MCS

Web Technologies

MCS

Outline

Introduction

Conventional software-system development is the process of designing a correct HOW for a well-defined WHAT. Patridge (1992)

- By seperating the content from presentation, using an appropriate technology becomes easy.
- Additionaly, we have to consider requirements for the distribution and integration of other systems according to a selected or existing architecture.
- Web technologies stem from the three views of Web Standards
  - Client:: request
  - Server:: response
  - Protocol:: rules for the communication between client and server

Fundamentals

- The very roots of the Web are in the markup and hypertext
- SGML (Standard Generalized Markup Language) is the basis of HTML and XML
- **Markup**
- Concept of markup originates from the publishing industry
- These are the typographic instructions for document formatting
  - *Hello* can be defined as **Hello**

Fundamentals-II

- ISO Defines the following markup classes
  - *Markup:* This is text inserted in a document to add information as to how characters and contents should be represented in the document.
  - *Descriptive markup:* This is markup that describes the structure and other attributes of a document, regardless of how this document is processed for representation (e.g., comments).
  - *Processing instructions:* This is markup consisting of system-specific data; it controls the way a document is processed.
- **SGML** standards promoted by US publishing industry
- Primarily uses tags (<tags>)

Fundamentals-III

Hypertext and Hypermedia

- Hypertext is understood as the organisation of the interconnection of single information units.
- Links are used to express the relationships between these units
- It is the core concept of World Wide Web (WWW)
- ***Hypermedia*** is commonly seen as a way to extend the hypertext principle to arbitrary multimedia objects, e.g., images or video.

Client/Server Communication

- Basic communication mechanism through which a user (client) interacts with the Web application (server)
- Primarily a 2-layer architecture
- However, the server may involve addition steps of processing to integration various applications or data/information sources
- Discuss few of the protocols/technologies here
  - SMTP – Simple Mail Transfer Protocol
  - RTSP – Real Time Streaming Protocol
  - HTTP – Hypertext Transfer Protocol
  - Session Tracking

Protocols

SMTP – Simple Mail Transfer Protocol

- SMTP combined with POP3 (Post Office Protocol) or IMAP (Internet Message Access Protocol) is used to send and receive emails
- SMTP is increasingly used as a transport protocol for asynchronous message exchange based on SOAP

RTSP – Real Time Streaming Protocol

- It is standard designed to support the delivery of multimedia data in real-time conditions
- RTSP allows the transmission of resources to the client in a timely context rather than delivering them in their entirety (at once).
- This transmission form is commonly called streaming

HTTP - HyperText Transfer Protocol

- HTTP is a text-based stateless protocol, controlling how resources, e.g., HTML documents or images, are accessed
- HTTP builds on the TCP/IP stack, where the service is normally offered over port 80
- Resources are addressed by using the concept of a Uniform Resource Identifier (URI)
- A URI allocates unique identifiers to resources, regardless of their type

HTTP - II

- URL is on of the most common types of URIs
- A URI, e.g., (http://www.uni-karlsruhe.de/Uni/index.html), typically describes three things

HTTP - II

- URL is on of the most common types of URIs
- A URI, e.g., (`http://www.uni-karlsruhe.de/Uni/index.html`), typically describes three things
  - how a resource is accessed, e.g., (`http://`) if HTTP is used
  - the destination computer (host) where the resource is located, e.g., (`www.uni-karlsruhe.de`) and,
  - the name of that resource, e.g., `Uni/index.html`
- URIs also define a query delimiter, "?", which allows HTTP to pass on parameters.

HTTP-GET Method

GET

- The Get is the simplest http method.
- Its main job is to ask the server for a resource.
- Total amount of characters in a GET is limited.
- In GET method the data we send is appended with the URL
- Data is visible to other users

HTTP-POST Method

- The POST method is more powerful request
- By using POST we can request as well as send some data to the server.
- Data is sent along with the body of the request/content
- Normally we define the GET/POST method in an HTML form

Session Tracking

- The term *session* is used to define such a sequence of related HTTP requests between a specific user and a server within a specific time window.
- Web server cannot automatically allocate incoming requests to a session
- Two ways to achieve this:
  - In each of its requests to a server, the client identifies itself with a unique identification. This means that all data sent to the server are then allocated to the respective session.
  - All data exchanged between a client and a server are included in each request a client sends to a server, so that the server logic can be developed even though the communication is stateless.
- Session tracking is normally implemented by URL rewriting or cookies.

URL Rewriting

- URL rewriting is a mechanism that transmits session-relevant data as parameters in a URL
- Disadvantage
  - If a large data volume is required within a session, then the URL can easily become messy and error-prone.
  - Limiting the length of a URL
  - URLs have to be dynamically generated for each page and request

Cookies

- Cookies are small text files used to store server information (e.g., a session ID) on the client computer.
- information is written to a text file in the form of name-value pairs.
- Cookies are generated by Web servers and transmitted to clients in the header of the HTTP response.
- The major benefit of cookies is that information identifying a session can be exchanged transparently between a client and a server.
- Drawback of cookies is that some users deactivate the cookie functionality in their browsers to prevent their browsing behaviour from being captured.

Further Reading and References

1. Chapter 6, "Technologies for Web Applications" of *Web Engineering* book.