Verteilte Web-basierte Systeme – SS 2006

Verteilte Web-basierte Systeme

Dr.-Ing. Martin Gaedke
Universität Karlsruhe (TH)

Verteilte Web-basierte Systeme – SS 2006

Part II

Technology: Basics & Principles

Part 2 – Overview

1. Standards and Organizations
2. Network Basics
3. Web System Basics
4. Uniform Addressing
5. Content-Type
6. Markup
7. Hypertext Markup Language (HTML)
8. eXtensible Markup Language (XML)
9. Further Readings

- Web Engineering …
  … requires us to become aware of Technology and the distributed nature of the Environment

Verteilte Web-basierte Systeme – SS 2006

Chapter://1

Standards and Organizations

Early Standards

"When you measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of science."
– Lord Kelvin (William Thomson)

Why Standards?

- Understanding "Standards" is a MUST
  - Helps in drawing decisions
  - Helps building reusable and interoperable solutions
  - Helps communicating with others
- Requires
  - Understanding of processes
  - Knowing organizations and standard bodies
  - Knowledge of reading specifications (Standards)

World Wide Web Consortium

- World Wide Web Consortium (W3C)
  - http://www.w3.org/
  - Founded in 1994 by Tim Berners-Lee @ MIT
  - 450 member organizations (as of Jan 2003)
  - Output includes 20+ technical specifications
- W3C Mission
  - "The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding."

W3C´s Long Term Goals

- Universal Access
  - "Make the Web accessible to all by promoting technologies that take into account the vast differences in culture, education, ability, material resources, and physical limitations of users on all continents"
- Semantic Web
  - "To develop a software environment that permits each user to make the best use of the resources available on the Web"
- Web of Trust
  - "To guide the Web's development with careful consideration for the novel legal, commercial, and social issues raised by this technology"

W3C Process

- Workshops to measure Interest
- Notes from Members (e.g. SOAP, WSDL)
- Briefing Package with Membership Vote
- Creation of an Activity/Working Group
- Working Group Process
  - Requirements Document
  - Public Working Drafts every three Months
  - Last Call Working Draft
  - Candidate Recommendation (CR)
  - Proposed Recommendation (PR)
  - Recommendation (REC)
- Technical Architecture Group being formed

W3C Output Examples

- Recommendations
  - HTML: http://w3.org/TR/html401/
  - XML: http://w3.org/TR/REC-xml
  - Namespaces: http://w3.org/TR/REC-xml-names
  - XML Schema: http://w3.org/TR/xmlschema-1/
- Works in progress:
  - XML Query: http://w3.org/TR/xmlquery-req
  - XML Protocol: http://www.w3.org/TR/soap12/

Example: SOAP History

- Protocol workshops: XML 1999, WWW9, etc.
- SOAP 1.1 submitted as W3C Note (May 2000)
- Briefing package and vote (August 2000)
- Creation of XML Protocol WG (Sept 2000)
- XML Protocol WG Charter
- SOAP 1.2 Working Draft (July 2001)
- Last Call Working Draft (Nov/Dec 2001)
- No plan for Candidate Recommendation
- Recommendation (Estimated Time of Arrival May 2002)

Organizations for Standards

- Internet Engineering Task Force
  - http://www.ietf.org/
  - TCP/IP, HTTP, etc.
- Organization for the Advancement of Structured Information Standards (OASIS)
  - http://www.oasis-open.org/
  - ebXML, WSRP, WSIA, Rights Language, UDDI, etc.

Verteilte Web-basierte Systeme – SS 2006

Chapter://2

Network Basics

Attention!!!

- **This Section is just a short and very incomplete Overview of Network Technology and Concepts**
- It focuses only on some concepts that are very important for Web Engineering
- If you do not know about OSI etc., cf. references at the end of this chapter

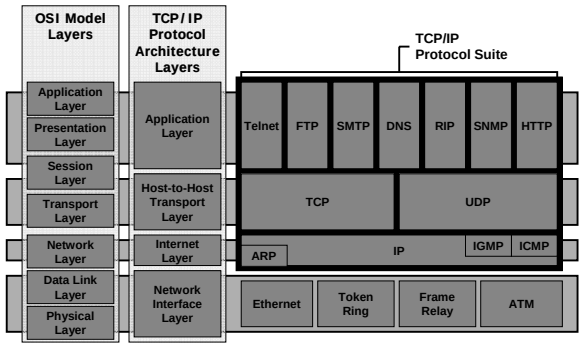
Introduction

- Interconnected collections of independent computers
- Why have networks?
  - Resource sharing
  - Reliability
  - Cost savings
  - Communication
- Web technologies add:
  - **New business models**: e-commerce, advertising, etc. (or even refine the existing business)
  - Entertainment
  - Applications “without” a client installation

