Components of the World Wide Web

HTML, HTTP, URL
Web-Browser, WWW-Server

Goal of this Module

- **Discuss most important services, protocols, and technologies used in the subset of the Internet that we call the World Wide Web**
- **Each topic represents an own world for itself**
- **So this module can give you an overview only!**

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Agenda

- Introduction WWW
- URL
- HTTP
- WWW Details
- HTML

WWW Principles 1

- **Information stored on Web-servers**
  - Documents in HTML format
    - Hypertext Markup Language
  - HTML is a text description language
    - HTML itself is exactly defined by the usage of Standard Generalized Markup Language (SGML)
    - Several HTML versions today
  - SGML is a system for defining structured document types and markup languages to represent instances of those document types
    - HTML is an application of SGML
    - HTML Document Type Definition (DTD) of a document is a formal definition of the HTML syntax in terms of SGML used within the corresponding document

© 2005, D.I. Manfred Lindner

WWW Principles 2

- **HTML is a semantic markup language**
  - Within the text specific “commands” (**Tags**) are included which describes the logical structure of the given text
  - Technically spoken a HTML document consists of elements (containers), which are bracketed by begin- and end-tags
    - **<h[1]>**text-lawa1**</h[1]>** …. for headline
    - ** **text-lawa2** ** ……. for paragraph
    - **<ul** type=„bullettype"**>**
      **
listentry1
**
      **
listentry2
**
      **
** ..... for lists with bullets
    - **bold**
    - *****italic*****

WWW Principles 3

- **Most important element is the link which makes the text “hyper”**
  - text-to-link
  - URL … Uniform Resource Locator -> unique identifier of a given resource in the Internet
    - http://www.ict.tuwien.ac.at/skripten/datenkomm/index.html
- **Tags are device independent**
  - Will be interpreted at the given output system (GUI)
  - GUI … Graphical User Interface
- **WYSIWYM instead of WYSIWYG**
  - What You See Is What You Meant (Get)
  - note: that was the original approach

© 2005, D.I. Manfred Lindner

WWW Principles 4

- **Concept of hypertext**
  - Information (text documents) is structured in an hierarchical way. Retrieval of text documents will follow this hierarchy.
  - References (“links”) within an document allow access to other documents located at a different level of hierarchy
  - Allows a new way of navigating within text documents
  - Allows links to documents residing on other machines
  - Later expanded to hypermedia
    - Including graphics, audio and video
- **Hypertext and GUI (Browser)**
  - The base for the success of the World-Wide-Web

© 2005, D.I. Manfred Lindner

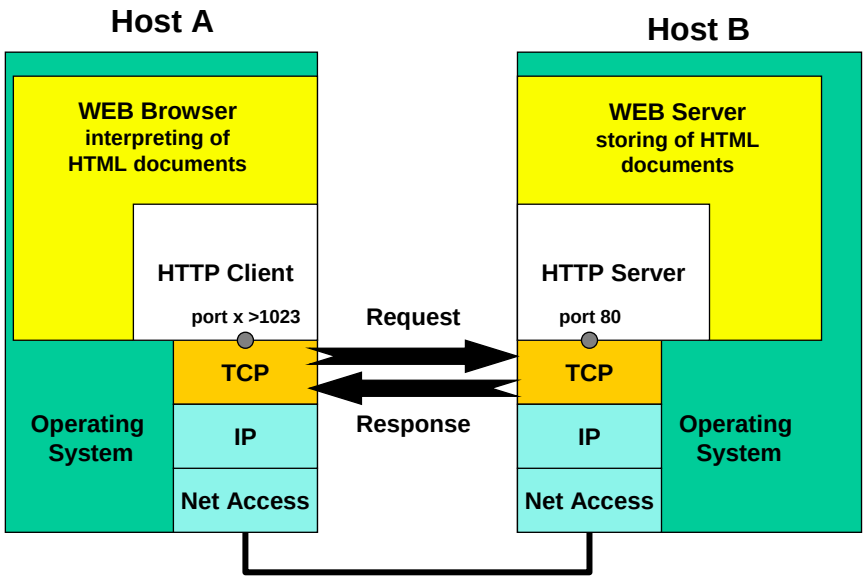
WWW Principles 5

- **Web Browser**
  - Can download and present HTML documents to the user
  - E.g. NCSA Mosaic, Netscape Navigator, Microsoft Internet Explorer, Opera, Mozilla´s Firefox
- **Web Browser use HTML Interpreter for the presentation of documents**
- **Web Browser use HTTP protocol for the download of documents**
  - Hypertext Transfer Protocol
  - Client - Server based
    - Browser as client, WEB Server - as server
  - Server accessible via well-known TCP port 80

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

HTTP - WWW Model



© 2005, D.I. Manfred Lindner

WWW Principles 6

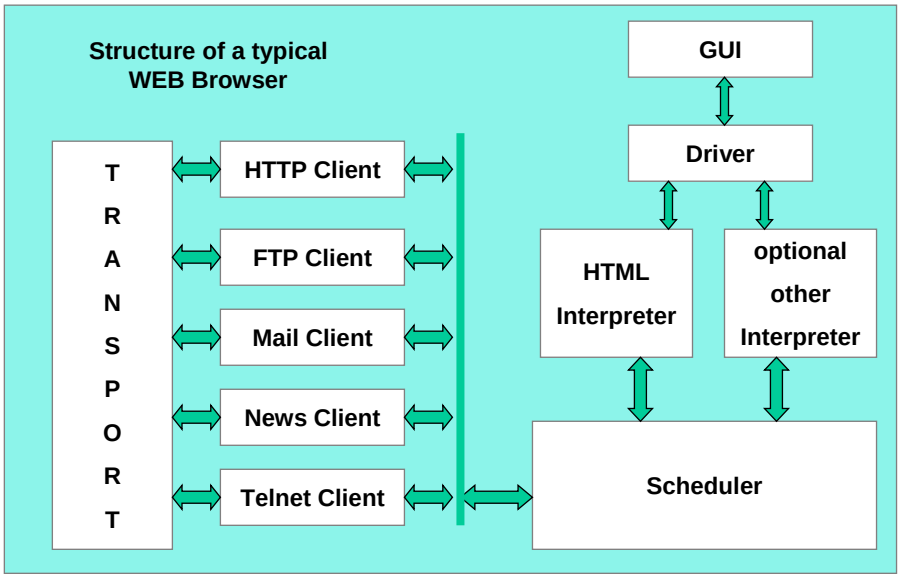
- **Web Browsing**
  - Usage of cache techniques to reduce network load and improve performance
    - Browser Cache
    - Proxy Server as caching system
    - Document attributes for decision what is newer or must be refreshed
      - Time and Date
      - Meta Information http-equiv “expires”
  - Good for static Web content
  - More complicated for dynamic Web content

