Web-Technologies

- Chapters
  - Server-Side Programming: Methods for creating dynamic content
  - Web-Content-Management
  - Client-Side Programming
  - Excurs: Server Apache
  - Search engines and Spiders

Client-Side Programming 1

- To Recall: HTML
  - HTML = HyperText Markup Language
    - Developped since 1989 as platform independend markup language
    - International standardized by the W3C
    - Last release: Version 4.0
    - Often extendend with non-standardized tags by developer of browsers and web-authoring-programs

Client-Side Programming 2

- Example base structure of a HTML-document

```
<HTML>
     <HEAD>
        <TITLE>My HTML-Document</TITLE>
     </HEAD>
     <BODY>
        <P>Hallo World!</P>
     </BODY>
</HTML>
```

Client-Side Programming 3

- XML
  - Extensible Markup Language
  - With help of XML ist possible to define content and layout of a page in several parts => automatic analysis is possible. In other words:
  - „XML is a set of rules for designing text formats, in a way that produces files that are easy to generate and read (by a computer), that are unambiguous, and that avoid common pitfalls, such as lack of extensibility, lack of support for internationalization, and platform-dependency.“

Client-Side Programming 4

- Simple example of XML Usage:

```
<?xml version=„1.0“ ?>
<!DOCTYPE greeting [
        <!ELEMENT greeting (#PCDATA)>
        <!ELEMENT content (#PCDATA)>
]>
<greeting>Hallo XML! </greeting>
<content>
Here, we write a nice text that says nothing, but is our content...
</content>
```

- See also: http://www.w3.org/XML/
  http://www.w3.org/TR/2000/REC-xml-20001006

Client-Side Programming 5

- JavaScript
  - JavaScript is a cross-platform, object-oriented scripting language.
  - Used mostly within HTML-pages.
  - JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.
  - Created originally by Netscape and Sun Microsystems. (Within MSIE „extended" with the JScript-Library).
  - Allows also usage for server-side programming

