Part I

Web Technologies for Interactive Multimedia
## Chapter 2: Interactive Web Applications

2.1 Interactivity and Multimedia in the WWW architecture

2.2 Client-Side Interactivity and Multimedia (Example HTML5)

2.3 Interactive Server-Side Scripting (Example PHP)

2.4 Data Storage in Web Applications (Example Database Access in PHP)

2.5 Integrated Server/Client-Side Scripting (Example jQuery/AJAX)
Dynamic Web Contents

• Content shown to user in browser is dependent on some external variables
  • Examples of external variables:
    – Date and time
    – Contents of an information archive (e.g. recent news)
    – Actions of the user
      » Pointing to elements
      » Clicking at a certain position
      » Filling out forms
  • Wide-spread applications:
    – E-Commerce
    – Interpersonal communication media (forums, discussion boards)
    – Mass media (news and other information services)
Server-Side vs. Client-Side Realisation

• Client-side realization:
  – Browser contains execution engine for scripts
  – Web server does not need to execute scripts
  – Script is sent to client as part of server response
  – Example: JavaScript

• Server-side realization:
  – Web server contains execution engine for scripts
  – Browser does not need to execute scripts
  – Script is executed on server and computes response to client
  – Example: PHP
Server Scripts vs. Client Scripts

**Client-Side Scripts** (e.g. JavaScript)
- Fast reaction times – *good for fluid interaction*
- Works also without network connectivity
- Independent of server software

**Server-Side Scripts** (e.g. PHP)
- Computation of page contents dependent on external variables
- Data storage on server – *good for accessing media archives*
- Access to central resources (e.g. for request processing)
- Independent of browser software
Web Architectures for Interactivity

• Early approaches: “Common Gateway Interface (CGI)”
  – Informally defined, programs invoked to create HTML code
  – Drawbacks: Security problems, high processor load (separate process)
• Later: Web server software add-ons
  – Interfaces to common scripting and programming languages
e.g. Java, Perl, Ruby, PHP
  – Scripting languages specifically designed for Web development
e.g. PHP
• Web server software integrated with specific execution environments (“Application Server”)
  – Complex, highly optimized for good throughput
  – e.g. Servers for Java Enterprise Edition, Microsoft .NET framework
• Trend: Web servers written in I/O-efficient languages
  – e.g. Express server written in JavaScript (Node.js)
Media Support – Functions of Client Only

• Media rendering:
  – Recognition of media file types
    » MIME registry of browser
  – Local media playing software
    » Plugins or separate programs

• Interactivity:
  – Local interactions
    » Highlighting, dynamic menus etc.
Media Support – Functions by Server Only

- Media rendering:
  - Storage of media files and meta-information
  - Indexing and querying
- Interactivity:
  - Interactions with server-side effect
    » E.g. database updates (registration, buying, ...)
  - Interactions with global effect for all users
    » E.g. adding a comment, uploading a video
Media Support – Functions by Client & Server

- **Media streaming:**
  - Playback of incomplete content in client
  - Play-out in defined order from server
  - Synchronization, rate control, buffering
  - Flow control (stop, start, pause)
  - Adaptation to network conditions
- **Interactivity:**
  - Near real-time interactions
    » E.g. status notifications, data ticker
Chapter 2: Interactive Web Applications

2.1 Interactivity and Multimedia in the WWW architecture

2.2 Client-Side Interactivity and Multimedia (Example HTML5)

2.3 Interactive Server-Side Scripting (Example PHP)

2.4 Data Storage in Web Applications (Example Database Access in PHP)

2.5 Integrated Server/Client-Side Scripting (Example jQuery/AJAX)

Literature:

Embedding Media in HTML

• Media embedding requires:
  – Media data (a file)
  – Player software

• Typical media data:
  – Sound files (e.g. .wav, .mp3, .ogg, .midi)
  – Movie files (e.g. .avi, .mov, .ogv, .flv)
  – Programs to be executed on a virtual machine ("universal player"), e.g.:
    » Java applets
    » Flash runtime code (Shockwave Flash, .swf)
    » Silverlight application packages (.xap)

• Browser integration:
  – Built-in: Browser "knows" about player for media type
  – Plug-in: Flexible association between player and media type

• Incompatibilities in older versions of HTML
  – embed by Netscape, object by W3C & Microsoft, strange combinations!
HTML 5

• HTML Version 5
  – Draft W3C standard (most recent draft 25 October 2012)
  – Developed in parallel to XHTML 1.0
    » XHTML 2.0 development has been stopped
    » XML representation of HTML5 exists ("DOM5")
• HTML 5 is partially supported already by most modern browsers
• HTML 5 contains standardized and simple media embedding tags
  – audio
  – video
  – embed
Audio Embedding in HTML 5

• Example:

```html
<html> ...
  <body>
    ...
    <audio src="nightflyer.ogg" autoplay>
      Your browser does not support the <code>audio</code> element.
    </audio>
  </body>
</html>
```