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

L63 - Components of the WWW

WWW Principles 7



© 2005, D.I. Manfred Lindner

History (1)

- **1945**
  - First visions about hyperlinked texts
  - Vannevar Bush: "Memex"
    - A photo-electrical-mechanical device
    - Could create and follow links on microfiche
- **1960s**
  - Doug Engelbart: "oNLine System" (NLS)
    - Hypertext browsing, editing, and email
    - Invented the mouse
  - Ted Nelson: Creates the word "Hypertext"
  - Andy van Dam: Hypertext Editing System

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

History (2)

- **1980**
  - Tim Berners-Lee: "ENQUIRE" –Program
    - Allowed links to be made between arbitrary nodes
    - Each node had a title, a type, and a list of links
- **1989**
  - Tim Berners-Lee: Idea for a Hypertext Environment
    - Paper "HyperText and CERN"
    - Paper "Information Management: A Proposal"
- **1990**
  - September – Tim Berners-Lee received the order to invent a global hypertext system
  - Christmas – First demonstration of "World Wide Web", a web-browser+editor. Later called "Nexus".

History (3)

- **1991**
  - May – General release of WWW on central CERN machines
  - June – First "computer seminar" on WWW
  - August – Files available on the net by FTP
  - October – WAIS gateways installed
  - December – W3C announced
- **1992**
  - Several new browser efforts: Erwise, Viola
  - Introduction of CVS
  - 26 reliable web servers world-wide(!)

© 2005, D.I. Manfred Lindner

History (4)

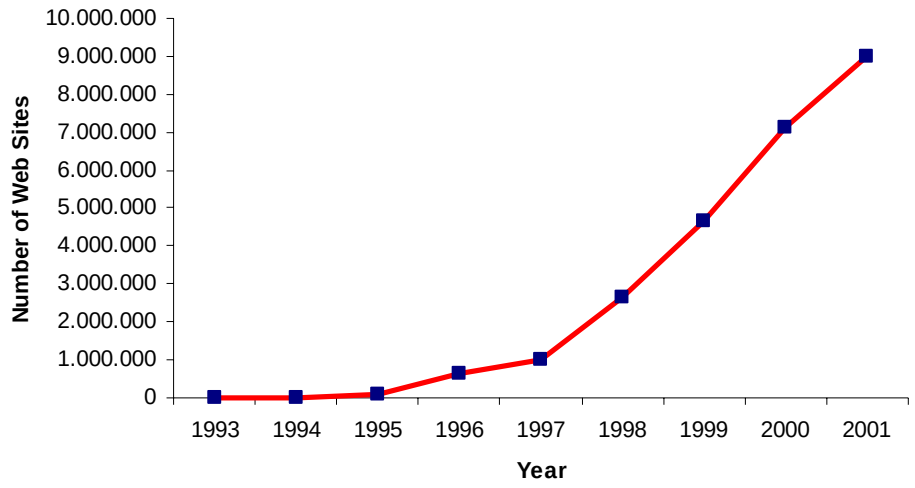
- **1993**
  - February – Alpha release of MOSAIC
    - by Marc Andreessen
  - April – A WWW Milestone: The CERN Declaration
    - WWW technology should be freely usable by anyone
  - September – MOSAIC for several platforms available
    - X, PC/Windows, Macintosh
  - October -- Over 200 known HTTP servers
    - First European Union web-project: WISE
  - December – World took notice about the Web
    - Articles in "The New York Times", "The Guardian", "The Economist" etc.
    - 623 web-pages world-wide

History (5)

- **1994**
  - May – First International WWW Conference
  - June – Over 1500 registered servers
    - Load on the first Web server (info.cern.ch) 1000 times what it has been 3 years earlier
  - December – CERN stops WWW development
    - Budget focuses primarily on LHC (Large Hedron Collider) accelerator
    - WebCore project moved to INRIA (Institut National pour la Recherche en Informatique et Automatique, FR)
  - December – First release of Netscape Navigator (successor of Mosaic)
- **After 1994**
  - The Internet has grown exponentially
  - The World Wide Web became the most exiting application

© 2005, D.I. Manfred Lindner

Growth of the WWW



© 2005, D.I. Manfred Lindner

WWW – Today

- **Exponential growth**
  - 1996: 100,000 websites
  - 1997: >1,000,000 websites
  - 1998: 3,700,000 websites and > 150 million users
  - 1999: 9,500,000 websites and > 259 million users
  - .....
- **W3 Consortium**
  - Maintains the WWW standards
  - Founded by Tim Berners-Lee at MIT
  - http://www.w3.org

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Introduction to HTML Tutorial

- **SELFHTML by Stefan Münz**
- **Excellent InfoBase about HTML**
- **Portal:**
  - http://selfaktuell.teamone.de/
  - http://selfhtml.teamone.de/
- **Download of selfhtml80.zip (7 MB)**
  - allows you to install tutorial on your PC