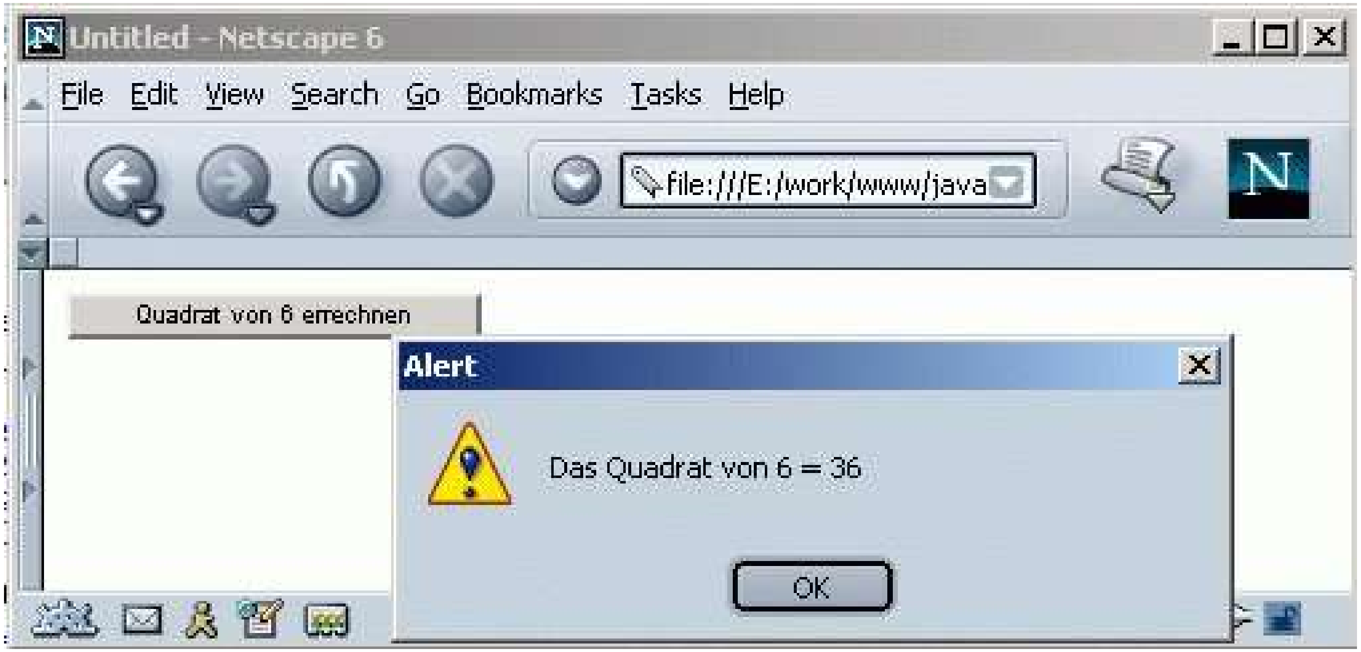
Client-Side Programming 6

- Sample JavaScript

```
<html>
<head>
        <title>Beispiel</title>
        <script language="JavaScript">        <!--
         function Quadrat(Zahl) {
          Ergebnis = Zahl * Zahl;
          alert("Das Quadrat von " + Zahl + " = " + Ergebnis);
          }       //-->
        </script>
</head>
<body><form>
<input type=button value="Quadrat von 6 errechnen" onClick="Quadrat(6)">
</form></body></html>
```

Client-Side Programming 7

- Sample JavaScript (cont.)



Das Quadrat von 6 = 36

Client-Side Programming 8

- JavaScript (cont.)
  - JavaScript is mostly used as enhancement for webdesign; Due to its possibility to access and chance objects (like HTML-Tags), it allows effects fo improve the usability of websites.
    - Often used: onMouseOver, onClick
    - Professional effects in combination with CSS
    - Replaces Netscape's experiment with „DHTML"
  - JavaScript's core features can be enhanced by new libraries, like DYNAPI

Client-Side Programming 9

- Cascading Style Sheets (CSS)
  - HTML specification lists guidlines on how browsers should display HTML-tags.
    CCS allows to modify these specifications.
  - Example:

```
<style type=„text/css“>
        h1,h2,h3,h4 {
                color: navy;
                font-family: Garamond, Helvetica, serif;
        }
        h1.dark {
                color: black;
        }
</style>
```

