ??

5.2.8 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.

5.2.9 Other approaches to database storage

Landbrugets rådgivningscenter[13] uses a LDAP-server[14] to provide easy access to commonly used web elements.

Others?

5.3 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?),

5.3.1 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.

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.

5.3.2 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.

In the past two years, the OpenSource developers have basically rewritten the Communicator into a truly impressive product (this is based on the M14 milestone release). A tentative release is midsummer 2000.

[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]Light Directory Access Protocol? - me thinks
[15]. . . The Mozilla browser. . . – `http://www.mozilla.org`

5.3.3 Opera

This Norwegian browser from ?? has been highly successful on Windows for providing a good browser to smaller, older machines. They have a beta port under development what is the status? XML?.

5.3.4 HotJava 3.0

HotJava is a Netscape 3.0 compatible browser written in Java.

It is a bit slow, especially in 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 no information regarding the status of parsing XML internally. Full access to the source is probably necessary before any third-party implementations come

Rumours has it that development on this browser has ceased. I have tried to extend version 1.1.4 to show TIFF files, but had to give up due to the miserable documentation 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 - previously Netbeans), which could indicate that they had evaluated their experience with previous releases, and found that nothing short of full OpenSource will do.

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" system for JavaURL, 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 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. 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.

5.3.5 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). 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[16] - 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.

Try Solaris version of MSIE under KDE

5.3.6 Amaya

Amaya is the W3C testbed for HTML-development. Is a nice browser combined with a reasonable editor. Does not do XML at all. Don't they have a testbed?

5.3.7 AWTviewer in fop

probably for fop stuff only

5.3.8 StarOffice

What can we do here? - try i tout.

StarOffice include a Netscape 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.

5.3.9 OTHER BROWSERS?

5.4 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.

5.5 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.

[16]. . . A certain lawsuit springs to mind. . . – ??

Chapter 6

SGML, XML and DTD's

A major ingredient in developing large, complex systems 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.
- Expensive printers provide the PostScript language which is an abstraction for "electronic paper". Any Postscript file can be printed to any PostScript printer
- 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.

SGML history[1]

The same thing happened in the printing industry, where SGML (Standard General Markup Language) was developed in the late sixties and early seventies 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.

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 become able to fully present XML documents individually tailored to the users preferences. 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.

When XML is common place in browsers, we will hopefully have much more content in webpages allowing for better search engines, less usage on system dependent facilities, and better tools for writing XML

[1]. . . SGML history. . . – `http://www.sgmlsource.com/history/roots.htm`

6.1 "HTML is a SGML DTD" - the concepts

Table 6.1 shows how the most frequently used acronyms relate when mapped to the programming domain. See section 2 on page 5 for the full descriptions.

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 6.1: The analogy between programming languages and SGML

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.

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 XSL-Formatting Objects (XSLFO).

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

6.1.1 The author should not do layout

I need articles here.

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.

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 "windows" 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 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 TeX 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 handled this merely by increasing the margins of a normal page, and nothing else. The result was that the layout looked awkward - the line spacing seemedq 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 on a CD-ROM, or into a Braille version for blind persons. It might even be read aloud by a computer.

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.

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.

6.1.2 The design and evolution of SGML

Look up SGML history when/where/whom/*WHY*. What does SGML do well? Why is it so? Initially hard to write in. Development of DTD’s to use in applications, with Stylesheets (FOSI, DSSSL (when)). Development of tools for making writing easier (complex, expensive, slow). Use liking to printer drivers for printers, which was sweeped away by PostScript for LaserPrinters (bitmaps are too hard to handle - talk about Stibo rendering in 2400 dpi to guarantee idential prints).

6.1.3 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 e.g. Windows Help, or something else.

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

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

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

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!

6.1.4 The creation of XML

XML was as HTML built on SGML, but with a lot stricter syntax than HTML and a lot less facilities than SGML. XML is a new standard - it was made a W3C recommandation on ??????1999, and the accompanying XSLT and XPath recommandations?...