Agenda

- **Introduction WWW**
- **URL**
- **HTTP**
- **WWW Details**
- **HTML**

© 2005, D.I. Manfred Lindner

URL (RFC 1738)

- **Uniform Resource Locators**
  - Compact string representation for a resource available via the Internet
- **General term: URI**
  - Uniform Resource Identifier
  - "URL" was created by the IETF URI working group
- **Schemes**
  - Just as there are many different methods of access to resources, there are several schemes for describing the location of such resources
  - Are used to locate resources, by providing an abstract identification of the resource location

URL

- **Having located a resource**
  - A system may perform a variety of operations on the resource. Might be characterized by such words as “access”, “update”, “replace”, “find attributes”
- **In general, only the “access” method needs to be specified for any URL scheme**
- **URLs are written as follows:**
  - < scheme > **:** < scheme-specific-part >

    access method **:** path to the resource
- **Most URL schemes are hierarchical organized**
  - The components of the hierarchy are separated by "***/***"
    - e.g. ftp-, http-, and file-scheme

© 2005, D.I. Manfred Lindner

URL Scheme´s

- **Internet Assigned Numbers Authority (IANA) maintains a registry of URL schemes**
  - http Hypertext Transfer Protocol
  - ftp File Transfer protocol
  - gopher Gopher protocol
  - mailto Electronic mail address
  - news Usenet news
  - nntp Usenet news using News Network Transport Protocol access
  - telnet Reference to interactive sessions
  - wais Wide Area Information Servers
  - file Local file access

© 2005, D.I. Manfred Lindner

URL Scheme - Specific Part

- **General syntax**
  - //<user>:<password>@<host>:<port>/<url-path>
  - Some or all of the parts may be excluded
- **Examples:**
  - < URL: ftp://@host.com/ >
    - has an empty user name and no password
  - < URL: ftp://host.com/ >
    - has no user identification at all
    - If FTP Server request authentication
      - User: anonymous
      - PW: email-address of end user
  - < URL: ftp://foo:@host.com/ >
    - has a user name of "foo" and an empty password

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Examples: URL for HTTP

- **scheme and common-part**
  - http://<host>:<port>
    - if :<port> is omitted, the port defaults to 80; no user name or password is allowed
- **url-path:**
  - /<path>?<searchpart>
    - <path> is an HTTP selector, and <searchpart> is a query string
    - <path> is optional, as is the <searchpart> and its preceding "?"
    - if neither <path> nor <searchpart> is present, the "/" may also be omitted
    - Note: index.html is meant by default and finally presented

Examples: URL for FTP

- **scheme and common-part:**
  - ftp://<user>:<password>@<host>:<port>/
- **url-path:**
  - <cwd1>/<cwd2>/.../<cwdN>/<name>;type=<typecode>
    - Each of the <cwd> elements is to be supplied, sequentially, as the argument to a CWD (change working directory) command
    - If the typecode is "d", perform a LIST (name list) command with <name> as the argument, and interpret the results as a file directory listing
    - Otherwise (typecode “a” (Ascii) or “i” (Image)), perform a TYPE command with <typecode> as the argument,and then access the file whose name is <name> (for example, using the RETR command)

© 2005, D.I. Manfred Lindner

URL for Files

- **scheme and specific part**
  - file://<host>/<path>
    - where <host> is the fully qualified domain name of the system on which the <path> is accessible, and <path> is a hierarchical directory path of the form <directory>/<directory>/.../<name>
    - as a special case, <host> can be the string "localhost" or the empty string; this is interpreted as “the machine from which the URL is being interpreted”

Examples: URL for Mailto

- **scheme and specific part**
  - mailto:<rfc822-addr-spec>
    - where <rfc822-addr-spec> is (the encoding of an) addr-spec, as specified in RFC 822
    - within mailto URLs, there are no reserved characters

© 2005, D.I. Manfred Lindner

Examples: URL for Telnet

- **scheme and specific part**
  - telnet://<user>:<password>@<host>:<port>/
    - user, password, host and port as specified in previous slide
    - the final "/" character may be omitted
    - if :<port> is omitted, the port defaults to 23
    - the :<password> can be omitted, as well as the whole <user>:<password> part

Examples: URL for Gopher

- **scheme and common-part**
  - gopher://<host>:<port>/<gopher-path>
    - if :<port> is omitted, the port defaults to 70; No user name or password is allowed
- **gopher-path:**
  - <gophertype><selector>
  - <gophertype><selector>%09<search>
    - <gophertype> is a single-character field to denote the Gopher type of the resource to which the URL refers
    - the entire <gopher-path> may also be empty, in which case the delimiting "/" is also optional and the <gophertype> defaults to "1".
    - <selector> is the Gopher selector string; Gopher clients specify which item to retrieve by sending the Gopher selector string to a Gopher server

© 2005, D.I. Manfred Lindner

Examples: URL for NNTP

- **The nntp URL scheme is an alternative method of referencing news articles, useful for specifying news articles from NNTP servers (RFC 977)**
  - News Network Transfer Protocol
- **scheme and specific part**
  - nntp://<host>:<port>/<newsgroup-name>/<article-number>
    - where <host> and <port> are as described in previous slide
    - If <port> is omitted, the port defaults to 119
    - <newsgroup-name> is the name of the group, while the <article-number> is the numeric id of the article within that newsgroup

© 2005, D.I. Manfred Lindner

URL for WAIS

- **the WAIS URL scheme is used to designate WAIS databases, searches, or individual documents available from a WAIS database**
  - Wide Area Information System is described in RFC 1625
- **scheme and specific part**
  - wais://<host>:<port>/<database>
  - wais://<host>:<port>/<database>?<search>
  - wais://<host>:<port>/<database>/<wtype>/<wpath>
    - if :<port> is omitted, the port defaults to 210
    - the first form designates a WAIS database that is available for searching, the second form designates a particular search
    - <database> is the name of the WAIS database being queried
    - the third form designates a particular document within a WAIS database to be retrieved