Network Scope

- Internet: a collection of connected networks
- Internet: a specific world-wide network based on TCP/IP, used to connect companies, universities, governments, organizations and individuals. Originated as ARPANET, funded by the US DoD.
- Intranet: a network based on Internet technologies that is internal to a company or organization
- Extranet: a network based on Internet technologies that connects one company or organization to another

Protocol Layers Overview



Internet Layer

- Internet Protocol (IP)
  - Responsible for getting packets from source to destination across multiple hops
  - Not reliable
- **IP Address (version 4)**: 32 bit value usually written in dotted decimal notation as four 8-bit numbers (0 to 255)
  - E.g. 127.0.0.1

Transport Layer

- Provides efficient, reliable and cost-effective service
- Ports identify application
  - Well-known ports identify standard services, RFC 1700
  - E.g. HTTP port 80, SMTP port 25
- Transmission Control Protocol (TCP)
  - Provides reliable, connection-oriented byte stream
- User Datagram Protocol (UDP)
  - Connectionless, unreliable

Transmission Control Protocol

- TCP - Motivation (September 1981), RFC793
  - "Computer communication systems are playing an increasingly important role in military, government, and civilian environments. This document focuses its attention primarily on military computer communication requirements, especially robustness in the presence of communication unreliability and availability in the presence of congestion, ...
  - As strategic and tactical computer communication networks are developed and deployed, it is essential to provide means of interconnecting them and to provide standard interprocess communication protocols which can support a broad range of applications. ....
  - TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols."

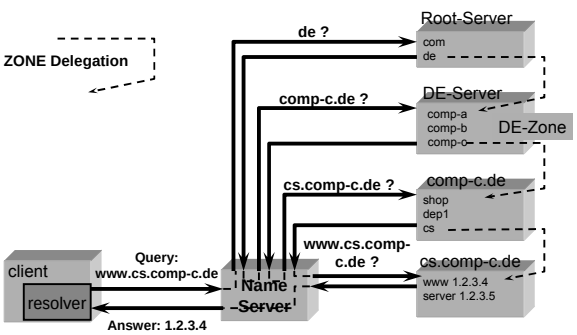
Application Layer

- Telnet: Remote sessions
- File Transfer Protocol (FTP)
- Network News Transfer Protocol (NNTP)
- Simple Network Management Protocol (SNMP)
- Simple Mail Transfer Protocol (SMTP)
- Post Office Protocol (POP3)
- Interactive Mail Access Protocol (IMAP)
- Many more…

Domain Name System (DNS)

- **DNS** – Idea: User-friendly domain names
  - RFCs 1034 and 1035
  - Map Names to IP Addresses
  - E.g. webe.tm.uni-karlsruhe.de
- **Zone** - Part of the Domain Name Space provided by a Name-Server (DNS-Server)
  - DNS Database often called Zone File

DNS host name resolution



Example: Part of a Zone File

- Example:

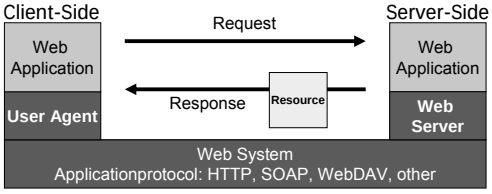
```
; SOA … Name Server …
; Host addresses
localhost. IN A 127.0.0.1
www.mysite.edu. IN A 1.2.3.4

; Multi-homed hosts (Remember ROUND ROBIN!)
hightraffic.mysite.edu. IN A 1.2.3.6
hightraffic.mysite.edu. IN A 1.2.3.7
hightraffic.mysite.edu. IN A 1.2.3.8
; Aliases (Remember HTTP-Server)
news.mysite.edu. IN CNAME www.mysite.edu
```

Verteilte Web-basierte Systeme – SS 2006

Chapter://3

Web System Basics

Technologies Involved…

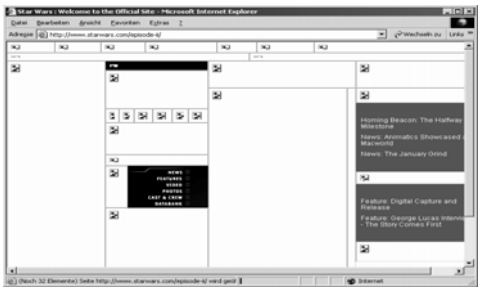


- Browser, User Agent
  - Content
  - Interaction
  - Presentation
  - Navigation
  - Process
- Web System / Network
  - Communication
  - Protocols, like HTTP, WebDAV, SOAP, FTP, SMTP
- Web Server
  - Content
  - Interaction
  - Presentation
  - Navigation
  - Process