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 simple.
- 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

6.1.5 XHTML - XML compliant HTML

The XHTML W3C recommandation[4].

[5]A big problem with XML 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. Doocuments 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 is written as "<hr />", where the space is important to make this acceptable to HTML-4.0 parsers.
- More stuff... Netvwork lag

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

[5]...

... – http://www.w3.org

The well-formedness of an XHTML document is achieved by using tricks like:

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

Why is this smart???

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.

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 SUGGEST? WHY DO THEY THINK THIS IS A GOOD IDEA

6.1.6 What DTD should be used?

The Document Type Definition (*DTD*) defines the way a SGML/XML document should look in order to mechanism which defines , which specifies exactly

- 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 ————————-

6.1.7 How can a *ML document be viewed?

In order to be viewed, 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. Table 6.2 on the following page discusses the current *de facto* document formats on the Internet.

available: Discuss this in a earlier chapter? "As discussed in chapter foo..."

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.

Table 6.2: Document distribution format

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.

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.

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

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?

This style sheet 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. Line numbers in this listing would be nice. What about colour encoding? Can it be done?

```
<!--
| 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 6.3 on the next page lists the tested XSLT processors which conforms to the W3C XSLT-19991102 recommendation. emphasis on java due to platform indenpendance

Where should I talk about SAX?

During testing it showed very quickly that the Java tools are quick enough to 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

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 6.3: Leading XSLT processors implementing XSLT-19991102 in February 2000

advantages of Java over C (reference big list, probably at JavaSoft?) are

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 was indirectly prompting the implementation of remote jobs? 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 B.3.1 on page 67 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>XXXXXXXXXX</username>
   <password>XXXXXXXXXX</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>iteml</t
<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>
```

6.1.8 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. Table

Due to the rather inflexible DTD-requirements in XML, it is most convenient that such a tool generates stand-alone XML (SGML ok if it can be converted? With what?), which can be processed freely in the system. This is hopefully something which gets resolved soon, since the validation of the tree structure in an XML document is as important as it being well-formed. rewrite.

Move filter descriptions to Cactus chapter discussing filters?

Majix	Java based RTF to XML, which I have adapted to DocBook
pod2docbook	Converts Perl documentation to XML
wordviewothers 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 6.4: Conversion tools to XML [February 2000]

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.

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.

tex4h - convert TeX to XML

What daelen does it do

Conclusion:

wordview with friends

Parses binary stream and produces HTML, WML plus more.

6.1.9 Formatting 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[6], 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 Apache is easily capable of saturating a 10Mbps ethernet connection.

Dynamic rendering is suitable whenever the data changes very often, or there is a wish for the user to be able to personalize their view of the presented documents (should frames be used or not? do the user want large or small versions of images?), or if the conversion process is light and

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.

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

6.2 DocBook considerations

DocBook[7] 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 with request for a sgml conversion. The stylesheet generates a HTML file pr section in a chapter which is rather too much. The HTML is *very* ugly - all induced line breaks are *within* the tags to avoid introducing any artificial whitespace. The HTML is Lynx-compatible.

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.

Having XML files in DocBook format (basically having a valid DocBook DTD) allow to use all the available DocBook stylesheets (both XSL and DSSSL) to ?? them. These are:

XSL –

- **XHTML-1.0** –
- **HTML-**4.0 –
- **XSL-FO** – The W3 page description language. Renderer? like fop and the commercial one
- Others ?

DSSSL –

- **RTF** – Rich Text Format. Renderers?
- **MIF** – Frame format. What about it
- **TEX** – Renders with the JadeTeX package.
- Others ?? What else can we do with this.

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

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 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. What can be done about that?

6.3.2 XML to RTF

Use `jade` with the appropriate stylesheet and ....

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

[7]...DocBook...– http://www.docbook.org

where docbook.dsl is one of the stylesheets from the DocBook Modular StyleSheets (dsssl/docbook/print/docbook.dsl 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, other texteditors

Chapter 7

Standard Query Language, Databases and Webservers

"You 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.