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Agenda

- Introduction WWW
- URL
- HTTP
- WWW Details
- HTML

HTTP Principles

- Hypertext Transfer Protocol
  - First Version 1.0 (RFC 1945)
  - Current Version 1.1 (RFC 2616, 2817)
- Base for transport of WWW documents
  - Between client (Browser) and server (Web-Server)
- On top of TCP
  - Hence connection-oriented
  - Well-know server port 80
- Stateless
  - Client opens a TCP connection, requests a document, server responds with document, client closes TCP connection (remedy ->cookies)

© 2005, D.I. Manfred Lindner

HTTP Characteristics

- **Application-level protocol**
  - With the lightness and speed necessary for distributed, collaborative, hypermedia information systems
- **Object-oriented protocol**
  - Methods are applied on objects (sequentially)
- **HTTP messages consist of**
  - Header
  - Body
- **HTTP allows usage of a set of methods**
  - Methods specify the purpose of a request
  - Methods are applied on URLs included in the header

© 2005, D.I. Manfred Lindner

HTTP Header and Body

- **Header**
  - contains the URL, method, and parameters
  - HTTP v1.0 methods: GET, HEAD, POST
- **Body**
  - Contains user data described by a MIME header
    - MIME = "Multipurpose Internet Mail Extensions"
    - Also used by Internet Mail
  - User data can be
    - HTML information (= a web page)
    - Graphics, videos, sound-data, ...

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

HTTP Messages

- A client establishes a connection with a server and sends a request to the server containing
  - a request method
  - URL
  - protocol version
  - MIME-encoded data (optionally)
- The server's response contains
  - a status line
    - containing messages protocol version
    - and a success or error code
  - MIME-encoded data
    - Server information (e.g. expiration time for cache)
    - entity meta-information
    - and possible body content

© 2005, D.I. Manfred Lindner

HTTP v1.0 Methods

- **GET**
  - This method allows the client to retrieve the data which was determined by the request URL.
- **HEAD**
  - This method allows the client to retrieve meta-information about the entity which does not require to transfer the entity body
    - Check if document was changed since last retrieval and hence to be refreshed
- **POST**
  - This method allows the client to store documents on the server. The post function may be supported by the server.

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

HTTP v1.1 Methods

- **PUT**
  - This method is similar to the post method with one important difference which is the URL in post request identifies the resource that will handle enclosed entity whereas with put request the URL identifies the enclosed entity itself.
- **DELETE**
  - This methods requests that the server delete the source determined by the request URL.
- **TRACE**
  - Trace method allows the client to see how the message was retrieved at the other side for testing and diagnostic purposes (remote application-layer loopback)

© 2005, D.I. Manfred Lindner

HTTP Request Methods and CGI data

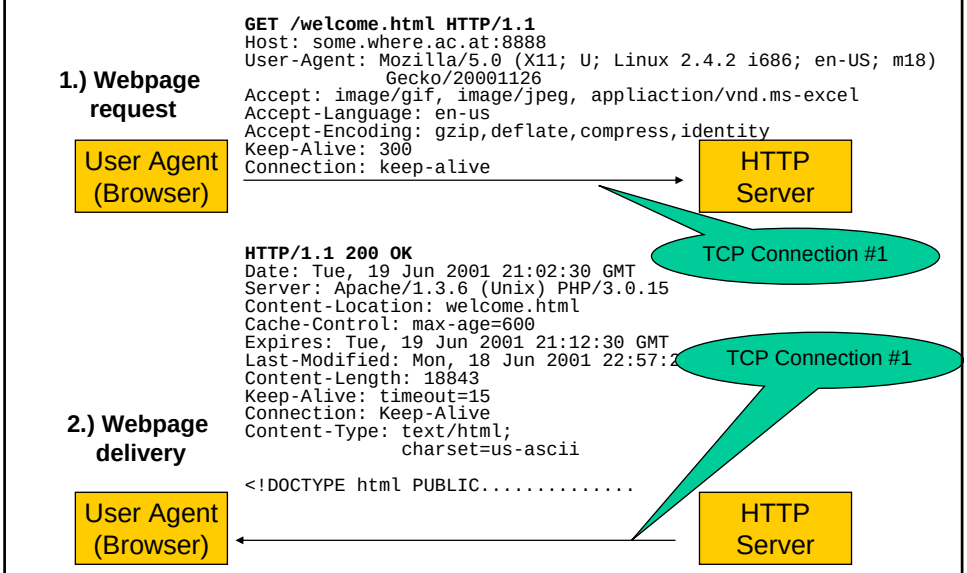
- **HEAD or GET request**
  - Only headers, no body
  - Form data for CGI is encoded in HTTP_QUERY_STRING
    - CGI script receives the environment variable QUERY_STRING which contains the whole information
- **POST request**
  - Header and body
  - Body contains user data (also form data)
- **Other differences**
  - HEAD request does not expect body in the response message
  - GET and POST accept responses with or without body

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

L63 - Components of the WWW

HTTP Communications (1)



