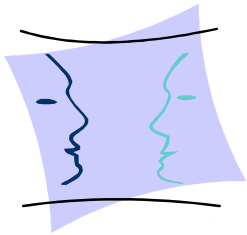


http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mühlhäuser



Scope

Contents



Multimedia-Systems: Documents

Prof. Dr.-Ing. **Ralf Steinmetz**

Prof. Dr. **Max Mühlhäuser**

MM: TU Darmstadt - Darmstadt University of Technology,

Dept. of of Computer Science

TK - Telecooperation, Tel.+49 6151 16-3709,

Alexanderstr. 6, D-64283 Darmstadt, Germany, max@informatik.tu-darmstadt.de Fax. +49 6151 16-3052

RS: TU Darmstadt - Darmstadt University of Technology,

Dept. of Electrical Engineering and Information Technology, Dept. of Computer Science

KOM - Industrial Process and System Communications, Tel.+49 6151 166151,

Merckstr. 25, D-64283 Darmstadt, Germany, Ralf.Steinmetz@KOM.tu-darmstadt.de Fax. +49 6151 166152

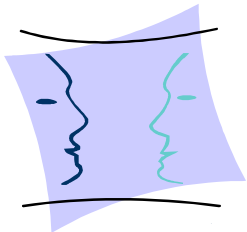
GMD -German National Research Center for Information Technology

httc - Hessian Telemedia Technology Competence-Center e.V.



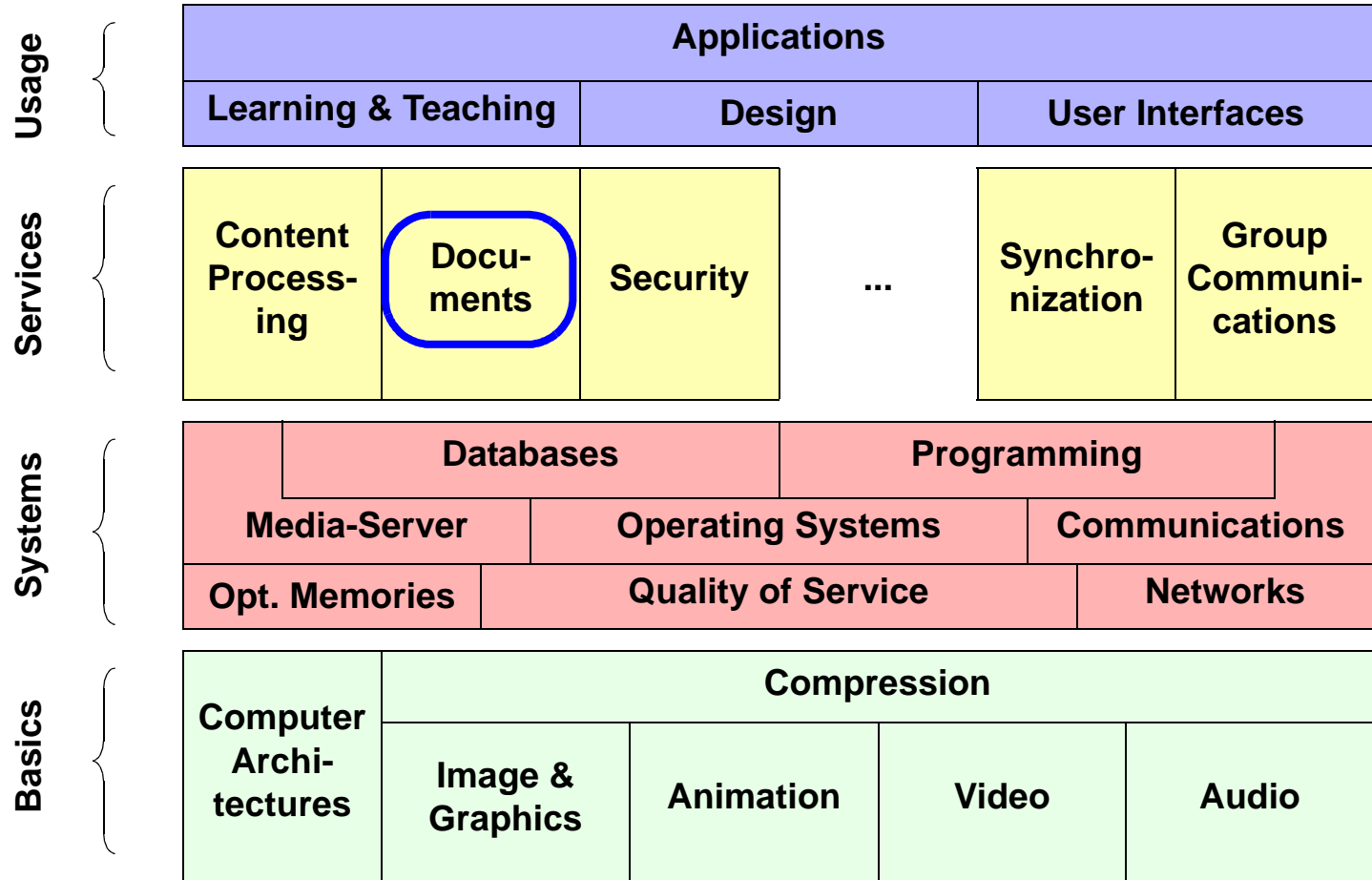
Scope

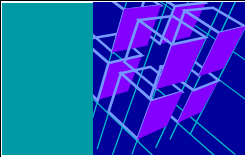
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
 © R. Steinmetz, M. Mülhäußer



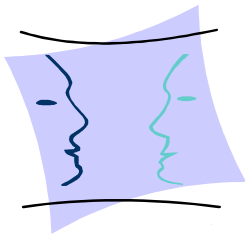
Scope

Contents





<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents

Contents

1. Documents

Definition: Document

Definition: Multimedia Document

2. Hypertext and Hypermedia

History of Hypertext

History of Hypertext

Hypermedia

3. Standard Generalized Markup Language (SGML)

4. The World Wide Web

WWW Architecture

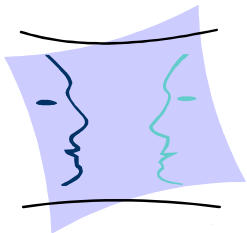
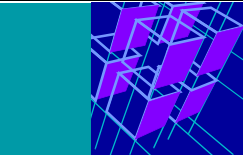
Browsers

HTTP / 1.0: (RFC 1945)

Forms and CGI (Common Gateway Interface)

Java: Fundamentals





Scope

Contents



1. Documents

Definition: Document

- **is a set of structured information:**
 - prerecorded or
 - generated at presentation time
- **covering different media**
- **intended for human perception**
- **accessible to processing by a computer**

Definition: Multimedia Document

Document characterized by at least:

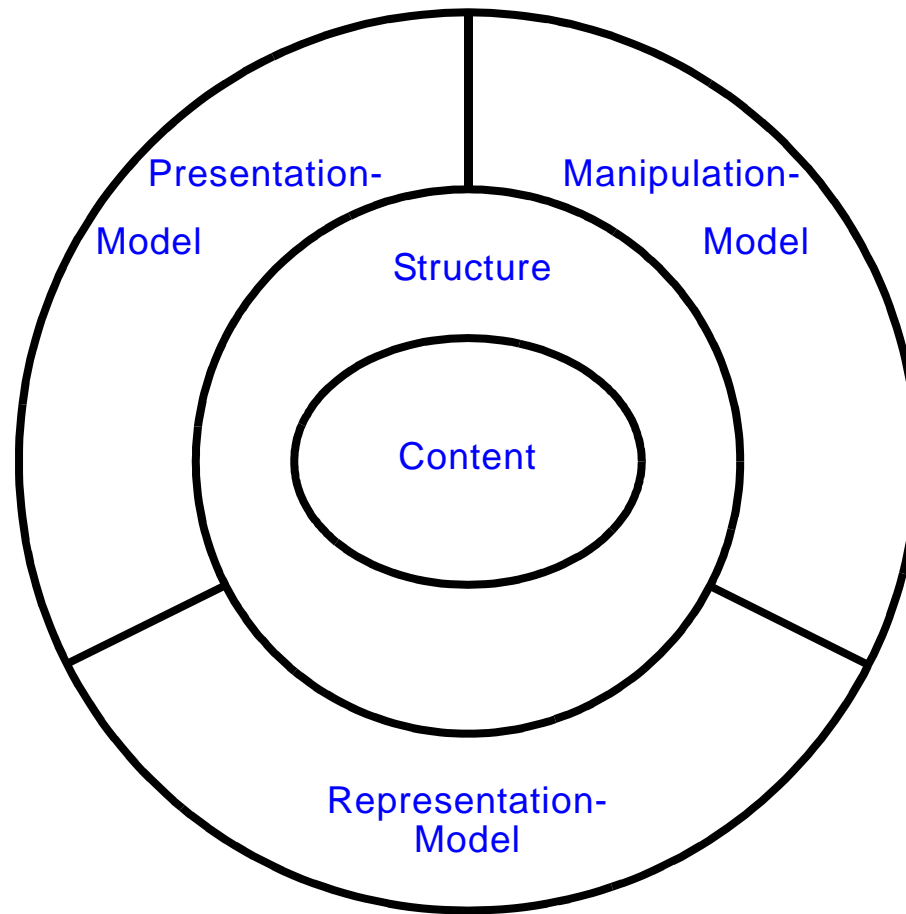
- **one continuous (time dependent) and**
- **one discrete (time independent) media**

Integration determined by close relationship between components that are handled in different manner

Basic idea:

- **Abstraction from different views and means of manipulation**
- **allows for integrated and uniform description and handling**

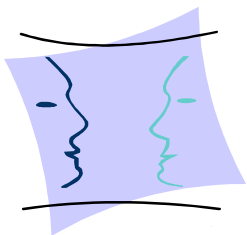
Document Architecture



Examples:

- **Standard Generalized Markup Language (SGML)**

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlpäuser



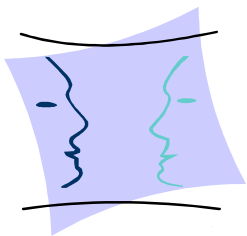
Scope

Contents



Architecture

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents

Presentation / User Interface:

- User input
- Display of nodes and edges

Hypermedia Abstract Machine:

- Structure of the content
- Knowledge about references, data structures, attributes

Database:

- Storage of data
 - No semantics
- Storage management
 - Multiuser access, persistency, fault tolerance, ...



2. Hypertext and Hypermedia

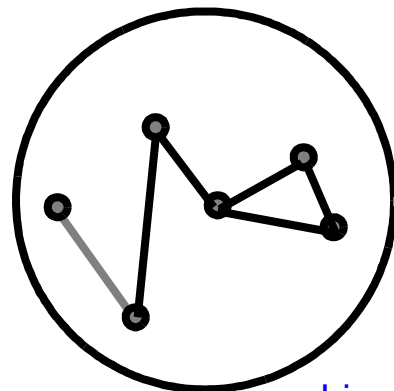
Distinction:

- transfer format (e.g. a classic book) - linear
- but we don't think that way - (associations, hierarchies, trees, semantical networks)

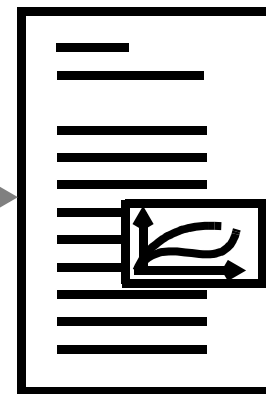
Conclusion:

- describe information structure in an appropriate way
- find adequate means of presentation
- transmit / exchange
 - content
 - structure

Author / Content Provider

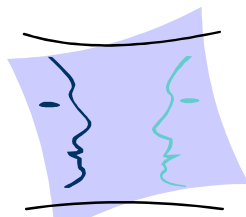
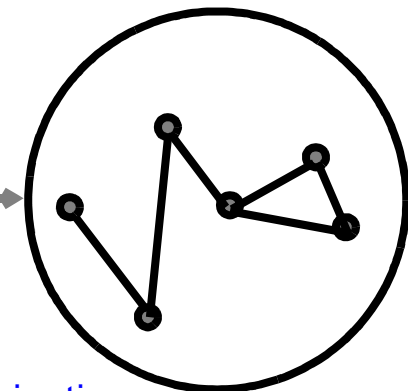


Linearisation



De-Linearisation

Recipient



Hypertext and Hypermedia (cont.)

Process somewhat different from traditional authoring / publishing

- **Author associates certain navigation structure within content**
- **Linearisation before information exchange**
- **Recipient gets impression of intended structure and may navigate individually**

Non-linear linkage of information:

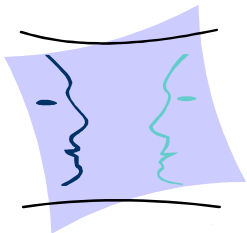
- **More than only one sequential order to retrieve information**
- **Reader determines 'reading path'**

Association through references (links):

- **Reference**
 - contains information
 - connects pieces of information
- **Producer determines the references**
- **Reader selects references**

Content can / must be separated from representation

- **Corporate identity**
- **limited storage or transfer bandwidth**
- **limited presentation support**



Scope

Contents



History of Hypertext

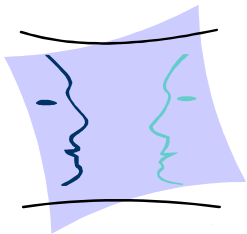
Memory Extender „Memex“ (1945):

- **Microfilm-based**
- **Several windows**
- **Associative index, linking areas together**
- **Vannever Bush:**
 - 1932/33: first papers
 - 1945: „As we may think“
- **No realizations**

Augment/NLS (1962-1976):

- **Doug Englebart at SRI (Stanford Research Institute)**
- **Project: Augmentation of human abilities**
- **Part of it is NLS (oN Line System):**
 - Common document storage for all scientists in this project
 - Using links
 - Incorporated about 100 000 items (nodes)

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents





<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer

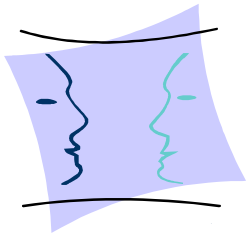
History

Xanadu (1965):

- **Ted Nelson: first use of the term “Hypertext“**
- **Intended to contain all information ever written**
- **No implementation**

Hypertext Editing System (1967) and FRESS (1968):

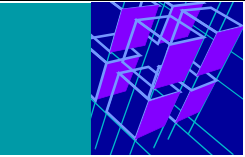
- **Andries van Dam: implementation of the first Hypertext System**
- **Brown University**
- **Documentation of the Apollo mission**
- **Later: file retrieval and editing system**



Scope

Contents



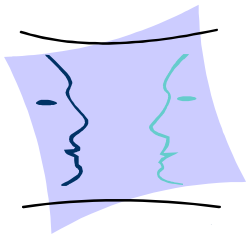


History

Aspen Movie Map:

- **First real Hypermedia system**
- **Developed by Andrew Lippman at MIT**
- **'Surrogate Travel' through the city of Aspen**
- **Pictures were taken at all streets:**
 - 90 degree angles (front, back, right, left)
 - Every 3m
- **Stored on videodisk**
- **Driving is simulated with playback of at most 2 pictures/s (~ 21.6 km/h)**
- **Joystick control**