Client-Side Programming 10

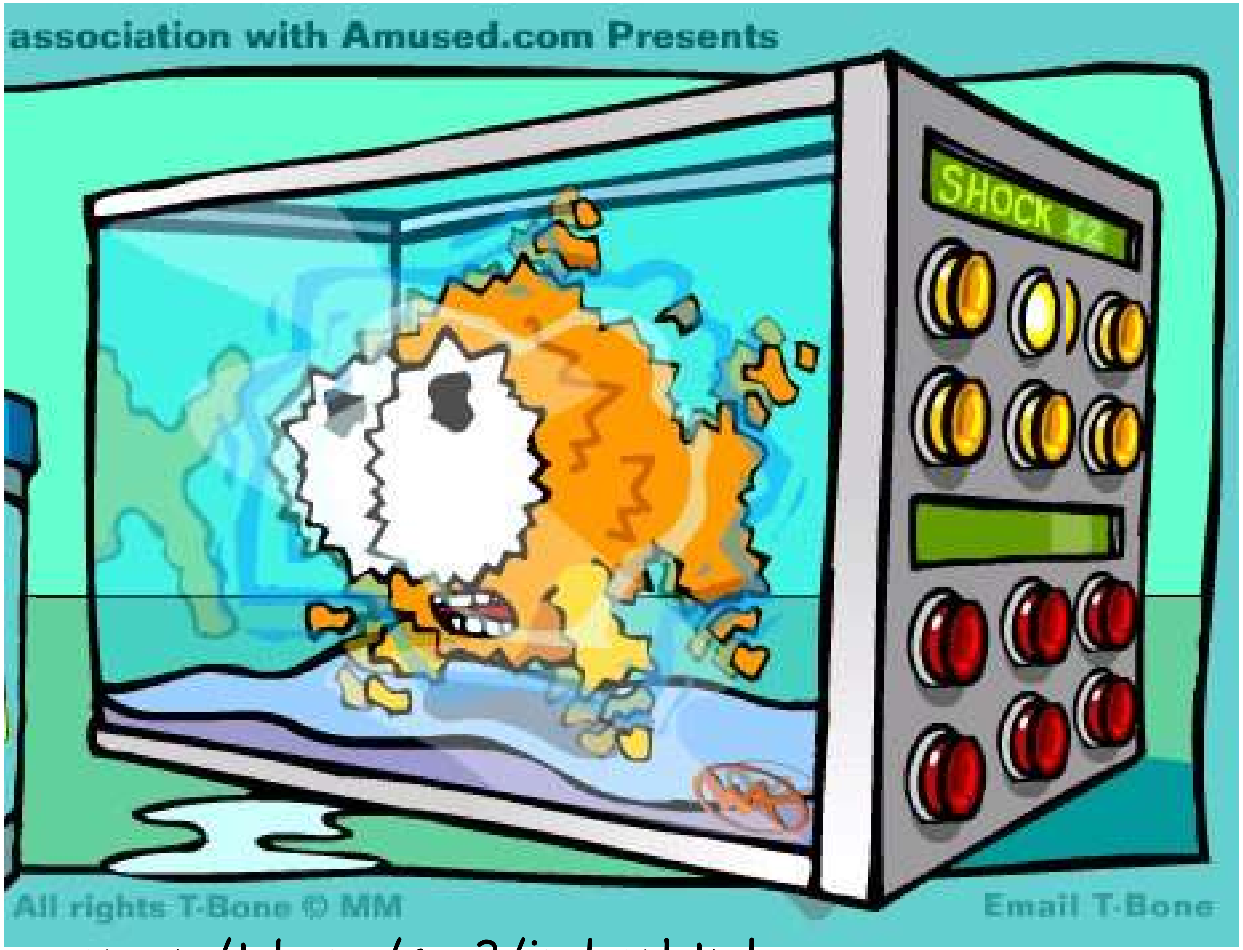
- Cascading Style Sheets (cont.)
  - CSS is, like HTML, standardized by the W3C http://www.w3.org/Style/CSS
  - In combination with new HTML-Versions, it will replace old HTML-tags, like <font>, <hr>, <strong>, ...
  - CSS requires browsers that supports this format (IE / NS >= V4.0)
  - CSS definitions can be placed within a file; Therfor it's possible to chance the layout of all webpages by changing one single css-file.

Client-Side Programming 11

- Flash
  - Browser-Plugin by Macromedia (http://www.macromedia.com)
  - Allows interactive vector-graphics and animations
  - New versions are supporting database-access
  - Mostly used for special effects, small movieclips and 3D-graphics

Client-Side Programming 12

- ⌘ Flash (cont.)
  - Example:



http://www.thewax.com/t-bone/sra3/index.html

Client-Side Programming 13

- Other client-side technics
  - cURL
    - Approach to make dynamic webpages: „client-side applications that get their information from the web during run-time.“
    - language, similar to lisp, which helps to make dynamic text, 3D-graphics and web-accesses.
  - VRML (Virtual Reality Modeling Language)
    - VRML files define worlds, which can represent 3D computer generated graphics, 3D sound and hyperlinks
    - 3D objects can be composed to form new objects. All are made out of polygons
    - Texture mapping is used to add realism