Return Of The Resource (1)

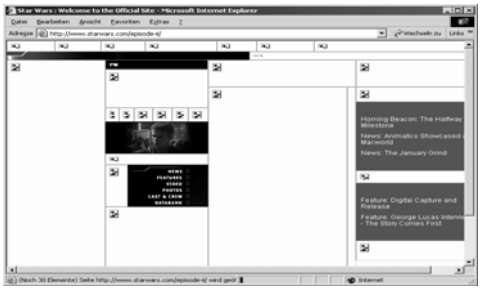
Screenshot from the site www.starwars.com

Return Of The Resource (2)



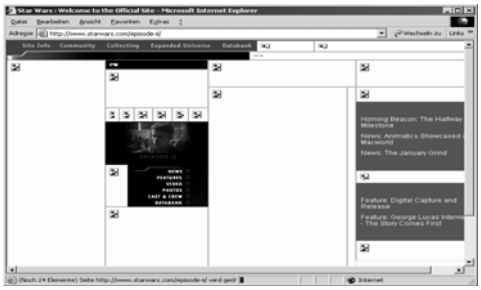
Screenshot from the site www.starwars.com

Return Of The Resource (3)



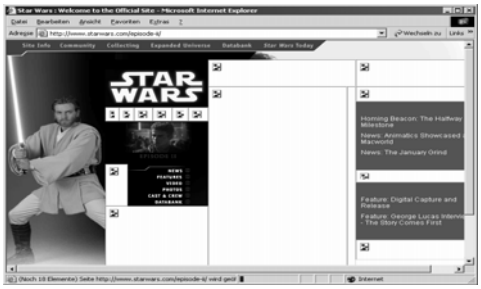
Screenshot from the site www.starwars.com

Return Of The Resource (4)



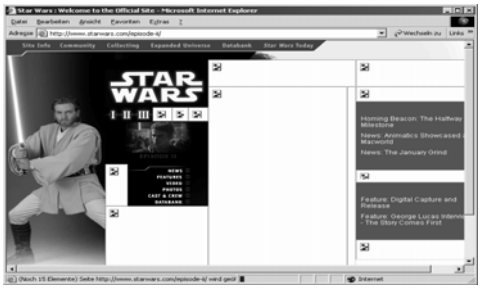
Screenshot from the site www.starwars.com

Return Of The Resource (5)



Screenshot from the site www.starwars.com

Return Of The Resource (6)



Screenshot from the site www.starwars.com

Return Of The Resource (7)

Screenshot from the site www.starwars.com

Return Of The Resource (8)

Screenshot from the site www.starwars.com

Retrieving Information

Client-Side:

- (1) Prepare Request
- (2) Request Resource
- (3) Handle Response
- (4) Process/Render Data of Resource

Server-Side:

- (A) Handle Request
- (B) Process
- (C) Send Resource

1 - Prepare Request

- Address the resource (URI)
  - Example: **http://localhost/**
- Find URI-Resolver for scheme in use
  - Example: URI-Resolver for **http**
- URI Resolver
  - Get address of resource (scheme-specific)
  - Example **URL-Resolver**:
    - Host: localhost
    - Resource: /

2 - Request Resource

- Send request to address (**Communication**)
  - Depends on scheme of URI, e.g. mailto, http
  - Transmission Protocol defined by scheme
  - Remember: allows for caching!
  - Example:

    Use TCP, connect with localhost at port 80 and send:

    GET / HTTP/1.0

    [CRLF]

Request Resource – Demo

- URI: http://localhost/
- URL-Resolver:
  - Scheme=http
  - Transmission-Protocol: TCP/IP
- Scheme-specific-part = localhost
  - Check scheme-specific-part for Host, retrieve IP using DNS
  - IP Address for localhost: 127.0.0.1
  - Port: 80 (if not specified by scheme-specific-part)
- Send Protocol Code

3 - Handle Response

- Handle protocol, e.g. HTTP 302 Object moved, cookies
- Retrieve resource
- Example:
  - Further action needed
  - Response includes resource