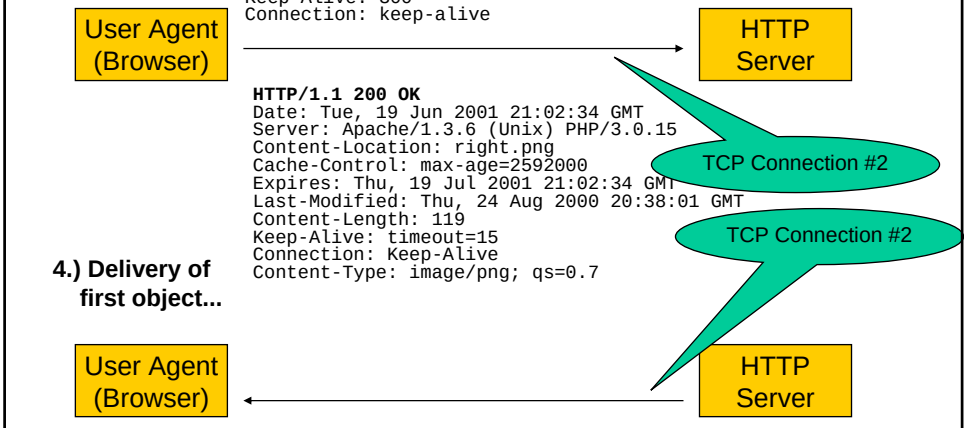

```
GET /welcome.html HTTP/1.1
Host: some.where.ac.at:8888
User-Agent: Mozilla/5.0 (X11; U; Linux 2.4.2 i686; en-US; m18)
            Gecko/20001126
Accept: image/gif, image/jpeg, appliaction/vnd.ms-excel
Accept-Language: en-us
Accept-Encoding: gzip,deflate,compress,identity
Keep-Alive: 300
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Tue, 19 Jun 2001 21:02:30 GMT
Server: Apache/1.3.6 (Unix) PHP/3.0.15
Content-Location: welcome.html
Cache-Control: max-age=600
Expires: Tue, 19 Jun 2001 21:12:30 GMT
Last-Modified: Mon, 18 Jun 2001 22:57:2
Content-Length: 18843
Keep-Alive: timeout=15
Connection: Keep-Alive
Content-Type: text/html;
              charset=us-ascii

<!DOCTYPE html PUBLIC.............
```

© 2005, D.I. Manfred Lindner WWW Components, v4.3 41

HTTP Communications (2)




```
GET /Icons/right HTTP/1.1
Host: some.where.ac.at:8888
User-Agent: Mozilla/5.0 (X11; U; Linux 2.4.2 i686; en-US; m18)
            Gecko/20001126
Accept: image/gif, image/jpeg, appliaction/vnd.ms-excel
Accept-Language: en
Accept-Encoding: gzip,deflate,compress,identity
Keep-Alive: 300
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Tue, 19 Jun 2001 21:02:34 GMT
Server: Apache/1.3.6 (Unix) PHP/3.0.15
Content-Location: right.png
Cache-Control: max-age=2592000
Expires: Thu, 19 Jul 2001 21:02:34 GMT
Last-Modified: Thu, 24 Aug 2000 20:38:01 GMT
Content-Length: 119
Keep-Alive: timeout=15
Connection: Keep-Alive
Content-Type: image/png; qs=0.7
```

© 2005, D.I. Manfred Lindner WWW Components, v4.3 42

© 2005, D.I. Manfred Lindner

HTTP Communications (3)

- **Current HTTP communications requires**
  - That the connection is established by the client prior to each request
  - And closed by the server after sending the response
- **If a webpage consists of several components**
  - Every component is downloaded by a separate TCP connection even if the component resides on the same machine !!!!
  - In HTTP v.1.1 concept of persistent connections
    - Objects of same type are retrieved via single TCP connection

HTTP 1.1 (RFC 2616)

- **HTTP/1.1 was developed**
  - to overcome version 1.0 problems
    - HTTP/1.0 does not sufficiently take into consideration the effects of hierarchical proxies, caching, the need for persistent connections, and virtual hosts
    - GET, HEAD or POST method only
    - Basic Authentication (Base64 coding of user-id, passw.)
  - to make HTTP a good Internet citizen
    - in HTTP/1.1, a persistent connection may be used for one or more request/response exchanges
  - to increase functionality
    - GET, HEAD, POST, PUT, DELETE, TRACE methods
    - caching control support to improve overall performance

© 2005, D.I. Manfred Lindner

HTTP 1.1 Associated Documents

- **RFC 2617 Basic and Digest Authentication**
  - Eliminates clear-text password of basic authentication
  - Uses cryptographic hashes (MD5)
  - Still the problem remains how to distribute a common secret safely between agent and server
- **RFC 2964, 2965 State Management Mechanism**
  - Specifies a way to create a stateful session with HTTP requests and responses
  - Based on two new headers
    - Cookie and Set-Cookie which carry state information between participating origin servers and user agents

Cookies

- **Originally, developed by Netscape Corporation**
- **To circumvent the stateless nature of HTTP**
  - HTTP servers respond to each client request without relating that request to previous or subsequent requests
  - Difficult to create services such as virtual shopping carts
- **"Cookies" introduce session information**
- **Two additional HTTP headers**
  - Set-Cookie
  - Cookie

© 2005, D.I. Manfred Lindner

What is a Cookie?

- **A Cookie**
  - Small piece of information (a string)
  - Basically, a special HTTP header (sent by the server)
  - Returned by the browser for each reconnection
- **String contains up to five attributes**
  - Name and value
  - Domain
  - Path
  - Lifetime
  - Security Info

Cookie Structure

- **Name and value**
  - The actual information (<name> = <value>)
  - Usually a session ID
  - Only these two parameters are mandatory!
- **Domain**
  - The domain the cookie is valid for
  - Tells browser about valid domain names whose servers would recognize this cookie
  - Promiscuous cookies are not allowed
  - Domain attribute must not contain top level domains