Client-Side Programming 14

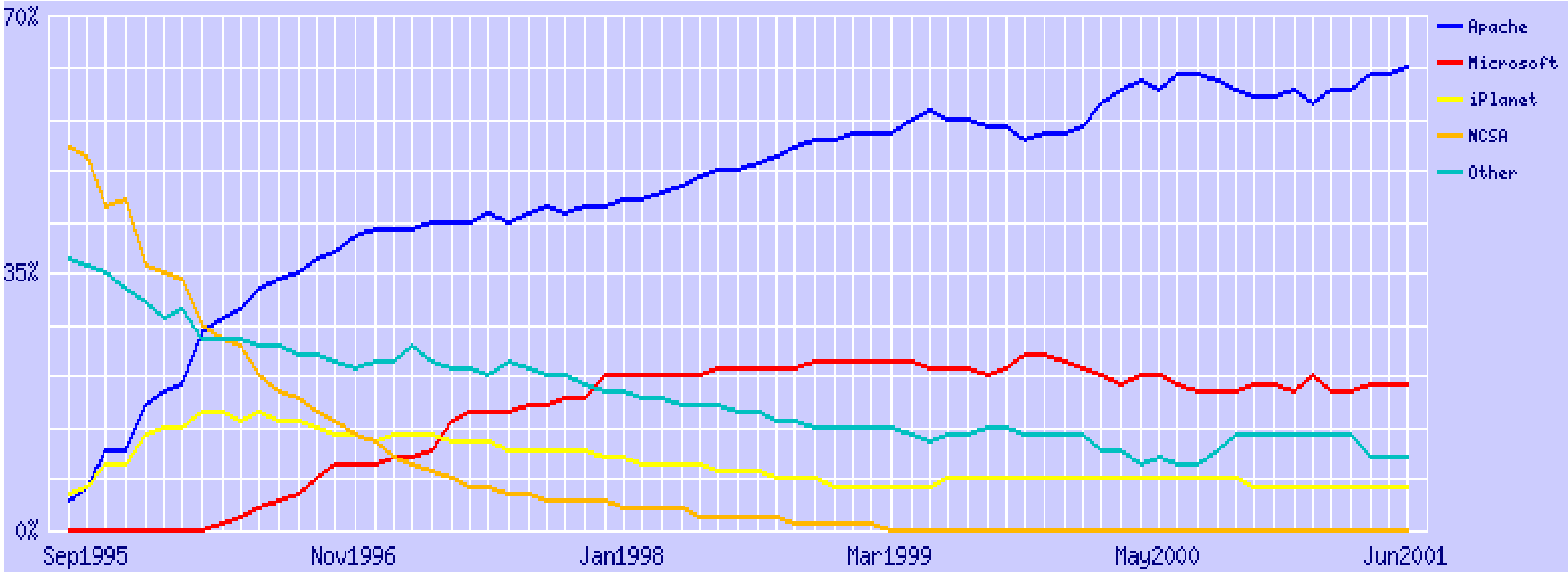
- Design and Usability
  - Fundamentals:
    - Design and textual representation of content of websites and single webpages is dependent on it's target group
    - The reader of a webpage is aware of thousand other pages similar to the current; The starting page has to show a clear navigation or/and show within the first 5 seconds what it is about
    - Animations, interactive scripts and design are audivisual aids for most websites. Content (text) is more important. Do visitors come to see a jumping frog or to read some informations?
  - Several guiding rules in the web. E.g. Jacob Nielsen (http://www.useit.com)

Excurs Apache 1

- Apache („a patchy server“)
  - Free HTTP server, supports HTTP/1.1 (RFC2616)
  - Useable on nearly all OS (but not Mac)
  - Build upon NCSA httpd (V1.3) since 1994. First release of Apache: April 1995, V 0.6.2 as beta
  - First public version in December 1, 1995
  - Developer-Team consists out of volunteers – open source project
  - Today the #1 webserver on the internet
  - Current version (Jul 2001): 1.3.20 as final and 2.0.18 as beta
  - http://www.apache.org

Excurs Apache 2

- Apache (cont.)
  - Currently used by appr. 63% of all servers in use. (MS-IIS: 20%, Netscape-Enterprise/iPlanet: 6%)



http://www.netcraft.com/survey

Excurs Apache 3

- Principle:
  - After start Apache will listen to requests onto port 80 (or any other defined port)
  - Configuration is stored within a textfile „httpd.conf“, which is read by the httpd-process
  - On a request it will fork itself;
  - The child-process will answer the request, close the connection and then die
    - Before sending an answer, the process will parse the requesting URL and look it up for errors.
    - If the request aims a special filetype (like a server-parsed SSI-document), needed moduls are dynamically loaded or called