4 - Process Data of Resource

- Processing depends on user agent
- Browser
  - Process header
  - Check content-type
  - Process resource data depending on MIME-Type
  - E.g. render: text/html, text/text, image/gif
- Other User Agents
  - Process header
  - Process data
  - E.g. WebService-Client: Process XML-Resource, wget/robots retrieving collections of Web-Pages

Lessons Learnt…

- Addressing and retrieving resource
- Content-Type
- Presentation

Verteilte Web-basierte Systeme – SS 2006

Chapter://4

Uniform Addressing

Addressing Resources

- As with DNS and IP-Adresses…
- Goals:
  - It must be possible to identify resources
    - By Name
    - By Address resp. Location
  - Any resource in the Internet should be identified
    - Web pages, FTP-Resources, Mailboxes, Directories, interactive services
- Requirements: Identification mechanism should be
  - Extensible
  - Complete
  - Printable (to be represented as string of 7-bit characters)

Uniform Resource Identifier

- Uniform Resource Identifier (URI)
  - Generic term for all textual names/addresses
  - URI is URL or URN or URC
- Uniform Resource Locator (URL)
  - The set of URI schemes that have *explicit instructions on how to access* the resource over the Internet
- Uniform Resource Name (URN)
  - A URI that has an institutional commitment to availability, etc.
  - A particular scheme intended to identify resources
- Uniform Resource Characteristic (URC)
  - A URC provides Meta Information

Uniform Resource Identifier

- **URI – Syntax for identifiers [RFC1630]**
  <uri> ::= <scheme>":"<scheme-specific-part>
- **<scheme>**
  name of the scheme
- **<scheme-specific-part>**
  identifier in a format that is according to the scheme

Reserved Characters

- For all types of URIs the following Rules apply:
- The percent sign ("%", ASCII 25 hex)
  - Escape character
- Hierarchical forms ("/", ASCII 2F hex)
  - Delimiting of substrings whose relationship is hierarchical
- Hash - fragment delimiter ("#", ASCII 23 hex)
  - Identifies a fragment in a resource
- Query Delimiter ("?", ASCII 3F hex)
  - To delimit the boundary between the URI of a queryable object

Uniform Resource Name

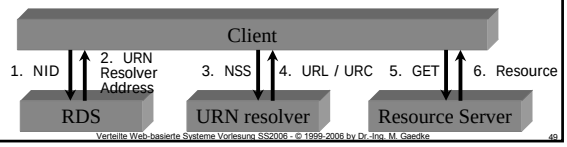
- URN – Scheme definition [RFC 1737, RFC 2141]
  - URNs serve as persistent, location-independent, resource identifiers
- <scheme> ::= "urn"
- <scheme-specific-part> ::= <nid> ":" <nss>
  - nid = Namespace Identifier
  - nss = Namespace Specific String
- E.g. urn:schemas:httpmail:subject

URN Properties

- Global scope and uniqueness
- Persistence
- Scalable
- Legacy support
- Extensible
- Independent
- Resolvable

URN - Resolution

- Infrastructure for URNs is still experimental
  - Resolver Discovery Service (RDS)
  - Name service, name resolution (URN resolver)
  - Result of the resolution is a URL or a URC
  - Cf. RFC 1737, 2276
  - <urn> ::= "urn:" <nid> ":" <nss>



Uniform Resource Locator

- URL – Scheme definition [RFC1738]
  - *explicit instructions on how to access …*
- <scheme> ::= “http” | “https” | “ftp” | “news” | “mailto” | “nntp” …
- Specific Part defined in a general format
- <scheme-specific-part> ::=
  ["//"] [user [":"password] "@"] host [":"port] ["/"url-path]
- Definitions are maintained by the Internet Assigned Numbers Authority (IANA)
- URLs can also be relative [RFC 1808]

Example – HTTP URL

- HTTP URL

<scheme> ::= “http”

<scheme-specific-part> ::=
"//"[user[":"password]"@"]<host>[":"<port>][<abs_path>]

<host> = <A legal Internet host domain name or IP address (in dotted-decimal form), as defined by Section 2.1 of RFC 1123>

<port> = *DIGIT

<abs_path> = "/"[<path>][";"<params>]["?"<query>]["#"<fragment>]

<path> = <fsegment> *("/")

- Example
  - http://webe.tm.uni-karlsruhe:8080/a/b?x=1#2345
  - http://userid:pwd@www.secret.xyz/account/euro?add=100#FragId