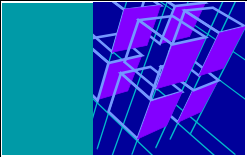
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



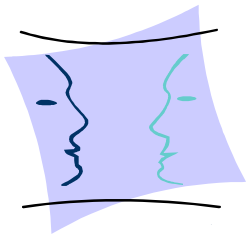


History

Since 1985: many Hypertext systems

- **Symbolics: Document Examiner (1985):**
 - first product
 - one application: documentation of the Symbolics workstation
- **Xerox: NoteCards**
- **Brown University: Intermedia**
- **Office Workstation Limited: Guide**
- **Apple: HyperCard**

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Hypertext: Application Areas

Computer applications:

- **Online documentation**
- **Extended help function**
- **World Wide Web (WWW)**

Business Applications:

- **Repairing and operation instructions**
- **Exhibition and product catalogues, advertisement**

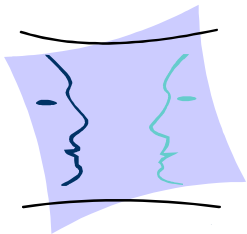
'Intellectual' Applications:

- **Organization of ideas and brainstorming**

General:

- **Whenever the relation of information may be non-sequential**

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer

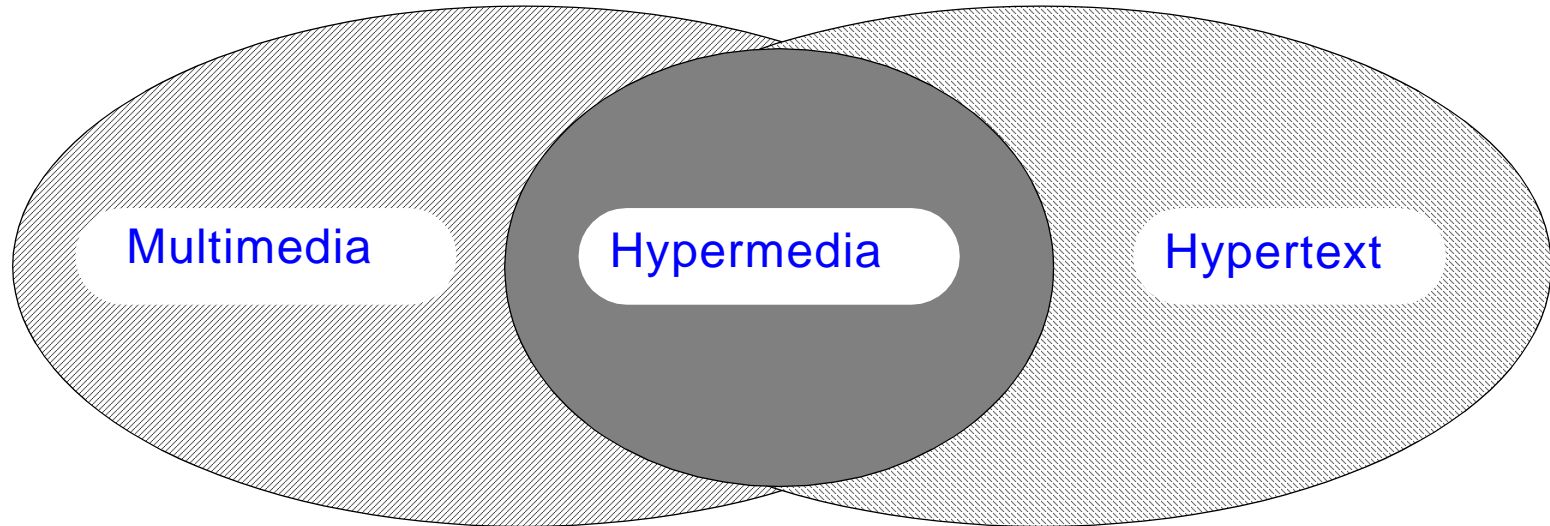


Scope

Contents



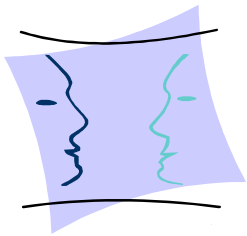
Hypermedia



Inclusion of additional media:

- **graphics, photos, audio clips, animations, video sequences**

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



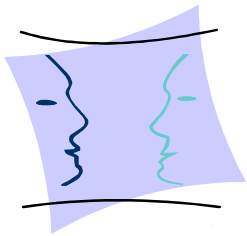
Scope

Contents



Example of a Hypertext System (1)

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlpäuser



Scope

Contents



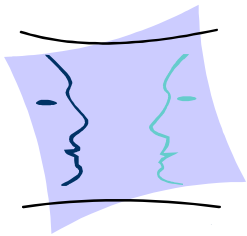
time since you where here: 7 days

LECTURE
HYPERTEXT

help quit

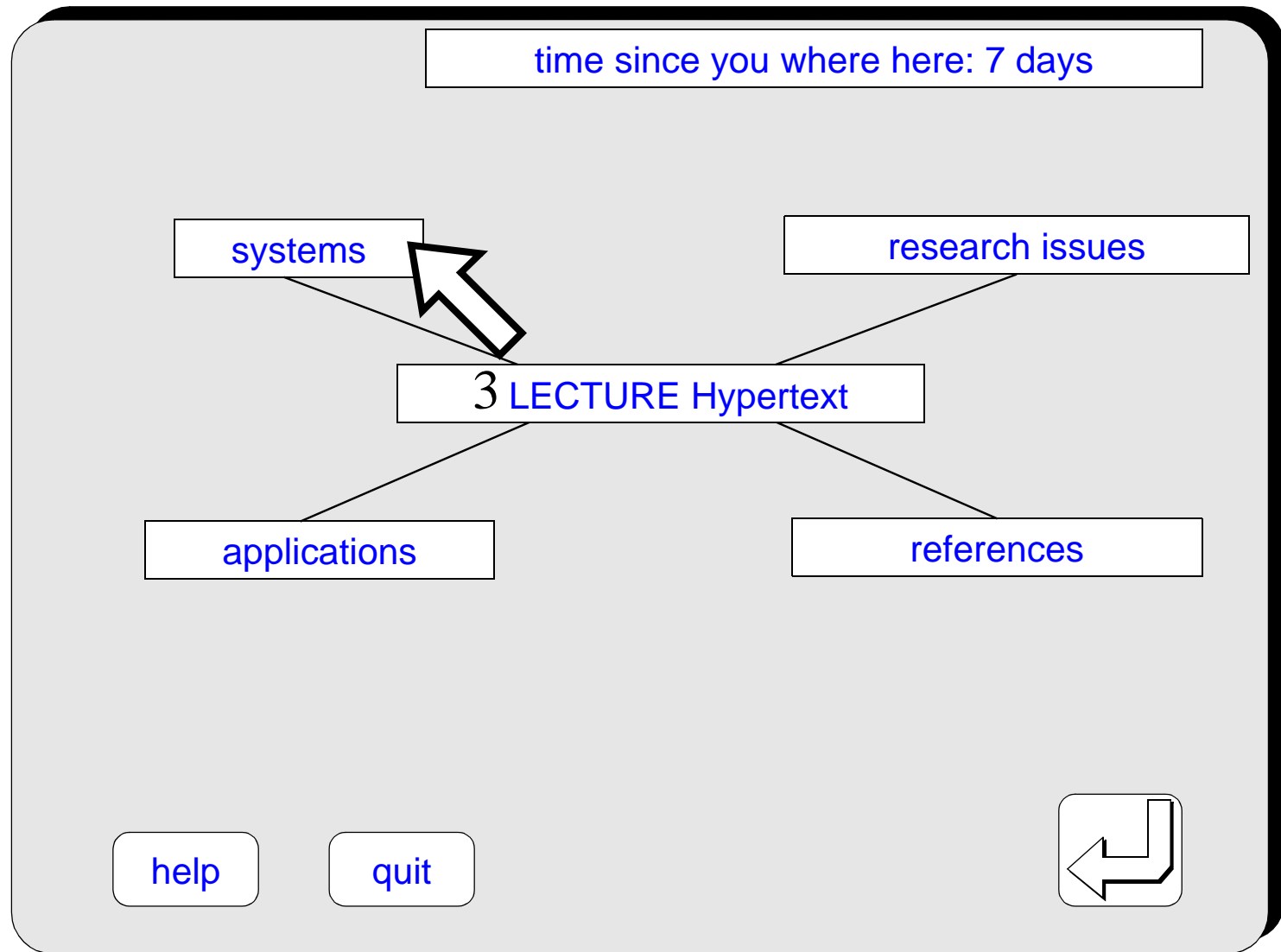
Example of a Hypertext System (2)