Excurs Apache 4

- Sample configuration file (extract)

```
Listen 131.188.3.67:80
ServerName www.rrze.uni-erlangen.de
User www
Group www
PidFile logs/httpd.pid
ServerRoot /usr/local/apache
MaxClients 220
...
LoadModule vhost_alias_module libexec/mod_vhost_alias.so
...
AddModule mod_vhost_alias.c
...
```

Spider & Search Engines 1

- Overview:
  - Local search engines
  - Catalogues
  - Web search engines

Spider & Search Engines 2

- Local search engines
  - Real-Time search engines:
    - CGI-script, which opens a list of files and greps it for the searched word:
    - Filelist contains out of all files of a special type (mostly HTML) in a predefined start-directory
    - Subdirectories of the start-directory may be included optionally
    - Duration of search dependent of amount of webfiles, their size and the programming language;

Spider & Search Engines 3

- Local search engines (cont.)
  - Index search engines
    - Avoids time-consuming real-time search though many files
    - Search only in a prepared index file
    - Index file is generated on regular time intervals
    - Two types of index files:
      - Summarization of all searchable files: Contains as entries the simple addition of all files without any chance and the reference to the orginal file
      - Parsed index file: Contains as entries only special Meta-Tag informations, like title, description and keywords of every file and the reference to the original file
    - Index often as textfile.

Spider & Search Engines 4

- Local search engines (cont.)
  - Client-side index search engine
    - Search engine consists out of a client-side script that contains prepared datafields
    - The script will perform the search within these fields and return prepared result on success
    - Mostly implemented with JavaScript
    - Example datafield within script:

```
Portal|info,eingang,start,main|My Startpage|http://www.somewhere.com
Contact|contact,email,adress, impressum|Contact Page|http://www.....
...
```