Comparison URN vs. URL

	URN	URL
Scope	Global	Global (abs. URL) Local (rel. URL)
Globally Unique	Yes	Yes (abs. URL) No (rel. URL)
Persistent	Yes	No
Scalable	Yes	Yes
Legacy Support	Yes	Limited
Resolution	Not yet determined	Partly using DNS

Verteilte Web-basierte Systeme – SS 2006

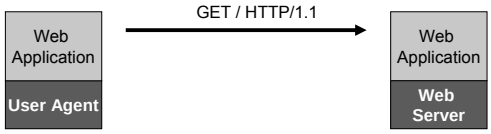
Chapter://5

Content-Type

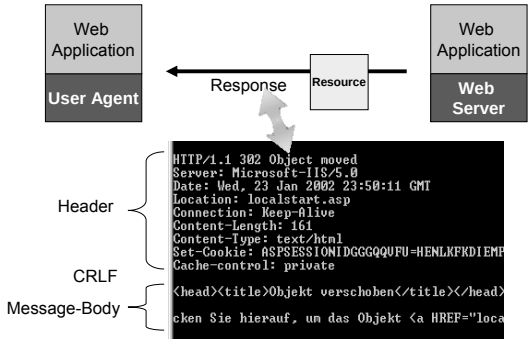
Introduction

- **Content** – Data provided in a specific format.
- **Content-Type** – Defines the format or type used for encoding data
  - Represent Media Types, HTML-documents, images, audio, video
- For further information regarding
  - Audio
  - Video
  - Cf. References at the End of this Chapter

Resource Example Again - 1



Resource Example Again - 2



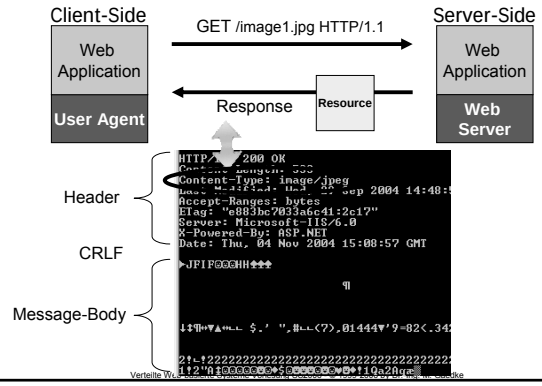
Resource Example Again - 3

- (3) Handle Response

Resource Example Again - 3

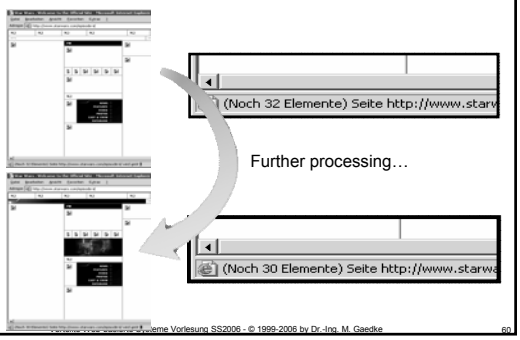
- (4) Process/Render Data of Resource

Resource Example Again - 4



Resource Example Again - 5

- (4) Process/Render Data of Resource

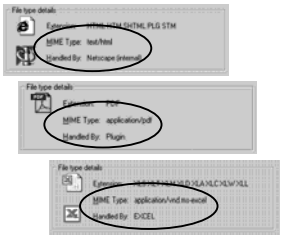


MIME

- MIME - Multipurpose Internet Mail Extension: RFC-Series 2045, 2046, 2047, 2048, 2049 (1996)
- Constraint
  - Backward compatibility with ASCII mail
- Concept
  - MIME-Messages can consist of many parts
    - Multi-part messages
  - Message parts can have different types of content
    - Content-Type
  - Each part of a message has its own headers to describe the content
    - Type, ID, Coding

Concept

- Open concept to integrate arbitrary media
  - Transmitted in the MIME format
- Interpretation of different media types in the WWW
  - Browser build-in for most basic types, e.g. text, HTML, GIF
  - Using browser Plug-Ins, e.g. for Acrobat PDF, Shockwave, Flash
  - Using external applications (helper applications)
    - E.g. Ghostscript for PostScript, other proprietary formats/applications
  - Save files
    - Download of arbitrary formats



MIME Content-Types

- Basic Types
  - text, audio, video, image
  - multipart (more than one body part)
  - message
  - model (multi dimensional Objects, e.g. für Virtual Reality)
  - application (application specific Content-Types)
  - x- for self defined types
- Subtypes
  - Basis Type and Subtype define together the Media Type
  - E.g. subtype of image: gif, jpeg
  - x- for self defined sub types
- Parameter
  - Dependent on the data type, e.g. name, boundary, ...