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



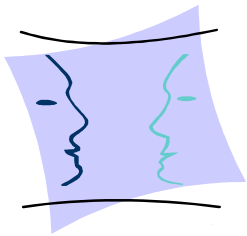
Scope

Contents



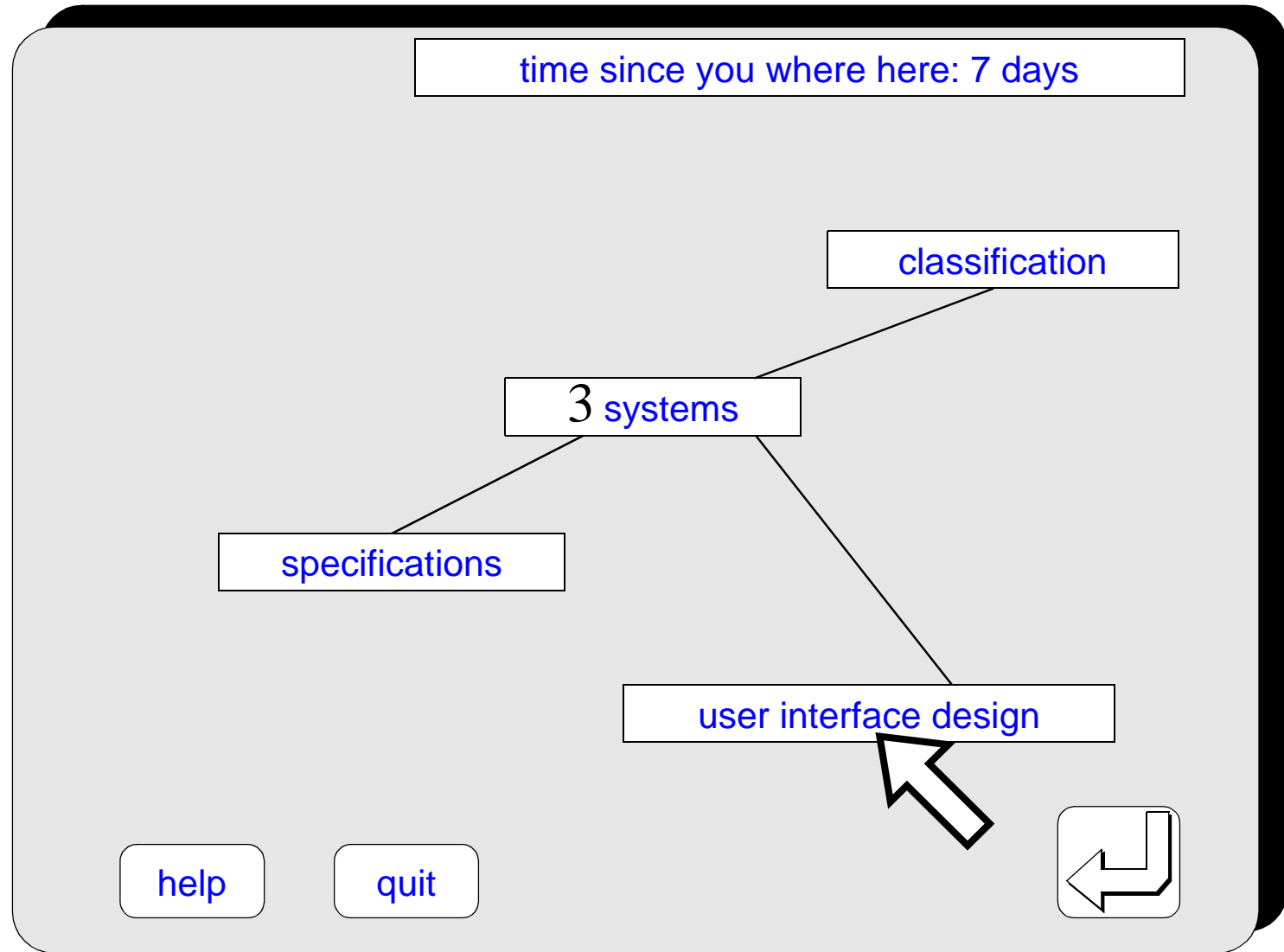
Example of a Hypertext System (3)

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



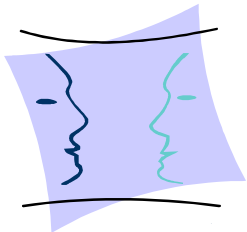
Scope

Contents



Example of a Hypertext System (4)

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhåuser



Scope

Contents



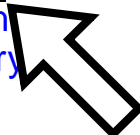
time since you where here: 2 min

User Interface Design

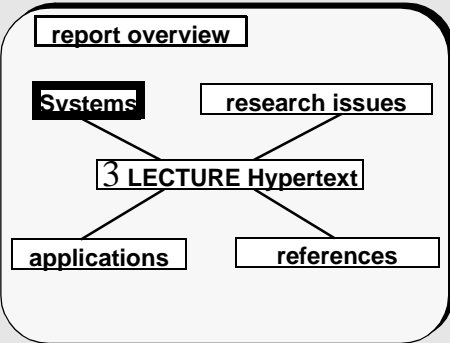
Studies showed that users often loose orientation while navigating through a HYPERTEXT system.

In order to guarantee permanent orientation in the system the following functions are necessary:

- display overview diagrams
- creating history files
- backtracking

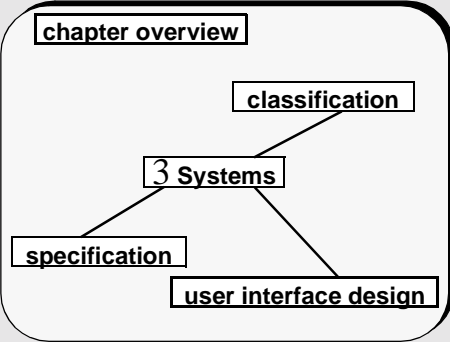


report overview




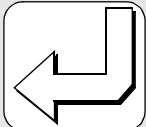
```
graph TD; A[3 LECTURE Hypertext] --- B[Systems]; A --- C[research issues]; A --- D[applications]; A --- E[references];
```

chapter overview



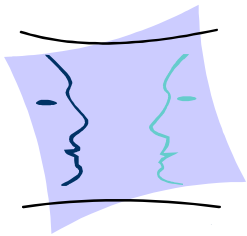
```
graph TD; A[3 Systems] --- B[classification]; A --- C[specification]; A --- D[user interface design];
```

help quit 



Example of a Hypertext System (5)

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlpäuser



Scope

Contents

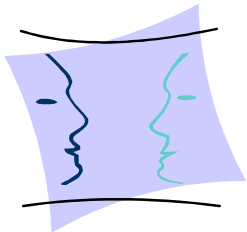


time since you where here: never

help quit

Example of a Hypertext System (6)

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



time since you where here: 2 min

User Interface Design

Studies showed that users often loose orientation while navigating through a HYPERTEXT system.