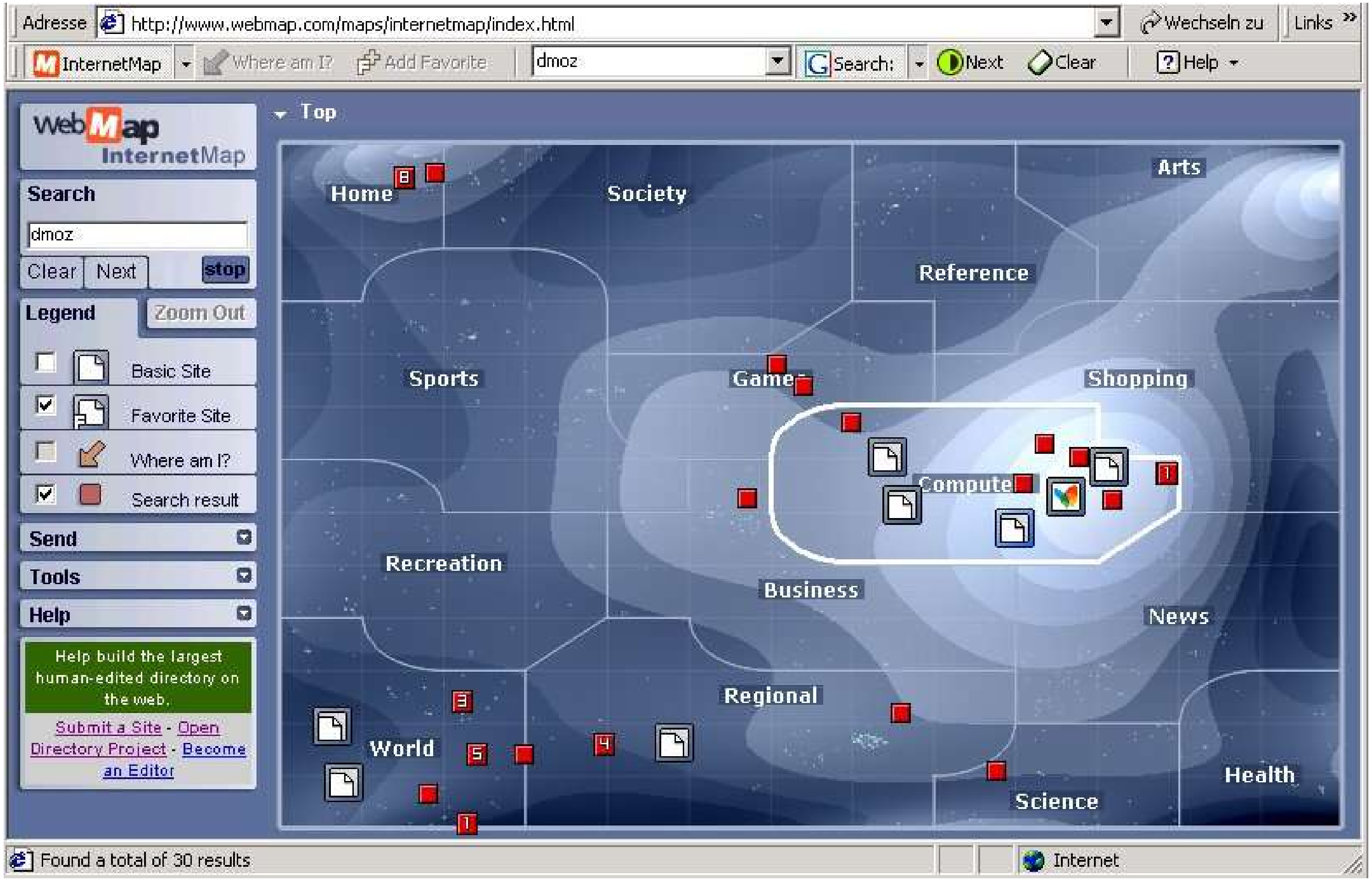
Spider & Search Engines 5

- Catalogues
  - As Websites
    - Examples: Yahoo, Web.de, dmoz Open Directory Project, Portals, ...
    - Entries are made manually or by submit-tools within predefined categories
    - Often entries are checked by humans before their are commited into the index database
    - Indexes without human check get out of control after some time. Entries may get into wrong categories.
    - Management of categories gets complexe on big indexes

Spider & Search Engines 6

- Catalogues (cont.)
  - As Browser-Plugin or standalone Client
    - Index is loaded on demand from a (website) catalogue
    - Examples: Netscape's „What's Related", WebMap
      - WebMap: Graphical Interface to categories of a catalogue
      - Implemented as Plugin
      - Searches within the topics of categories by a changing rating scala for hits, depending on the search-deep.
      - Paints categories depending on the amount of hits