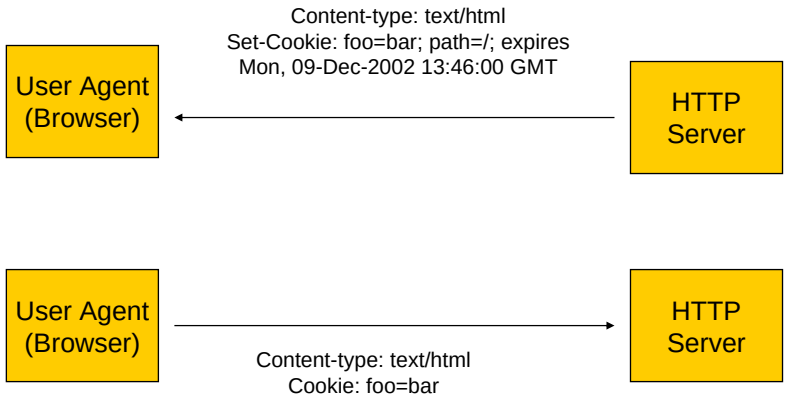
© 2005, D.I. Manfred Lindner

Cookie Structure

- **Path**
  - Scope of validity on the server's site
  - Pages outside of that path cannot read or use the cookie
  - E.g. "/ " means: valid for the entire site
- **Lifetime**
  - = "Expiration date"
  - If the lifetime is longer than the time the user spends at that site, then this cookie is saved on disk for future reference
- **Security Info**
  - Whether a secure connection must be established before sending this cookie (E.g. SSL)

© 2005, D.I. Manfred Lindner

HTTP Cookie Header Exchange



© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Cookie Risks

- **Note: Often used to track user behaviour !**
- **Advertisers smuggle cookies on your disk!**
  - Since most banners are references to other sites
    - E.g. "DoubleClick" (ad.doubleclick.net)
  - One of the most controversial issues of the Web!
- **Cookies can be turned off on most browsers**
  - But some sites might not continue its service
  - Possible remedy for Unix machines
    - Create a symbolic link from the cookies directory to /dev/null

Agenda

- **Introduction WWW**
- **URL**
- **HTTP**
- **WWW Details**
- **HTML**

© 2005, D.I. Manfred Lindner

Web-Browsers

- **Browsers**
  - Also known as "User Agents"
  - Use HTTP to access special encoded documents from a web-server
    - This documents are usually encoded in HTML
  - Should correctly display document content
- **Very complex applications**
  - Must handle different HTML versions
  - Cascading Style Sheets
  - JavaScript
  - Several graphic formats
  - Security options
  - Several additional services
    - FTP, MAIL, and USENET capabilities

Web-Browsers

- **Big browser companies intentionally write non-standard-conforming Browser software**
- **Who controls Web-standards?**
  - Led to "Browser-Wars"
  - Netscape vs. MS Internet Explorer
- **Other Browsers**
  - Opera
  - Lynx
    - Very fast text browser
  - Mozilla
    - Best implementation of W3C HTML standards
    - Most sophisticated HTML engine "Gecko"
    - OpenSource
  - etc.

© 2005, D.I. Manfred Lindner

Web-Browsers Dynamic 1

- **So far we have handled a static behaviour only**
  - The client requests a document and the server provides the content
- **To make the behavior more dynamic on the client side**
  - JavaScript / JScript
  - JavaApplets
  - ActiveX
  - Flash

Web-Browsers Dynamic 2

- **JavaScript (ECMA-Script)**
  - HTML extension and programming language by NetScape
  - Now standardized by ECMA-262
  - JavaScript programs are embedded as a source directly in an HTML document
    - Structures like frame, form, window are implemented
  - The program is executed on the client browser while the downloaded HTML is interpreted
    - Browser must be JavaScript enabled
  - JavaScript programs can control the behaviour of forms, buttons and text elements. In addition, they can be used to create forms whose fields have built-in error checking routines.
  - JScript is Microsoft´s answer to JavaScript

© 2005, D.I. Manfred Lindner

Web-Browsers Dynamic 3

- **JavaApplets**
  - Java is a platform-independent, object-oriented programming language inspired by C and C++ developed by SUN
  - As part of an HTML document Java programs are downloaded from the server and executed on the client within a restricted area (Java Virtual Machine (JVM) -> “sandbox” ; execution by interpreter),
  - A Java program started from inside an HTML (Web) page is called a Java Applet as opposed to a Java program, which is executed from the command line or otherwise on the local system
  - Java Applets were not supposed to touch anything local (outside of its JVM), and could only communicate back to theserver it was downloaded from.
  - With Java 1.1, applets can be signed with security keys and certificates and can therefore be authenticated. Thus, an applet can be authorized to access local resources, such as file systems, and it may communicate with other systems.

© 2005, D.I. Manfred Lindner

Web-Browsers Dynamic 4

- **ActiveX**
  - Is Microsoft´s answer to Java
  - Could be used to make MS-OS specific things visible and usable for the WEB
    - e.g. content from HTML via OLE to Excel-table
  - ActiveX Controls could be compared with Java Applets but there is no “sandbox” principle
    - User can specify barrier of trust only (similar to Internet Explorer)
  - Compiler must support **Component Object Model** (COM)
- **Flash**
  - Is a proprietary SW product which can be used for animation of Web-pages
  - It works just like a plugin by opening the corresponding program
- **Both ActiveX and Flash are not Internet standards**

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Web-Servers

- **Web-servers are basically http-servers**
  - Listen at port 80
- **Examples**
  - httpd NCSA
    - First http server
  - Apache
    - Most frequently used
    - Freeware
  - Internet Information Server (IIS)
    - Microsoft
  - Netscape Communications Server
- **Targets for DoS Attacks**
  - No solution against it!

Web-Servers Dynamic 1

- **So far we have handled a static behaviour only**
  - The client requests a document and the server provides the content
- **To make the behaviour more dynamic on the server side**
  - Common Gateway Interface (CGI)
  - PHP (Hypertext Preprocessor)
  - Active Server Pages
  - Servlets
  - Server-Sides-Include (SSI)
  - Java Server Pages (JSP)

© 2005, D.I. Manfred Lindner

Web-Servers Dynamic 2

- Common Gateway Interface (CGI)
  - Allows a Web server to execute a program that is provided by the Web server administrator, rather than retrieving a file.
  - CGI programs allow a Web server to generate a dynamic response, based on the client's input.
  - A variety of programming languages (C, Pascal, etc) can be used to develop programs that interface with CGI. In principle any compiled code could be executed, but normally only Scripts are used which are interpreted at the running time.
  - The most popular interpreter is PERL (Practical Extraction and Report Language) because programs are easily portable across platforms.