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 (liftet 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;
```

which then returns the table shown in figure 7.1 on the next page:

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

The table shown here was originally in ASCII; inside an application program it is common to take a row at a time retrieving a structure with the content of each field.

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(1997)], 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 is 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.

7.1.2 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.

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.

I decided to dvelwe into MySQL since it had excellent Perl support, which I had chosen for developing the Cactus system (see A.1.4 on page 59 for reasons), and was OpenSource software available on Linux.

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

[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`

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.

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.

7.2 Some of the possible ways of accessing a database from a webserver

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.

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

Perl 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[8]. Currently are Informix, Interbase, MySQL, Oracle, Sybase and the generic ODBC interface supported.

Vendor dependant – Each vendor have their own way of moving HTML (and some XML) in the database itself. These are usually not portable, and therefore not interesting from a generalist point of view. The databases surveyed

7.3 What databases are available?

[4]. . . ArsDigita. . . – `http://www.arsdigita.com`
[5]. . . photo.net. . . – `http://photo.net`
[6]. . . Open Database Connectivity. . . – `http://www.microsoft.com/odbc`
[7]. . . JDBC Data Access API. . . – `http://www.javasoft.com/jdbc`
[8]. . . database interface modules for Python. . . – `http://www.python.org/topics/database/modules.html`

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

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.

7.5 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.

7.6 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

Conclusion

8.1 Conclusion

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

Appendix A

Sample implementation – Cactus

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

A.1 abstract

The Cactus system fills a gap in the current integration of the web with normal office procedures, as it provides easy web publishing for occasional authors, by letting them submit documents in several ways with their usual software.

The system automatically produces other versions of the documents on demand of the users reading the documents, as well as provide the navigational framework, relieving the local webmaster from these tedious and error prone tasks.

A.1.1 System description

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

A.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.

The Yggdrasil system was designed to provide a simple, consistent navigational framework around a dynamic set of web pages provided by users, as well as providing a ”recently changed pages”-list plus an overview of pages written by each person. This was a very successful idea in the start, but gradually lost momentum due to these factors:

The development platform was changed from Unix to NT, which made it much more difficult for the users to access their web-directories, as these were no more a part of their home directory[1]

The internal document format became Microsoft Word, which at that time could not be converted to HTML. Such documents were therefore unable to be published.

The users could trigger an update by sending an empty email to the system, as well as rely on an automatic nightly update. The trigger mechanism was not brought along when the email system was converted to NT.

New users was not informed about the system.

A.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.

[1]The Apache webserver uses the /public_html directory for the users personal web pages. The transition to NT meant that the users - in addition to their normal file manipulations - should telnet to a Unix server, and change the file attributes every time the file was updated.

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,

There are the following tasks:

Derivers - derive another version of a given item. In theory, this is a reversible operation. These should generally be designed to be as fast as possible, since these will be called from CGI-programs with users waiting. Avoid compression (gzip should maximally be level 1).

Converters - create another item from one or more originals. This could be a multi-part MIME file which should be assembled into a fresh original. It could be a DVI file from several source files (tex files and images). These should generally be reasonably fast, since their speed specifies how fast a new item can be made available to the system.

Optimisers - A deriver is usually designed for being usable in a real-time interactive setting (also called being fast), which normally means that the result is sub-optimal. An optimiser complements this by doing a suitable optimisation step whenever the system is sufficiently idle. This could be running ”pngcrunch” on PNG files (which yields 30% on Ghostscript output), or ”tifftotiff” on TIFF files. The idea is that the file is still the same basic type, and can be used without further modification. It is an ”in-place optimisation”. Some file formats are using the gzip compression scheme internally. These would not benefit further from archiving.