In order to guarantee permanent **orientation** in the system the following functions are necessary.

- **display overview diagrams**
- **creating history files**
- **backtracking**

report overview

```
graph TD; A[Systems] --- B[3 LECTURE Hypertext]; C[research issues] --- B; B --- D[applications]; B --- E[references]
```

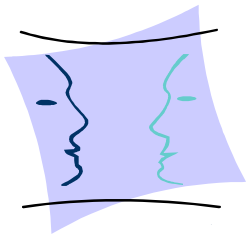
chapter overview

```
graph TD; A[classification] --- B[3 Systems]; B --- C[specification]; B --- D[user interface design]
```

help quit

Example of a Hypertext System (7)

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



time since you where here: never

HISTORY LIST

(time since you read the article) click on the name of an article to jump back to it

5 min	front cover of LECTURE Hypertext
2 min	overview LECTURE Hypertext
4 min	chapter overview map: systems
2 min	chapter systems: user interface design

help quit

3. Standard Generalized Markup Language (SGML)

Evolved from the IBM internal GML project

Strongly supported by American publishers:

- Authors define titles, tables in uniform way
- Publisher determines layout

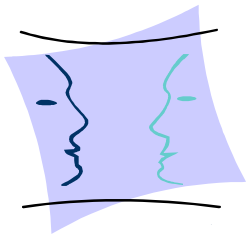
Basic ideas:

- Author uses tags to mark parts of the text
- SGML determines how tags have to look like
- User groups agree on the meaning of the tags
- Formatter generates document layout from tags

SGML defines:

- Syntax
- Not semantics

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



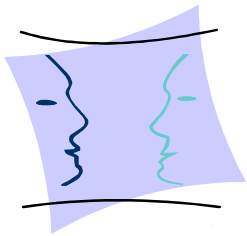
Scope

Contents



SGML: Concept

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Relationship between:

- **Document**
 - data content
 - markups
- **Document Type Definition (DTD):**
 - set of markup declarations
 - define
 - element types
 - attributes of elements
 - hierarchical relationships between elements
- **Procedures**
 - specify the document processing
 - correspond to functions of the formatter

Example:

```
<title> MM Systems </title>  
<author> Felix Gatou </author>  
<site> IBM </site>  
<summary> This paper ...  
</summary>  
...
```

SGML: Markup Categories

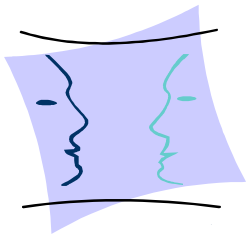
Descriptive Markup (Tags):

- **Defines the structure of the document**
- **Form:** `<tag> text </tag>`
- **Example:** `<paragraph> This is a paragraph ... </paragraph>`

Entity Reference:

- **For symbol substitution and file, data set, and variable imbedding**
- **Example:** `&Ae.rger ...` means ... Ärger

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



SGML: Markup Categories

Markup Declaration:

- **Can define the entities referred to by the Entity References:**

- **example:** `<!ELEMENT Ae (...)>`

- **Can define rules for the structure of the document:**

- **example:**

```
<!Element paper (preamble, body, postamble)>
```

```
<!Element preamble (title, author, side)>
```

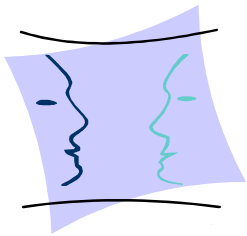
```
<!Element title (#CDATA)> --character data
```

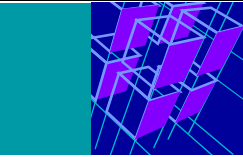
```
<!Element body (...)>
```

...

Processing instructions:

- **Instructions used by other programs, e.g. a formatter**
- **E.g. for the imbedding of different media**





SGML and Multimedia

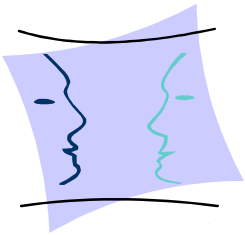
Imbedding of Multimedia Data:

- **As external document parts from separate files**
- **Originally embedded as CGM (Computer Graphics Metafile) graphic**
 - concrete data referred by using NDATA
- **Now: HTML, HyTime**

Example:

```
<!ATTLIST video id ID #IMPLIED>
<!ATTLIST video synch #IMPLIED>
<!ELEMENT video (audio, movpic)>
<!ELEMENT audio (#NDATA)> -- non-text media
<!ELEMENT movpic (#NDATA)> -- non-text media
...
<!ELEMENT story (preamble, body, postamble)>
```

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



SGML Conformant Languages

HTML

- Hypertext Markup Language
- For description of hypertext / hypermedia in the World Wide Web (WWW)

HyTime

- ISO standard for structured presentation of hypermedia information

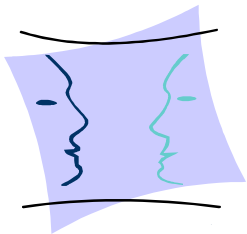
SMDL

- Standard Music Description Language

XML

- Extensible Markup Language
- driven by W3C consortium (XML working group)
- designed to enable the use of SGML on the WWW
- note:
 - metalanguage
 - lets you design your own markup language
- its: **SGML-- rather than HTML++**

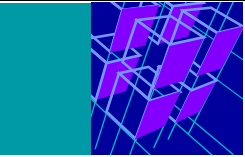
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents





4. The World Wide Web

Overview

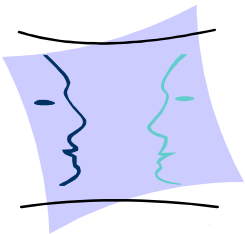
World Wide Web (WWW, W3, „The Web“):

- Framework for hyperlinked documents
- Huge collection of documents spread over the whole Internet
- The layman's opinion: „WWW is the Internet“

History

- 1989 (March) Tim Berners-Lee (CERN, Geneva) published first Ideas
- 1993 (Begin of)appr. 50 Web Servers
- 1993 (Feb)Mosaic distributed first version as shareware
- 1994CERN and MIT founded W3 organization (W3O)
Inria joint to become W3-Consortia (W3C)
objective: to promote WWW (see <http://www.w3.org>)
- 1995 (Nov)html defined as HTML 2.0. in RFC 1866
(<ftp://ds.internic.net/rfc/rfc1866.txt>)
- still growing at an exponential speed

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Some Additional Background

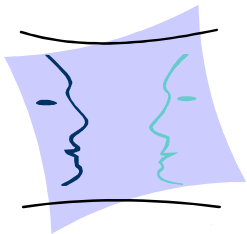
Netscape (history & role)

- **1952** Jim Clark founded Silicon Graphics
- **1993** Mark Andreessen developed Mosaic as “front end” at US National Center for Supercomputer Applications (NCSA)
- **1994 (April)** Jim Clark left Silicon Graphics
Clark & Andreessen founded Netscape Communications
- **1996** Netscape Browser appr. 80% market share

Sun Microsystems

- **appr. 1994** Java as plug-in (applet) defined for additional functionality at browser’s site
- **today** Java licensed by all most prominent companies in this domain