• Attributes (examples):
  – autoplay: Playback starts automatically
  – controls: Control UI elements are made visible
  – loop: Plays in an endless loop
  – preload: Hints about preloading expectations

• Subelement <source>:
  – Alternative way to specify data source
  – Multiple occurrence is possible, first supported version is taken
Video Embedding in HTML 5

• Example:

```html
<html>
  <body>
    <video controls>
      <source src="big_buck_bunny_480p_stereo.ogg" type="video/ogg">
      <source src="big_buck_bunny_480p_h264.mov" type="video/quicktime">
      Your browser does not support the <code>video</code> element.
    </video>
  </body>
</html>
```

• Additional Attributes compared to `<audio>` (examples):
  – height, width: Dimensions of video image
  – poster: Image to be shown until first frame becomes available

• Events (can be handled e.g. with JavaScript, examples):
  – empty
  – canplay
  – ended
  – abort
  – volumechange
<embed> in HTML 5

- HTML 5 contains a standardized version of the <embed> element
- Purpose:
  - Embed arbitrary content played back via plug-in software
- Examples:
  - Flash content
  - Java applets
- Not intended for media playback
Video Codecs and HTML5 Video

• HTML5 Working Group: All browsers should support at least one common video format
  – Good quality & compression, hardware-supported, royalty-free!

• Problems with mainstream formats:
  – Patents on H.264 and its successor HEVC/H.265
  – Fear of hidden patents for Ogg Theora

• Google:
  – Release of WebM to the public (after purchase of On2)
  – WebM container format based on Matroska container
  – VP8 video format with Vorbis audio (current)
  – VP9 video format with Opus audio (in preparation), open, royalty-free

• Patent battle between Google and Nokia on VP8

• Still no simple common solution for the key manufacturers available
  – Neither H.264 nor VP8 fully supported by all browsers on all platforms
  – H.264 appears to be in the best position currently
Client-Side Interactivity with HTML5

- Browser-executed scripting languages
  - JavaScript, mainly
- Processing of user input
  - Event handling for mouse and keyboard input
  - Additional controls
- 2D graphics drawing
  - `canvas` element
- Animations
  - JavaScript frameworks, e.g. jQuery or JSCreate
HTML5 Interactive Controls

• Standard controls for interactive applications have been integrated into HTML5
  – “range” element (slider control)
  – “color” element (interactive color picker)

• Potential:
  – Higher client-side (stand-alone) interactivity
  – Typical applications: Drawing, image editing
  – See discussion of “canvas” element below
Example: Slider in HTML5

```html
<!DOCTYPE html>

<html>
  <head>
    <title>Slider in HTML5</title>
    <style type="text/css">
      input[type=range]:before {content: attr(min)}
      input[type=range]:after {content: attr(max)}
      input[type=range]
        {width:500px; color:red; font-size:1.5em;}
    </style>
  </head>
  <body oninput="current.value=slider.value">
    <input name="slider" type="range" min="1" max="11"/>
    <output name="current">5</output>
  </body>
</html>
```
HTML5 Canvas

• “HTML5 Canvas is an immediate mode bitmapped area of the screen that can be manipulated with JavaScript.” (Fulton/Fulton)

• 2D Drawing Context:
  – Object associated with a Canvas object
  – Used as handler in JavaScript to address the canvas (drawing API)

• Typical drawing primitives:
  – Draw shapes
  – Render text
  – Display images
  – Apply colors, rotations, transparency, pixel manipulations, fills, strokes

• (Pure) Canvas works on (low) pixel level
  – Browser redraws whole canvas each time the Canvas is modified using JavaScript
  – “Retained mode” rendering is provided by JavaScript libraries (e.g. EaselJS, part of CreateJS, see http://www.createjs.com)
Example: Drawing on the Canvas

```html
<!doctype html>
<html>
<head>
  <title>Canvas Hello World</title>
  <script type="text/javascript">
    window.addEventListener("load", drawScreen, false);
    function drawScreen() {
      var c = document.getElementById("theCanvas");
      var ctx = c.getContext("2d");
      ctx.fillStyle = "lightgrey";
      ctx.fillRect(0, 0, c.width, c.height);
      ctx.font = "italic bold 32px sans-serif";
      ctx.fillStyle = "red";
      ctx.fillText("Hello World!", 50, 50);
    }
  </script>
</head>
<body>
  <canvas id="theCanvas" width=300 height=80>
    Your browser does not support Canvas!
  </canvas>
</body>
```