Archivers - creates a long-term storage version of an item. This would typically be running ”gzip -9” on the content, creating a new entry (with a new mimetype), and marking the item as decachable. The dearchiving process must be fast, since it might be needed by a deriver without notice.

Validators [spelling] - validates whether the content of the item is consistent with the MIME-type, and is it conforming to the standard. (Can a gzip-stream be decompressed? is a PostScript file parsing correctly? etc.). If a type is unknown, guess MIME-type from filename and/or contents, and validate it. If all fails, assign a type of application/octet-stream.

Examiner - looks into an item to look for references and encoded data. A reference may be an URL or an emailaddress in a signature. Encoded data could be an uuencoded image in a Usenet posting, or a pointer to a usenet article.

These are complemented by

Acquirers - gathers items from ”outside” Cactus and enters it in the ”incoming” table with an appropriate MIME-type. 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. Derived 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. The page should be cut in smaller pieces to allow easier processing by Unix Netscape.

In an ideal world we have infinite storage and infinite CPU-speed, meaning that everything would be done instantly when we need it. In order to decide the order in which to do tasks, a price system must be developed which would guarantee that the system processes every item, that the system is still responsive while converting,

What to do when the system runs so full that archiving cannot be done fast enough. Can data be moved to ”outside storage?”.

A.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.

A.1.5 Converters

Microsoft Word

The Microsoft Word DOC-format is widely used, but apparently so hard to use that even Microsoft cannot do it reference to Word97 not being able to create Word95 files, 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.

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 (URL). 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 also available for 16 bit versions of Windows, which behave better in WINE. A full Windows installation was not available to me when I tested this - this might have improved. This can be used for producing the PostScript printouts.

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 commandline. A StarBasic program must be written and invoked to do the job ask on newsgroups.. This is currently untested, but probably the way to go. 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.

AbiWord - An Open Source word processor which shows great promise. It was not able to parse my test documents in Word.

mswordview Others? 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-myself - 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 can also parse DOC files, by invoking Word to save the DOC file as RTF. This shows great promise, except that this mechanism requires the GUI to be available for Word (it needs to pop up the window). The ingenious scenario of a telnet server on a NT-machine, which accepted remote logins from Cactus, started Majix which again started Word, did not work. Word hangs. 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.Yes?

TEX and LATEX

A.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]

A.2 Programs

A.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.

A.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

[2]...KVEC...– `http://ourworld.compuserve.com/homepages/kkuhl/`
[3]...WMF docs...– `http://www.companionsoftware.com/PR/WMRC/WindowsMetafileFaq.html`

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).

Other potential webservers? Roxen? Jetty?

A.2.3 Squid - a high performance web cache

The http-accellerator mode[6]

A.2.4 Cocoon

rtf2latex http://www.tex.ac.uk/tex-archive/support/rtf2latex2e/

[4]. . . Apache. . . – `http://www.apache.org`
[5]. . . the `mod_perl` module. . . – `http://perl.apache.org/`
[6]. . . The http-accellerator mode. . . – `http://www.squid-cache.org/Doc/FAQ/FAQ-20.html#what-is-httpd-accelerator`

should this go away?

A.3 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 B

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.

B.1 LaTeX

Guide is [Kopka and Daly(1999)].

B.1.1 Including screen dumps

Screen dumps have been captured with xv[1] (which is now 6 years old and still unbeat) and saved as `tiff` files. These were then converted to `png` and `eps` with the NetPBM package[2], 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

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

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

B.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 also nice when English is not your native tongue.

CVS – Emacs supports CVS[3] 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[4] - consider writing a set of AUC-TeX compatible keybindings for DocBook

B.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.

[3] ...CVS...– ???
[4] ...The psgml homepage...– `http://www.lysator.liu.se/projects/about_psgml.html`

B.3 Available hardware and software

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

B.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 2.3.?? - security updates
- Apache `1.??` with mod_perl, ApacheJServ, Cocoon.
- IBM JDK 1.1.8.
- CGI, DBI::??? perl modules for writing CGI scripts and accessing MySQL databases.

The choice of JDK was made at a time when top bug at Javasoft 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.

B.3.2 Aalborg

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

B.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.

[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.

[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 .

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

Contents