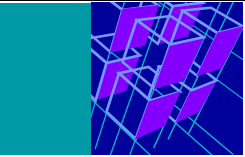
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents





WWW Architecture

Paradigm

- **Client-server architecture:**

Servers provide documents

- **written in HTML language**

Clients access and display documents

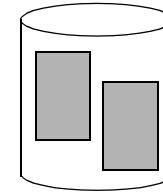
- **comprising multiple media**
- **by means of browsers**

Communication via special protocol

- **HTTP on top of TCP/IP**

Server:

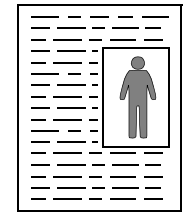
Document Files (HTML):



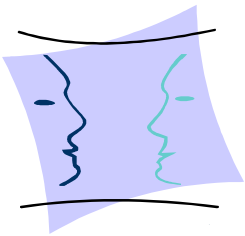
HTTP

Client:

Browser Display:



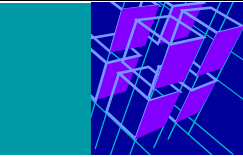
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



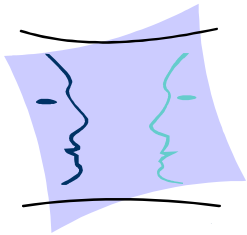
Scope

Contents





http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Browsers

Client uses browser to:

- **Communicate with server**
- **Display documents**

Steps to display a document:

- **Fetch document from server**
- **Interpret contents**
- **Generate local layout**
- **Display layout**

Popular browsers:

- **Netscape Navigator**
- **Microsoft Internet Explorer**
- **IBM Web Explorer**
- **Mosaic**

Client - Server Communication

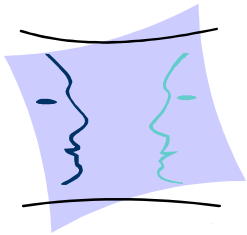
Communication steps:

- **Client opens TCP connection to server**
- **Client issues one request**
- **Server sends one reply**
- **TCP connection is closed**

HTTP - the document transfer protocol

- **„Hypertext Transfer Protocol“**
- **Defines legal requests and replies**
 - request: simple ASCII message
(command plus parameters)
 - reply: document embedded in MIME message
or other information
(MIME = Internet standard for multimedia mail)

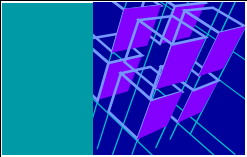
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents





HTTP / 1.0: (RFC 1945)

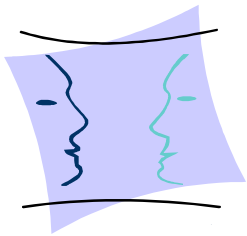
Used by:

- **CERN, NCSA, APACHE**
- **Allows basic hypermedia access to resources available from diverse Applications (including those supported by SMTP, NNTP, FTP, Gopher, WAIS)**

Features:

- **Messages are passed in a format similar to MIME**
- **Simple: Open, Operation, Close**
- **Request: Simple-Request | Full-Request**
- **Response: Simple-Response | Full-Response**

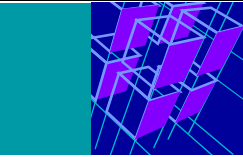
<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents





HTTP (cont.)

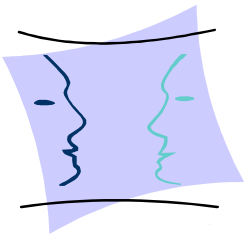
Full Request:

- **Request-Line: Method SP Request-URI SP HTTP-Version CRLF**
- **General-Header (Date, Mime-Version,) and/or**
- **Request-Header (Authorization, From, ..) and/or**
- **Entity-Header (Allow, Content-Type, Expires,...)**
- **CRLF**
- **Entity-Body**

Full Response

- **Status-Line:HTTP-Version SP Status-Code (200,..) SP Reason-Phrase (OK,..)CRLF**
- **General-Header (Date, Mime-Version,) and/or**
- **Response-Header (Location, Server, WWW-Authenticate) and/or**
- **Entity-Header (Allow, Content-Type, Expires,...)**
- **CRLF**
- **Entity-Body**

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



HTTP (cont.)

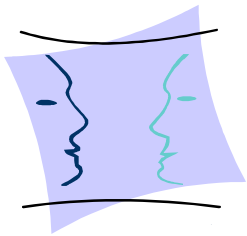
Difficulties & Inefficiencies of HTTP / 1.0

- Fetches single URL per TCP connection
- Closing connection causes loss of congestion information
- Connection opens may congest low-bandwidth links due to lack of flow control on TCP opens and closes
- Servers have large number of connections in close-wait state
- HTTP based on TCP spends more time waiting than it does transferring data

New Features introduced in HTTP / 1.1 (RFC 2086)

- Implemented in JIGSAW, APACHE 1.2b
- Persistent connection
- Cache features
- New requests-methods
- Range request

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



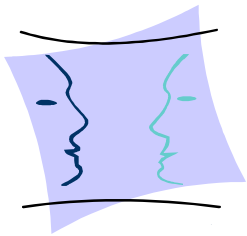
HTTP Methods

HTTP allows an open-ended set of methods that indicate the purpose of a request:

- **GET:**read whatever data identified bei the requested URI
- **HEAD:**read whatever data's header (containing info about data)
- **PUT:**store whatever data under the requested URI
- **POST:**append data under the supplied request URI
- **DELETE:**remove data identified bei the requested URI
- **LINK:**connect two resources
- **UNLINK:**break existing connection
-

RANGE:

- **request one or more subranges of the entity, instead of the entire entity**



Scope

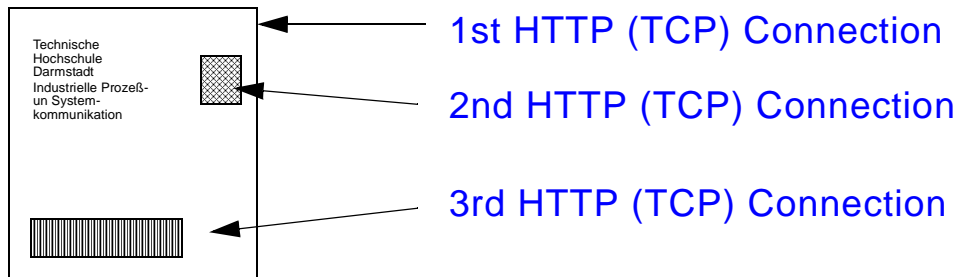
Contents



Persistent Connections

Prior to persistent connection:

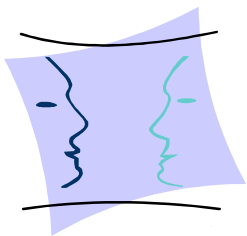
- **Establishing a separate TCP connection to fetch each URL**
 - increases the load on HTTP servers
 - causes congestion on the Internet



Persistent connections:

- **Establishing a unique TCP connection to fetch all URLs on the same server**
 - Open, Operations, close
- **Are the default behavior of any HTTP connection**