Web-Servers Dynamic 3

- PHP (Hypertext Preprocessor)
  - Alternative to CGI/PERL
  - PERL not optimized for dynamic Web pages
  - PHP is such an optimization
  - HTML documents stored on Web-servers can contain PHP programs.
  - If a client requests such a HTML document the server executes this program (interpreter), generates the final HTML-Code and transport the requested HTML document to the client
- Active Server Pages
  - Microsoft´s answer to PHP

© 2005, D.I. Manfred Lindner

Web-Servers Dynamic 4

- Servlets
  - In order to spare resources on clients and networks, Java Applets can be executed on the server rather than downloaded and started at the client. Such programs are then referred to as Servlets.
  - Servlets are Java Applets running at the server side
- Server-Sides-Include (SSI)
  - SSI is a technology which allows a Java enabled Web-server to convert a section of an HTML file into an alternative dynamic portion each time the document is sent to the client's browser.
  - This dynamic portion invokes an appropriate Servlet and passes to it the parameters it needs. Servlets may not be written in Java.
  - The replacement is performed at the server and it is completely transparent to the client.
  - Pages that use this technology have the extension .shtml instead of .html (or .htm).

Web-Servers Dynamic 5

- Java Server Pages (JSP)
  - This is an easy-to-use solution for generating HTML (or other markup languages such as XML) pages with dynamic content.
  - A JSP file contains combinations of HTML tags, NCSA tags (special tags that were the first method of implementing server-side includes), <SERVLET> tags, and JSP syntax.
  - JSP files have the extension .jsp.
  - JSP can be used to access reusable components, such as Servlets, JavaBeans (reusable Java objects), and Java-based Web applications. JSP also supports embedding inline Java code within Web pages.

© 2005, D.I. Manfred Lindner

How Search Engines Work

- **Web-pages may include HTML meta-tags**
  - Providing detailed information for search engine robots
- **This way, the author can influence the accessibility of his website**
- **Meta information may contain**
  - Description of site
  - Keywords about content in different languages
  - Supported languages
- **Examples**

```
<meta name="description" content="The best site for
  computer science and data communications">
<meta name="keywords" lang="en-us" content="HTML,
  Java, HTTP">
```


Today's Data Acquisition Problem

- **Number of web-pages grows exponentially**
- **The relation (useless information / useful information) goes to infinity**
- **What is a "good" site?**
- **How to find useful information?**

© 2005, D.I. Manfred Lindner

Search Engine Methods

- **Manual selection**
  - Evaluation problems
  - Does not scale
  - E.g. Yahoo!
- **Traditional approach: Index based search engines**
  - Every word is associated with a list of sites
  - Collection of lists is called an index
  - Evaluation according rules-of-thumb, e.g.
    - Count number of occurrences of given keywords
    - Occurrence at the top is better than at the bottom
    - Occurrences in special HTML tags (e.g. title or bold tag)

Evaluation Problem

- **Traditional index based search engines run into trouble**
  - Commercial sites exploit these principles
    - Include words with invisible colors or extremely small text-sizes
    - "Spamming"
  - Important sites do not always include the keywords
    - E.g. because of marketing policy rules
    - E.g. the IBM homepage does not include the word "computer" at all
  - Cannot take synonyms and polysemes into account
    - Polyseme: "Jaguar" = {a car, a cat, a football-team, ...}

© 2005, D.I. Manfred Lindner

Reference Based Rules

- **Look for semantic relations!**
  - Links are actually recommendations!
- **Divide the Web into**
  - Nodes (= sites with references, aka "Hubs")
  - Sources (= sites with information, aka "Authorities")
- **Good sources**
  - are referenced by good nodes
- **Good Nodes**
  - reference to good sources
- **Semantic information network can be created iteratively**

© 2005, D.I. Manfred Lindner

Reference Based Rules

- **Search engines try to find a subset of nodes which are densely linked**
  - Automatic separation in categories
- **Principle is similar to scientific paper evaluation**
  - "Impact factor" (= number of citations)
  - Used by "Science Citation Index"
- **Example: Google**
  - Jumps randomly from site to site by following links
  - Counts the number of site-hits
  - Evaluation: Sum of value of sites that reference to _this_ site

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Agenda

- **Introduction WWW**
- **URL**
- **HTTP**
- **WWW Details**
- **HTML**

Hypertext

- **The invention of the Web coined three new terms:**
  - The Link (or Hyperlink)
  - The Hypertext
  - The Browser
- **Hypertext**
  - Network of text-pages, associated via hyperlinks
- **Browsers**
  - Accesses hypertext from web-servers via HTTP
  - Interprets, formats, and displays the encoded text
    - Usually HTML encoded

© 2005, D.I. Manfred Lindner

General Document Markup Basics

- **Documents can be described using:**
  - Presentational Markup
  - Procedural Markup
  - Generic Markup
  - Standardized Markup

Presentational Markup

- **Involves highlighting, position, and other formating information**
- **No structure information about content**
- **No definite identification of text-elements possible**
- **Drawbacks**
  - Difficult to maintain
  - Only trivial text processing possible
  - Not database-ready
  - Bad portability
- **Advantage**
  - Transparent to the author (no visible tagging)
- **Example**
  - MS Word

© 2005, D.I. Manfred Lindner

Procedural Markup

- **Text contains typesetting directives**
  - Like a programming language
- **Advantages**
  - Device independent formatting languages
  - Typically used for high quality typesetting
- **Drawbacks**
  - Again, no structural information
  - Complex coding
  - Difficult to read for humans
  - Bad portability
- **Examples**
  - TeX