Content-Type – Examples

application/hylafax	fax	audio/basic	au snd
application/msword	doc	audio/x-aiff	aif aiff aifc
application/mspowerpoint	ppt	audio/x-wav	wav
application/pdf	pdf	image/gif	gif
application/postscript	eps ps	image/jpeg	jpeg jpg jpe
		multipart/mixed	
application/x-dvi	dvi	text/html	html htm
application/x-latex	latex	text/plain	txt
application/msdownload	exe bat com	video/mpeg	mpeg mpg
		video/quicktime	qt mov
		video/x-msvideo	avi
application/x-sh	sh	x-world/x-vrml	wrl
application/zip	zip	...	

Content-Type Syntax

- content := "Content-Type" ":" type "/" subtype *(";" parameter) ;
- type := discrete-type / composite-type
- discrete-type := "text" / "image" / "audio" / "video" / "application" / extension-token
- composite-type := "message" / "multipart" / extension-token
- Matching of Media Type and Subtype is ALWAYS **Case-Insensitive**.
- Multipart Media Type
  - Mixed Subtype
  - Alternative Subtype
  - Parallel Subtype

MIME Example

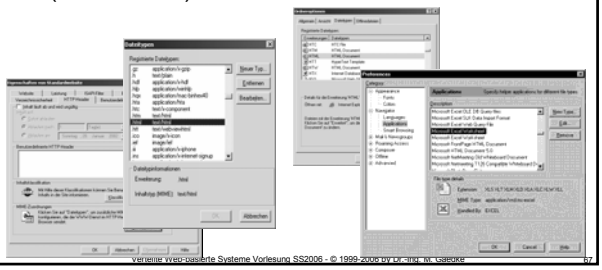
```
Part of the Email Header: MIME
Here: Text and Attachment
Content-Type: multipart/mixed; boundary="===-1203946231== ==========="

===-1203946231== ==========
Content-Type: text/plain; charset="us-ascii" ; format="flowed"

===-1203946231== ==========
Content-Id: <p05100306b83d3caaff11@[139.82.20.12].0.0>
Content-Type: application/vnd.ms-excel; name="WWW2002-Review-Paper-Assign.xls"
 ; x-mac-type="584C5334"
 ; x-mac-creator="5843454C"
Content-Disposition: attachment; filename="WWW2002-Review-Paper-Assign.xls"
Content-Transfer-Encoding: base64

===-1203946231== ==========
...
```

Internet headers:

```
To: Martin Gaedke <gaedke@teco.uni-karlsruhe.de>
From: schwabe@inf.puc-rio.br (Daniel Schwabe)
Subject: RE: FW: WWW2002 your track submissions
Content-Type: multipart/mixed; boundary="----------
==_==========_==========_=========="

---------- -1203946231== ==========
```

MIME Extension

- Mapping of file types (e.g. by file extension) to MIME types (**server-side**)
- Mapping of MIME types to applications (**browser-side**)



Verteilte Web-basierte Systeme – SS 2006

Chapter://6

Markup

Markup

- Computerized typesetting
  - Typesetting macros embedded in ASCII
    - Layout directives, e.g. TEX, RTF
  - Presentational markup
    - Commands to define the layout
- Examples
  - *Hello* → **Hello**
  - /Hello/ → *Hello*

Markup

- Semantic Markup
  - Authors put annotations into their texts to help the publisher to understand what type of text this is (e.g. “this is a heading”)
  - Annotations are agreed between author and publisher
- Publisher decides on the layout
  - Descriptive markup
    - Describing content not the layout
  - Markup to support search in documents
    - Words in headings are more important than in footnotes
    - Markup for machines vs. markup for humans

Markup – ISO-Definitions

- **Markup** – Text that is added to the data of a document in order to convey information about it
- **Descriptive Markup** – Markup that describes the structure and other attributes of a document in a non-system-specific way, independently of any processing that may be performed on it
- **Processing Instruction (PI)** – Markup consisting of system-specific data that controls how a document is to be processed

Markup Language Features

- Stylistic (appearance)
  - <I><B><U>
- Structural (layout)
  - <P>
<H2>
- Semantic (meaning)
  - <TITLE>
  - <META NAME=keywords CONTENT = " …... " >
- Functional (action)
  - <BLINK>
  - <A HREF = "[link]">Click here</A>

Example – I

- Example:

Important Announcement:

Register for Web Engineering course next week.