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

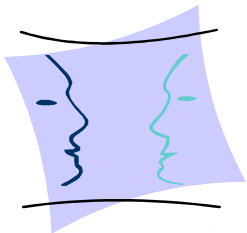
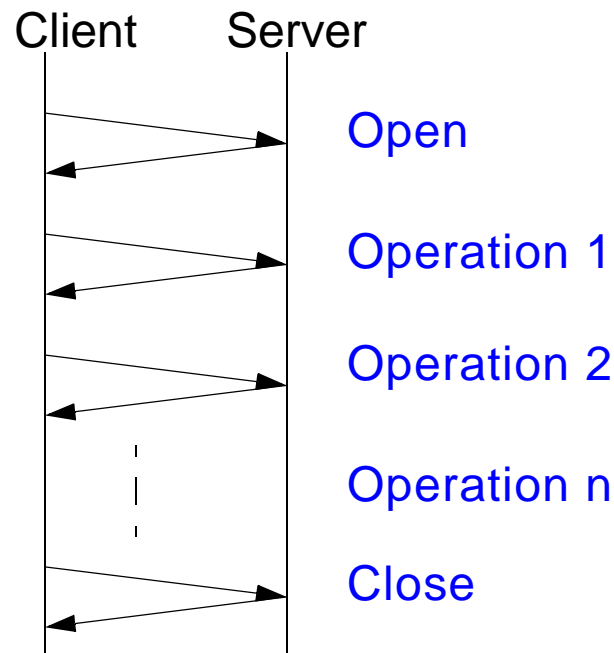
Contents



Persistent Connections

Persistent connections have a number of advantages:

- **Overhead (CPU & memory) for TCP control block management reduced**
- **HTTP requests and responses can be pipelined on a connection:**
 - Pipeline allows a client to make multiple requests without waiting for each response
- **Network congestion is reduced by decreasing the number of packets caused by TCP opens and closes**



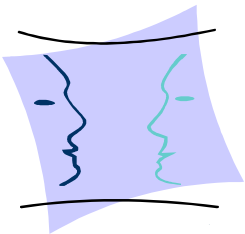
Caching in HTTP

The goal of caching in HTTP / 1.1 is:

- **eliminating the need to send requests in many cases**
 - reducing the number of RTT (Expiration mechanism)
- **eliminating the need to send full responses in many other cases**
 - reducing network bandwidth requirement (Validation mechanism)

Cache-control directives can be broken down into the following categories:

- **Restrictions on what is cachable (Server)**
- **Restrictions on what may be stored by a cache (Server / User-Agent)**
- **Modifications of the basic expiration mechanism (Server / User-Agent)**
- **Controls over cache revalidation and reload (User-Agent)**

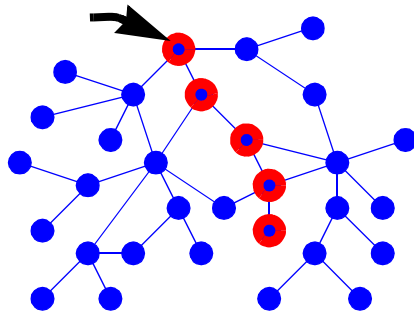


Scope

Contents

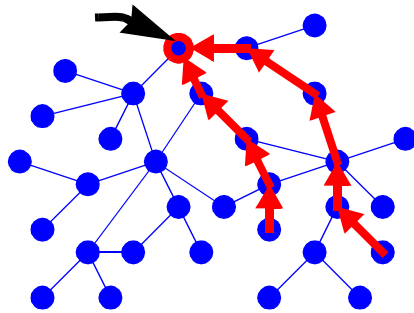


Proxy-Cache-Approaches



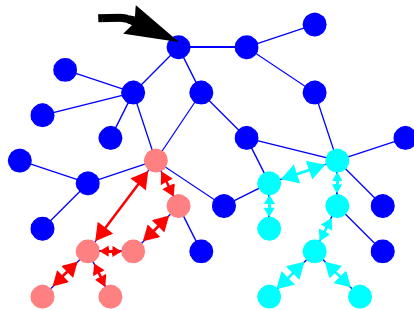
Autonomous Caching

- each Proxy itself decides whether to cache or to displace a document
- no additional management overhead



Hierarchical Caching

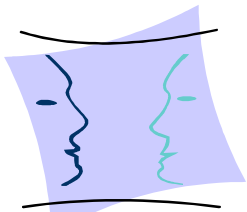
- information about requests is forwarded “upwards”
- central management



Cooperative Caching

- groups of proxies may exchange documents
- even documents that are retrieved very often don't need to be stored everywhere (allows for better scalability)
- group management

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlhäuser



Scope

Contents



Documents: Internal Representation

Representation of pages in HTML:

- „Hypertext Markup Language“
- **Application of SGML standard**
 - definition of markup tags
 - syntax and semantics
- **Browsers**
 - can understand tags
 - can convert them into page layout

Some HTML tags:

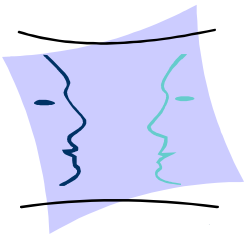
- **<HEAD>...</HEAD>:Page header**
- **...:Bold text**
- **<P>:New paragraph**
- **:Inclusion of picture**
- **...:Link to another document**

New approach:

- **usage of Cascading Style Sheets (CSS) - individual presentation**
- **e.g.: <STYLE TYPE=“text/css”>**
 H1 { color: blue }
 </STYLE>

HTML file:

```
<HTML>
<HEAD>My Page</HEAD>
<BODY>
This is my own Web page.
<P><B>Ain't it nice?</B>
<P>Here's my picture:
<IMG SRC="myself.jpg">
<P>That's all for now!
</BODY>
</HTML>
```



Documents: Inclusion of Links

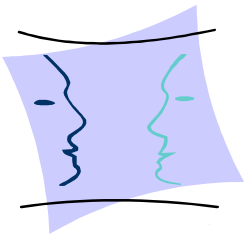
Tag <A> to define links:

- **Format:** ` clickable item `
- **Example:**
 - HTML: „Click ` here` for Fb. NT.“
 - Layout: „Click here for Fb. NT.“
 - User input: Clicking on „here“ loads document „www.fh-koeln.de/fb/fb-nt“

Uniform Resource Locator (URL):

- „Address“ of a page
- **Format:** `protocol_name://name_of_host/local_file_name`
 - Example: see above
- **URL includes three pieces of information:**
 - Protocol for page access
 - http, ftp, gopher, file, mailto, telnet, news
 - Name of host holding the page
 - Name of page local to the host

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Forms and CGI (Common Gateway Interface)

Forms to convey information from client to server:

- **Layout:**

Please send me more information!

Name:

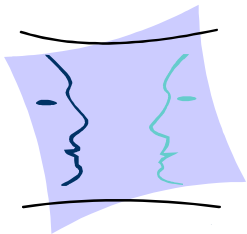
Street:

City:

- **HTML representation:**

```
...  
<FORM ACTION="http://www.info.com/cgi-bin/infoorder"  
  METHOD=POST>  
<P>Please send me more information!  
<P>Name <INPUT NAME="customer" SIZE=30>  
<P>Street <INPUT NAME="street" SIZE=30>  
<P>Name <INPUT NAME="city" SIZE=30>  
<INPUT TYPE=SUBMIT VALUE="Send!">  
</FORM>...
```

http://www.kom.e-technik.tu-darmstadt.de
http://www.tk.informatik.tu-darmstadt.de
© R. Steinmetz, M. Mülhäußer



Scope

Contents

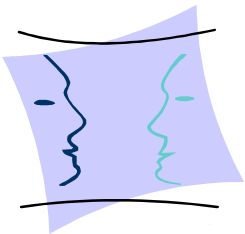


Documents: Forms and CGI (cont.)