Spider & Search Engines 7



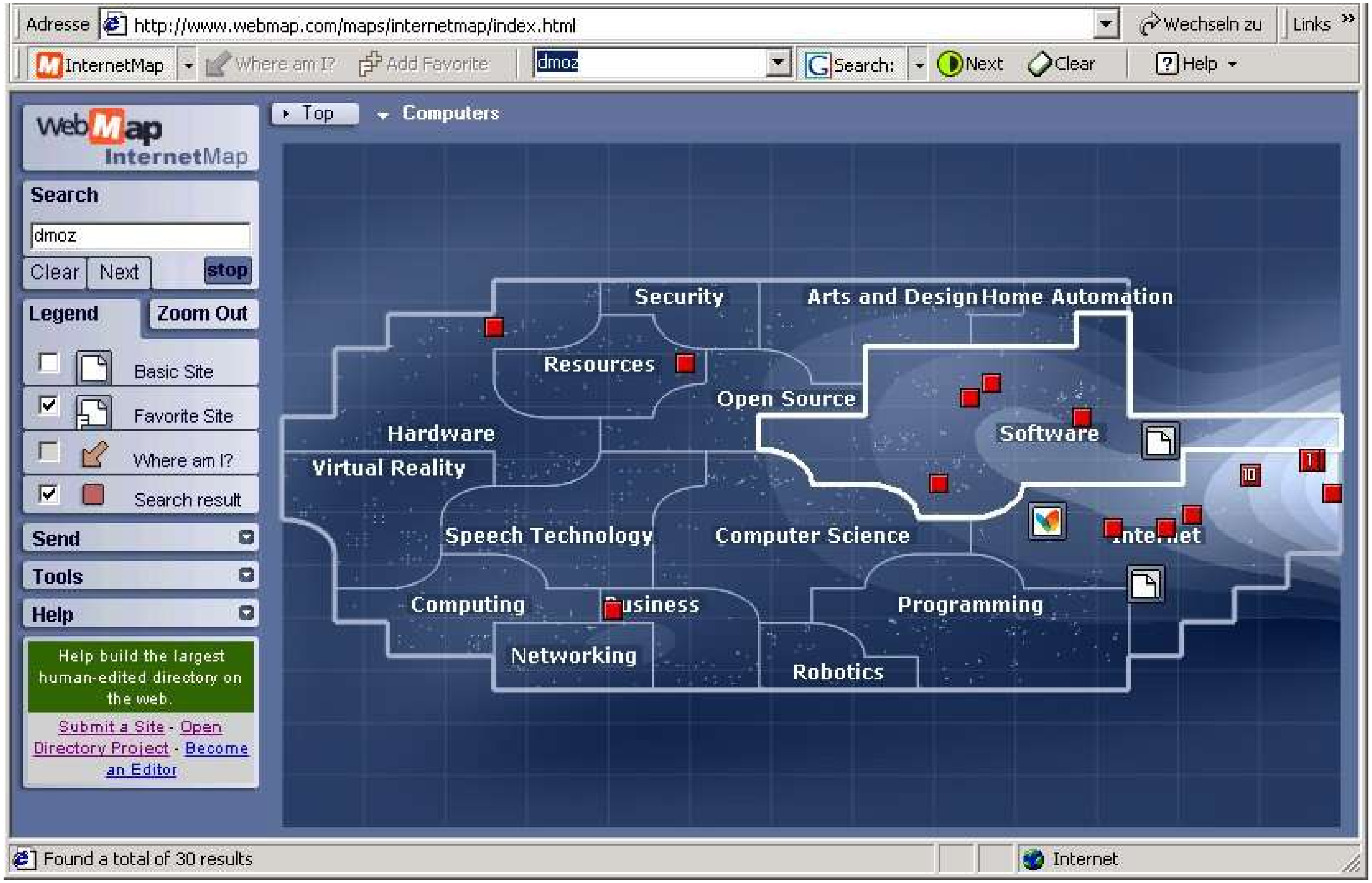
☐ Basic Site
☑ Favorite Site
☐ Where am I?
☑ Search result

Help build the largest human-edited directory on the web.

Submit a Site - Open Directory Project - Become an Editor

Found a total of 30 results

Spider & Search Engines 8



Adresse: http://www.webmap.com/maps/internetmap/index.html

Search: dmoz

Help build the largest human-edited directory on the web.

Found a total of 30 results

Spider & Search Engines 9

- Internet search engines
  - Original searchable files are located on other servers.
  - Real-Time search engines
    - Like local search engine, but instead of local file-open, access using HTTP-protocol
    - Very slow
    - Only used for special tasks like website-watchdogs (tools, that inform users about changes on a predefined URL)
  - Index search engines
    - All big comercial search engines: AltaVista, Google, HotBot, ...
    - Index is part of a high scalable database (Altavista: ~500.000.000 entries)

Spider & Search Engines 10

- ⌘ Internet search engines (cont.)
  - Index search engines
    - ☒ Database is filled up by „spiders“ (also called as robots or crawlers)
    - ☒ Spiders are processes, which are „crawling through the web“ by reading webpages and then following all unknown links defined within the page. At the next page it will do asame.
    - ☒ Spiders can work parallel (by forking) or serial
    - ☒ If a page contains no link, it will continue at the last unknow link or quit if it was started as parallel process
    - ☒ A spider runs over pages until it followed all unknown links (very unlikely!) or it reaches a predefined limit