`canvashello.html`
Example: Interactive Gradient (1)

```html
<!doctype html>
<html>
<head>
  <title>Canvas Gradient Fill</title>
  <meta charset="UTF-8">
  <script type="text/javascript">
    window.addEventListener("mousemove", drawScreen, false);
    function drawScreen(event) {
      var c = document.getElementById("theCanvas");
      var ctx = c.getContext("2d");
      var mx = Math.min(event.clientX, c.width);
      var my = Math.min(event.clientY, c.height);
      var grad =
        ctx.createRadialGradient(mx, my, 0, mx, my, c.width*1.5);
      grad.addColorStop(0,"#f00");
      grad.addColorStop(1,"#00f");
      ctx.fillStyle = grad;
      ctx.fillRect(0, 0, c.width, c.height);
    }
  </script>
</head>
<body>
</body>
</html>
```
Example: Interactive Gradient (2)

...  

```html
<body>
  <canvas id="theCanvas" width=500 height=500>
    Your browser does not support Canvas!
  </canvas>
</body>
</html>
```
Chapter 2: Interactive Web Applications

2.1 Interactivity and Multimedia in the WWW architecture

2.2 Client-Side Interactivity and Multimedia (Example HTML5)

2.3 Interactive Server-Side Scripting (Example PHP)

2.4 Data Storage in Web Applications (Example Database Access in PHP)

2.5 Integrated Server/Client-Side Scripting (Example jQuery/AJAX)
Server-Side Script Language PHP

(Only an example for a server-side script language!)

• PHP:
  – Personal Home Page Toolkit
    » 1995, Rasmus Lerdorf
    » 2003, new by Zeev Suraski, Andi Gutmans
  – PHP Hypertext Preprocessor (recursive acronym, backronym)

• Current version: 5.6.1 (October 2014) [version 6 has been stopped]

• OpenSource project:
  – see www.php.net
  – Can be used and modified freely (PHP license)

• Syntax loosely oriented towards C
  – Variations of possible syntax

• Extensive function library
  – being extended by community

• Advanced and popular Web development frameworks based on PHP
Prerequisites for Using PHP in Practice

• Always (even if using just one computer)
  – Installation of a Web server
    » OpenSource: Apache
    » Microsoft Internet Information Server
  – Invocation of PHP always indirectly by loading pages from server (http://...)
    » Loading from local computer: http://localhost/...

• Installation of PHP software as plug-in for used Web server

• Very often also installation of a data base system (e.g. MySQL)

• Frequently used acronyms for specific configurations:
  – LAMP: Linux, Apache, MySQL, PHP
  – WIMP: Windows, Internet Information Server, MySQL, PHP
  – MOXAMP: MacOS X, Apache, MySQL, PHP
Activation of PHP Module in Apache

• Example (MacOS X):
  – Apache + PHP module are pre-installed
  – Apache server needs to be started...
  – Configuration needs to be updated (remove a comment sign)
• /etc/apache2/httpd.conf:

```plaintext
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2> for detailed information.
...
LoadModule bonjour_module libexec/apache2/mod_bonjour.so
LoadModule php5_module libexec/apache2/libphp5.so
#LoadModule fastcgi_module libexec/apache2/mod_fastcgi.so
```
Hello World in PHP

<!DOCTYPE html>

<html>
<head>
  <title>Hello World with PHP</title>
</head>

<body>
  <h1><?php echo "Hello World!"; ?></h1>
</body>
</html>

File hello.php
in Web server directory
Embedding of PHP into HTML

• XML style (used here):
  – Like *Processing Instructions* in XML
    `<?php PHP Text ?>`

• SGML style:
  – Widely used in older scripts
  – Not really recommendable: PHP language not specified
    `<? PHP Text ?>`

• HTML style:
  – Using HTML tag:
    `<script language="php"> PHP Text </script>`
PHP Syntax (1)

- Inheritance from shell scripts
  - Variables start with "$"
  - Some UNIX commands part of the language, e.g.:
    ```php
echo "Hello";
```
- Control statements exist in different versions, e.g.:
  ```php
  if (bedingung1)
    anw1
  elseif (bedingung2)
    anw2
  else
    anw3;
  
  if (bedingung1):
    anwfolge1
  elseif (bedingung2):
    anwfolge2
  else:
    anwfolge3
  endif;
  ```
PHP Syntax (2)

• Various comment styles:
  – One-line comment, C style:
    ```php
echo "Hello"; // Hello World
```
  – One-line comment, Perl style / Unix shell style:
    ```php
echo "Hello"; # Hello World
```
  – "One line" ends also at end of PHP block
  – Multi-line comment, C-style:
    ```php
echo "Hello"; /* Comment
    spreads over multiple lines */
```
   – Do not create nested C-style comments!

• Instruction must always be terminated with ";;"
  – Exception: end of PHP block contains implicit ";;"
PHP Type System

• Scalar types:
  – boolean, integer, float (aka double), string

• Compound types:
  – array, object

• Special types:
  – resource, NULL
  – Resource type: refers to external resource, like a file

• "The type of a variable is not usually set by the programmer; rather, it is decided at runtime by PHP depending on the context in which that variable is used."

(PHP Reference Manual)