- **<FORM> tag specifies**
 - transmission method
 - GET
query appended to URL
 - POST
in this case, message body e.g.:
customer=Bill+Clinton&street=The+White+House&city=Washington
 - action to be taken by server
 - e.g. execution of a *CGI script*

CGI: Common Gateway Interface

- **CGI script: program**
 - executed at the server
 - on behalf of clients' request (as defined by transmitted parameters)
 - e.g. database query
- **Output of CGI script returned to client**
 - e.g. newly generated WWW page
- **Addressing of scripts via URLs**
 - usually in directory cgi-bin



Keeping State

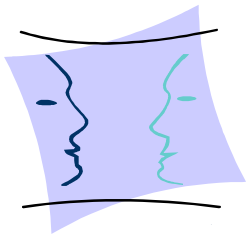
hidden values in forms

- **non-persistent, lost if “document track” is left**

Cookies

- **information persistently stored on client**
- **managed by means of JavaScript**
- **security implications**
 - unintended retrieval (limited by domain concept)
 - access to locally stored values

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlpfänger



Scope

Contents



Enhancing Browser Functionality - JavaScript

Features:

- **plattform independent**
- **event-driven**
- **interpreted**
- **has to consider security constraints**
- **enables interaction with:**
 - Browser
 - Java Applets

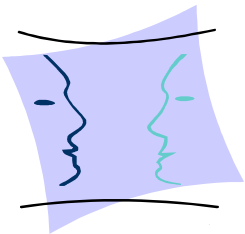
Structure:

- **<SCRIPT>**

Objects
Functions
Statements
Events

</SCRIPT>

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mülhäußer



Scope

Contents



Enhanced Client-Server Interaction

Problem with CGI scripts:

- **Client interactions are limited:** programs are executed at the server.

Solution:

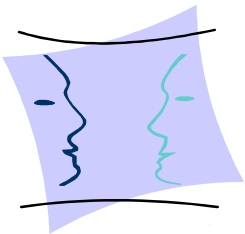
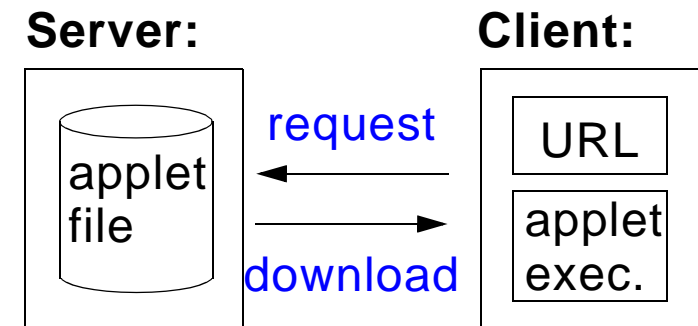
- **Execute scripts / programs at the client**

Main idea:

- **URL on page points to small program**
 - program is called *applet*
- **When clicked:**
Browser downloads applet to the client and executes it there

Advantages:

- **WWW pages become more interactive**
- **Browser capabilities can be extended on-the-fly**
 - e.g. loading of new viewers when needed



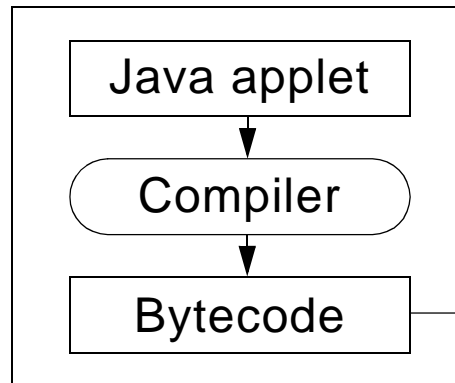
Java: Fundamentals

Java:

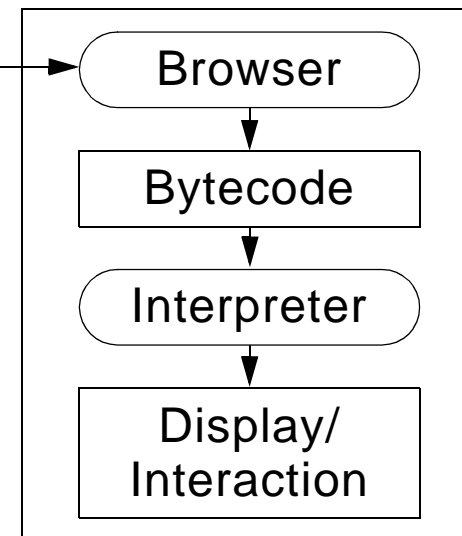
- Programming language to write applets
- Originated from Sun Microsystems

Steps to use a Java applet:

Server:



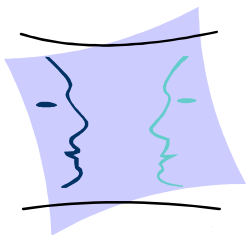
Client:



download

Inclusion of applets into HTML documents

- By new tag `<APPLET>`
- E.g. `<APPLET CODE=game.class WIDTH=100 HEIGHT=200></APPLET>`



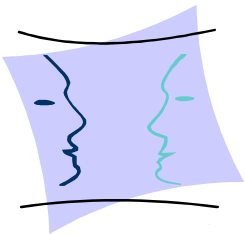
Applet lifecycle

```
import java.applet.*;

public class myApplet extends Applet {

    public void init()
    {
        ... // called on first startup
    }
    public void start()
    {
        ... // called when page entered (again)
    }
    public void stop()
    {
        ... // called when page left
    }
    public void destroy()
    {
        ... // called when browser / runtime finally left
    }
    public void myMethod(){
        ... // do something
    }
}
```

<http://www.kom.e-technik.tu-darmstadt.de>
<http://www.tk.informatik.tu-darmstadt.de>
© R. Steinmetz, M. Mühlerhäuser

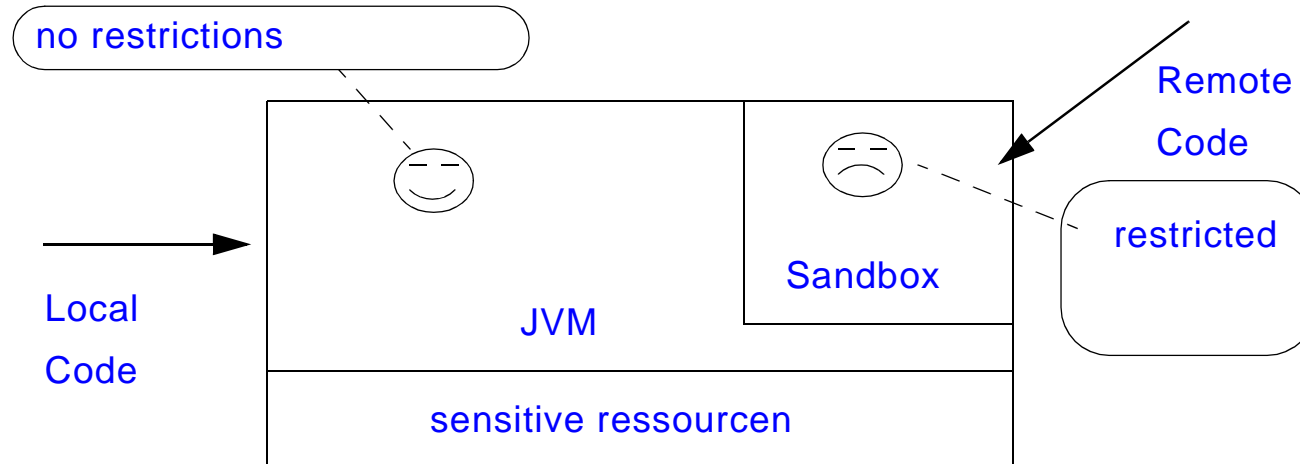


Scope

Contents



Java Applet Security Model



prevents downloaded (untrusted) code from malicious operations

- **no interaction with local file system**
- **network connections just with originating source**
- **special outlook for “untrusted applet windows”**

may be adapted

- **signed and trusted applets**
- **use certificates (public key cryptography) to ensure**
 - integrity
 - proof of origin