Spider & Search Engines 11

- ⌘ Internet search engines (cont.)
  - Spiders
    - ☒ Spiders never leave their machine: All „crawled“ pages are downloaded; Therfore the spider is also limited by the bandwidth of its machine
    - ☒ Each entry within the database will time out sooner or later
    - ☒ (Friendly) Spider are following a set of rules, the „Robots Exclusion Protocol“, which works through a standardized file „robots.txt“, that should be located on a webserver which' pages are beeing spidered

Spider & Search Engines 12

- Internet search engines (cont.)
  - Robot-Rules
    - ☒ http://www.robotstxt.org/wc/robots.html
    - ☒ Example „robots.txt“-file

```
User-agent: *
Disallow: /pictures/
Disallow: /intern/
```

- ☒ Robots META tag with a HTML-file

```
<META NAME=„ROBOTS“ CONTENT=„NOINDEX, NOFOLLOW“>
```

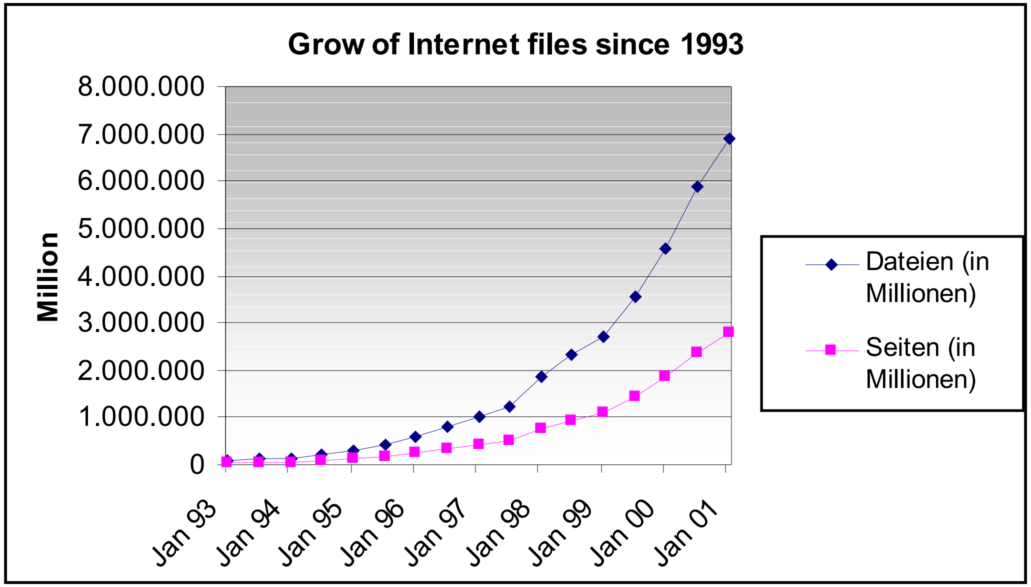
Spider & Search Engines 13

- Internet search engines (cont.)
  - Problems:
    - Due to limited bandwidth and space, it's not possible to index all webpages
    - Spiders cannot parse and index all internet files; They mostly fail at pages generated by client-side plugins
    - Spiders can only follow pages that are referenced! Without manual submit of the URL a spider would never visit a page noone is link is guiding to
    - Typical spiders index only up to 50 pages per domain

    - => Amount of existing internet files is much bigger as a search engine's database

Spider & Search Engines 14

- Statistical for internet files



Data transfered in Jan 2001: approx. 46.328 TeraByte

(Data based on Netstats and Analysis of the Webserver of the University Erlangen-Nuremburgh)

Spider & Search Engines 15

- Perspective – new concepts:
  - Automatically combinations of Catalogues and Index search engines with help of artifical intelligence
  - Distributed search engines
  - Personalized search engines