Example – II

- Example:

<announcement>

<type>Important</type>

<title uri="http://www.webe.aaa">Register for <bold>Web Engineering</bold> course next week.

</title>

</announcement>

Verteilte Web-basierte Systeme – SS 2006

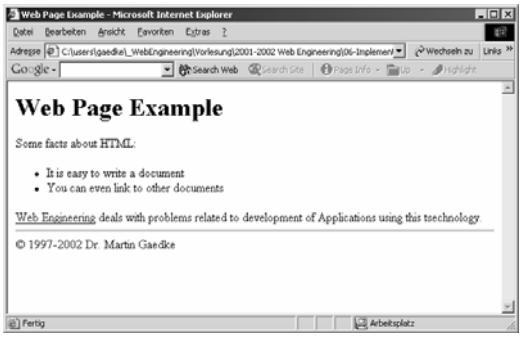
Chapter://7

Hypertext Markup Language (HTML)

Hypertext Markup Language

- HTML – The Markup Language used to represent Web pages for viewing by people
  - Rendered and viewed in a Web Browser
  - Not extendible
- Documents
  - Easy to write – Markup your data with tags
  - Platform independent
  - Can contain links to Images, documents, and other pages
  - HTML is an application of SGML **(Standard Generalized Markup Language, ISO 8879:1986 – used for defining Markup Languages)**
- For further information: http://www.w3.org/MarkUp/

Web Page Example



Web Page Example

HTML Header

HTML Body

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
    <head>
        <title>Web Page Example</title>
    </head>

    <body>
        <h1>Web Page Example</h1>
        Some facts about HTML:
        <ul>
          <li>It is easy to write a document
          <li>You can even link to other documents
        </ul>
        <a href="http://webengineering.org">Web Engineering</a> deals with problems related
        to development of Applications using this technology.
        <HR>
        &copy; 1997-2002 Dr. Martin Gaedke
    </body>
</html>
```

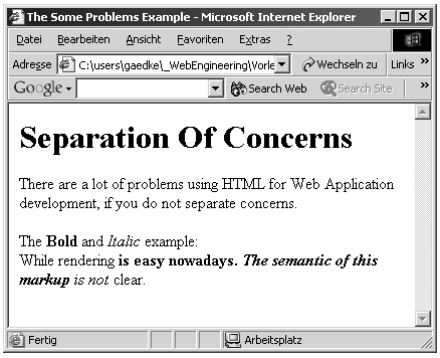
HTML – "Extension" History

- 1989 The WWW project proposal, Tim Berners-Lee
- 1990 First implementation at CERN
- 1993 HTML+ proposal to *extend* HTML
  - In summary form presented at WWW´1 Conference 1994
- 1995 HTML 2.0, RFC 1866
- 1995 HTML 3.0, Internet Draft
  - Extended version of HTML+
  - Combination of the features in Mosaic, Arena and few others
  - Enabling development of Web applications: Forms
  - Never standardized, but helped to stimulate community: Netscape founded

HTML – "Extension" History II

- 1997 HTML 3.2
  - W3C Recomendation
  - Specified: Tables, Applets, Script, etc.
- 1998 HTML 4.0
  - Separation of content and presentation with Cascading Style Sheets (CSS)
  - Three Document Type Definitions
    - Transitional DTD – Defines backwards compatibility for older documents
    - Strict DTD – Defines HTML 4.0 elements only and should be used for new documents
    - Frameset DTD – Defines framesets as seperate type of document
- 1999 HTML 4.01
  - Fixed a number of bugs in 1998

Some Problems…



Some Problems…

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html><head>
    <title>The Some Problems Example</title>
</head><body>
    <H1>Separation Of Concerns</h1>
    There are a lot of problems using
    HTML for <WebEngineering>Web Application development</WebEngineering>,
    if you do not separate concerns. <P>

    The <b>Bold</b> and <i>Italic</i> example: <br>
    While rendering <b> is easy nowadays.<i> The semantic of this markup </b> is
    not </i> clear.

</BODY></HTML>
```

- REMEMBER: Do not develop Applications in this manner!