© 2005, D.I. Manfred Lindner

Generic Markup

- **Goal: Simplification of the complex coding of procedural markup languages**
  - High-level generic markup commands represent a number of procedural markup commands ("macro" or "gencode")
- **Advantages**
  - Structural information about text
  - Database-ready
  - Easy to maintain
  - Electronic text processing possible
- **Drawback**
  - Rather bad portability (but easier than presentational markup)
- **Example**
  - LaTeX

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Standardized Markup

- **Simple mnemonic tags**
  - Describe structure and content
  - Identify information (!)
- **Advantages**
  - Good portability because of strict DocType validation
  - Easy to maintain
  - Database ready
  - Powerful text processing possible
  - Machine readability for many decades
  - Ideal interchange format between different environments
  - Extensible
- **Example**
  - SGML

© 2005, D.I. Manfred Lindner

SGML

- **Standard Generalized Markup Language (SGML)**
  - Revolutionary approach to describe document content
  - Also used to create markup languages
- **SGML applications**
  - HTML
  - XML
  - WML
- **Arbitrary tags can be defined**
  - By specifying a so-called "DocType“
  - SGML documents must refer to a DocType
  - Parsers validate document against this DocType

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

SGML

- **Basic Idea of SGML: Tagging**

```
<some_tag>
        blah, blah, blah
</some_tag>
```

- **DocType**
  - Is included at the top of a document
  - <! DOCTYPE .....>
    - Contains either the element specifications
    - Or a link to a doctype file (DTD)

SGML

- **SGML document must be "well-formed"**
  - Documents must conform to their DocType
  - One or more elements must be included
  - Exactly one root element must be included
  - Elements must be properly nested
- **Applications that recognize invalid SGML code**
  - Must stop the processing job (!!!)
- **Additionally, display or print information can be provided**
  - Using Stylesheets
  - DSSSL
    - Document Style Semantic Specification Language

© 2005, D.I. Manfred Lindner

HTML

- **The Hyper Text Markup Language (HTML)**
  - Is the most famous SGML application
  - HTML is completely defined using SGML
  - That is, HTML corresponds to an SGML DocType defining all possible HTML tags
  - ISO Standard 8879:1986
- **HTML provides**
  - Content tagging
  - Inline graphics
  - Stylesheets
  - Inclusion of scripting languages

DHTML

- **Dynamic HTML (DHTML)**
  - Commonly used abbreviation
  - But DHTML does not exist (!!!)
- **DHTML is short for**
  - HTML 4.0
  - CSS
  - DOM
  - Javascript

© 2005, D.I. Manfred Lindner

CSS

- **Cascading Style Sheets (CSS)**
  - W3C recommended method to describe procedural markup
  - CSS specifies how things should look like opposed to HTML which specifies the structure only
  - Additional feature to content tagging
  - Several presentation medias supported
- **HTML-Documents contain links to CSS files**
  - Otherwise, style information can be included in-line
- **Powerful features possible, such as**
  - Speech support for handicapped people
  - Transparent and fixed elements

© 2005, D.I. Manfred Lindner

HTML, XML

- **HTML is one markup language**
  - Defined by DTD (Document Type Definition) using SGML
  - So a certain HTML Version is not extensible
  - Extensions (e.g. new tags) must be coordinated by W3C
  - No strict separation between content and format -> primarily human as consumer
- **XML is a family of markup languages**
  - Every XML document contains ist own DTD which allows self defined tags (X ... Extensible)
  - XML usable universal data format
  - Strict separation between content and format -> machine and human are consumers

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

XHTML

- **XML is reduced SGML**
- **Many new Web standards are XML applications !!!**
- **Main problem of Browsers: Fault Tolerance**
  - Browsers always try to display the webpage
  - Even in case of malformed HTML codes
    - Against SGML/XML rules !!!
- **New portable web-clients (mobile phones, etc)**
  - Require small and smart browsers
  - Web-designers must follow W3C style
  - XHTML

XHTML

- **Simple transition from HTML 4.01**
  - All tags must be written in lower case
  - parameters must be double-quoted ("")
  - Tags must be well-nested
  - Every tag needs a closing tag
- **Goal: Support of lightweight Browsers for mobile devices**
  - Simple XML inclusion
  - CSS

© 2005, D.I. Manfred Lindner

WAP/WML

- **Wireless Application Protocol (WAP) consists of**
  - Wireless Markup Language (WML) specification
  - WMLScript specification and WMLSript Virtual Machine
  - Wireless Telephony Application Interface (WTAI) specification
- **Controlled by the WAP Forum**
  - Industry association
- **Designed for small wireless terminals**
  - Using MicroBrowsers
    - Minimal demand on HW, Memory and CPU
    - Displays information written in a restricted mark-up language called WML
- **WML**
  - Based on XML
  - Much stricter than HTML

© 2005, D.I. Manfred Lindner

XSL

- **eXtensible Stylesheet Language (XSL)**
  - The Style Sheet of XML
  - Similar as CSS is the style sheet of HTML
- **Since HTML uses predefined tags**
  - CSS is suitable for HTML
- **But XML does not know about predefined tags**
  - New Style Sheet specification language necessary
  - XSL is far more sophisticated than CSS
- **XSL defines methods**
  - For transforming XML documents
  - For defining XML parts and patterns
  - For formatting XML documents

© 2005, D.I. Manfred Lindner

© 2005, D.I. Manfred Lindner

Schema

- **XML Schemas**
  - Should replace DTDs
  - Originally proposed by Microsoft
- **Advantages**
  - Easier to learn than DTDs
  - Extensible
  - Written in XML
  - Support for data types
  - Support for namespaces

DOM

- **Document Object Model (DOM)**
  - Tree view of the XML document
  - Platform and language neutral interface
  - Using the DOM a program can access and manipulate the content, structure, and style of a XML document
- **Scripting Languages can access variables of the DOM**
  - Accessible in an object-oriented fashion
    - Objects and properties

© 2005, D.I. Manfred Lindner