Some Observations

- Powerful for Presentation (Focus on Client-Side)
  - Cascading Style Sheets
  - Allows for dynamic behavior using e.g. Script Code or DHTML
  - Allows for proprietary extension (ActiveX, PlugIns, etc.).
- Easy to write and generate, **but**:
  - Difficult to parse
  - No support for extending semantics, e.g. using your own tags
- Difficult to apply disciplined approaches

Verteilte Web-basierte Systeme – SS 2006

Chapter://8

eXtensible Markup Language (XML)

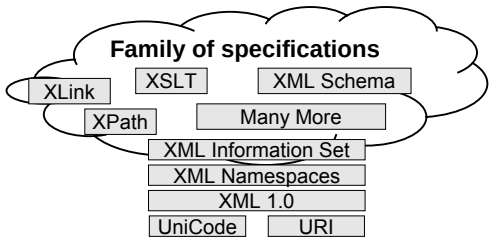
Introduction

- eXtensible Markup Language (XML) – universal format for structured documents and data on the Web
- W3C recommendation to complement HTML
- http://www.w3.org/XML/
- XML looks like HTML **but** in this context the interpretation of data is the job of the application

XML History

- 1996 Development started
- 1997 Public Drafts
  - E.g. Provided in paper form at WWW6, Santa Clara, CA, USA
  - March first XML conference
- 1998 W3C REC
  - Based on experience: simplified form of SGML
  - XML derived from SGML – both are used for defining Markup Languages
  - XML = 80% of SGML´s capabilities, 20% of SGML´s complexity

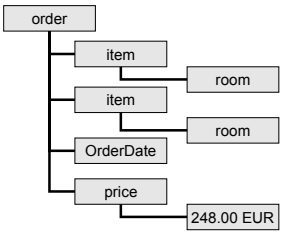
A Family of Specifications



- …and still growing!

XML Example

- A way of representing information
- XML documents (application of XML) are composed of **elements** and *attributes*




```
<?xml version="1.0"?>
<order OrderID="10643">
    <item>
        <room id="Room10"/>
    </item>
    <item>
        <room id="Room11"/>
    </item>
    <OrderDate
        ts="2004-05-17T00:00:00"/>
    <price>248.00 EUR</price>
</order>
```


XML Facts

- Important for Web Engineering because it removes two constraints, which were holding back Web developments:
  - Dependence on a single, inflexible Document Type (HTML);
  - The complexity of full SGML, whose syntax allows many powerful but hard-to-program options

XML and Web Engineering

- **Well-Formed** – An XML document is well-formed if it complies to the following rules:
  - Elements have an open and close **tag:** <tag>content</tag>
  - Empty elements are closed by " ***/*** " e.g. <emptyelem/>
  - Attribute values are quoted
- **Valid** – An XML document is well-formed and if its content conforms to the rules in its document type definition or schema
  - Validity allows an application to make sure the XML data is complete, is formated properly, and has appropriate attribute values.

Footnote

- XML is text, but isn't meant to be read
  - Applications can store their data or respond in Web-compliant style (Text) instead of binary format
- XML is verbose by design
  - Data + Markup is in most cases larger than a binary format – but disk space is cheap, HTTP supports compression on the fly (gzip)
- As XML defines Markup Languages…
  - XHTML, XSL, XForms, etc. are applications of XML

Verteilte Web-basierte Systeme – SS 2006

Chapter://9

Further Readings

Literature

- T. Berners-Lee, Information Management: A Proposal, CERN, Web Page 1990. http://www.w3.org/Proposal.html
- T. Bray, J. Paoli und C. M. Sperberg-McQueen, Extensible Markup Language (XML) 1.0, World Wide Web Consortium (W3C), Recommendation REC-xml-1998021, 02.10.1998 1998. http://www.w3.org/TR/1999/REC-xml-1998021
- J. December und N. Randall, The World Wide Web unleashed, 1st ed. Indianapolis, IN: Sams Pub., 1994.

Literature

- M. Gaedke, M. Nussbaumer, O. Jung, M. Diekmann, Kapitel 6: Implementierungstechnologien für Web-Anwendungen, in Web Engineering – Systematische Entwicklung von Web-Anwendungen, G. Kappel, B. Pröll, S. Reich, W. Retschitzegger (Hrsg.), dpunkt.verlag
- Chapter 3, 7: T. A. Powell, Web Site Engineering, Prentice Hall PTR
- R. Tolksdorf, XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web, in Informatik Spektrum, vol. 22(6), S. 407-421, 1999.
- V. Turau, Techniken zur Realisierung Web-basierter Anwendungen, in Informatik Spektrum, vol. 22(1), S. 3-12, 1999.

==========

Further information available at Lecture Web Site

==========

Web References

- URLs
  - **http://de.selfhtml.org/**
  - **http://w3.org**