UFCEKG-20-2
Data, Schemas & Applications

Lecture 2
Introduction to the WWW, URLs, HTTP, Services and Mashups

N. H. N. D. de Silva
(Slides adapted from Prakash Chatterjee, UWE)
Suppose all the information stored on computers everywhere were linked, I thought. Suppose I could program my computer to create a space in which anything could be linked to anything. All the bits of information in every computer at CERN, and on the planet, would be available to me and to anyone else. There would be a single, global information space.

*Tim Berners-Lee, Weaving the Web*
WWW : definition

The **World Wide Web** (abbreviated as **WWW** or **W3**, commonly known as **the Web**), is a **system** of interlinked **hypertext** documents accessed via the **Internet**. With a **web browser**, one can view **web pages** that may contain text, images, videos, and other **multimedia**, and **navigate** between them via **hyperlinks**.

*Wikipedia : World Wide Web*

Concept originally proposed by **Sir Tim Berners-Lee** (1989) based on earlier hypertext systems. Berners-Lee and Belgian computer scientist **Robert Cailliau** proposed in 1990 to use hypertext "to link and access information of various kinds as a web of nodes in which the user can browse at will", and they publicly introduced the project in December of the same year.
Mashups

- Originally a term for the sampling & mixing of two pieces of music together. Here the term refers to web applications which combine data from multiple sources to create added value sites.
  - on Wikipedia
  - Programmable Web run by John Musser tracks the emerging collection of mashups.

- Here we review the basic mechanisms for integration. Next week we will cover the basics of XML, one of the data formats widely used for integration and configuration.
Mashup pre-requisites

- HTTP protocol
- client - server interaction
- URI Schema
- HTML/HTML Forms - the simplest Mashup technique
- media type (Mime-type, content-type)
- URL Encoding
- Character encoding
HTTP

- **Request**
  - Query string, attached files and information about the client. Server can access all this data to determine the appropriate response.

- **Response**
  - Document formatted by the server, with a wrapper which identifies the kind of Content-type

- **GET**
  - Query string appended to URI - limited length, exposes the parameter names, easy to edit, use for development, formally only for requests which only read data and don't update.

- **POST**
  - Query string passed in HTTP request body, unlimited size, hides the interface, use for sending data to server for update

- **PUT**
  - Add a resource to the remote store

- **DELETE**
  - Delete a resource

Often authentication is required - username/password passed in the HTTP header.
HTTP interaction

- This sequence diagram explains the main processes in the HTTP Protocol. It is the foundation for much of the interaction on the web **Client-server interaction with HTTP**
HTTP interaction

- We can think of an HTTP request/response as a remote procedure call (RPC). There are other, more low-level mechanisms for RPC which are useful in special circumstances but the web is built around HTTP.

- Applications built on HTTP interaction are often called **RESTful**. REST is an abbreviation which stands for Representational State Transfer.

- Strictly this is a well-defined architectural style in which the HTTP operations are used in a specific restricted sense, and unique URIs identify each resource in the application.

- Informally it refers to any interface to a site in which all data is requested and transmitted via HTTP without any additional layers such as is found in SOAP and Web Services, and the state of the interaction is passed in the request.
Three essential technologies: uri, html & http

1. A system of globally unique identifiers for resources on the Web and elsewhere, the Universal Document Identifier (UDI), later known as **Uniform Resource Locator (URL)** and **Uniform Resource Identifier (URI)**;

2. The publishing language **HyperText Markup Language (HTML)**;

3. The **Hypertext Transfer Protocol (HTTP)**.
Anatomy of a URI

Uniform Resource Identifier (more general than URL). The structure of a URI is defined by the URI *scheme*. URIs are case-sensitive

```
```

- **http**
- **//www.example.com/modules/dsa/index.html**
  - **user info** - terminated by @
  - **hostname** - www.example.com
  - **port** - :80
  - **path** - /modules/dsa/index.html
- **year=2012** (query parameter)
Anatomy of a URI

- **scheme**: `foo``
- **authority**: `username:password@example.com:8042``
- **path**: `/over/there/index.dbt?type=animal&name=narwhal#nose``
- **fragment**: `nose``
- **query**: `type=animal&name=narwhal``
- **hostname**: `example.com``
- **port**: `8042``
- **userinfo**: `username:password``
- **urn**: `urn:example:animal:ferret:nose``
- **interpretable as filename**: `animal:ferret:nose``
- **interpretable as extension**: `nose``
- **interpretable as keys**: `type=animal&name=narwhal``
- **interpretable as values**: `nose``
- **hierarchical part**: `/over/there/index.dbt?type=animal&name=narwhal#nose``
URI Scheme names

- **http** - The most common scheme name - Hypertext Transfer Protocol. Typically web pages are requested and delivered using this protocol.
- **https** - secure HTTP
- **mailto** - an email address - usually handled by the browser handing responsibility to another application
- **file** - read a local file (but do not execute it)
- **ftp** - file transfer

*any others?*
URI hierarchical part

- **user info** - e.g. nhnisansa@gmail.com

- **hostname** – www.gmail.com - **converted** by DNS to an IP address - 74.125.230.214

- **port** - e.g. : 80 - default http port

- **path** - /modules/dsa/index.html
Query String

- Parameters passed to the script. Multiple parameters are passed in several common forms.
- Delimited values are positional and delimited by a special character such as ";".
- `/modules/dsa/index.html;2`
- Where the two parameters are `/modules/dsa/index.html` and `2`.
- Keyword/value pairs each parameter value is passed as a `keyword=value` pair, with pairs separated by `&`. This is the form used by HTML forms. The order of the parameters is not significant.
Fragment

- address a place within a document - place marked as

  <a name="fragid">
Uses of URIs

- Destination of HTTP request
  - a link in an HTML document **body**
    
    `<a href="http://en.wikipedia.org/wiki/URI">URI</a>`
  - a link in an HTML document **head**
    
    `<link rel="alternate" media-type="application/rss+xml" href="news.rss"/>`
  - typed into the location bar in a browser - or editing an existing URI
  - created in a browser by javascript
    
  - used by the Javascript AJAX technique to add interactivity to a web page
  - created by a server script e.g. PHP
    
    `$x = file("http://en.wikipedia.org/wiki/$term")`

- Unique id for a resource -
  - XML namespaces - [http://www.w3.org/1999/xhtml](http://www.w3.org/1999/xhtml)
  - semantic web resource id -
    
    [http://www.cems.uwe.ac.uk/rdffold/moduleRun/UFIEKG-20-2](http://www.cems.uwe.ac.uk/rdffold/moduleRun/UFIEKG-20-2)
URI re-write

- URIs are often re-written by the server e.g. using Apache mod-rewrite to map to a different internal location.

- http://www.cems.uwe.ac.uk/rdffold/module/UFI
  EKG-20-2

- re-written to
URI re-write

- This allows the actual server, file locations and script languages to be changed while providing a stable resource identifier.

“Any software problem can be solved by adding another layer of indirection.”

Steve Bellovin of AT&T Labs
Form interface to create URI

- The simplest way to reuse another application is to create a new form to create the appropriate URIs. To understand the application is to understand the interface, the scripts, the parameters to scripts and the range and meaning of parameter values.

- Here the example is a site in the US run by NOAA which gathers data on Weather observation stations at sea.
  - UK buoys
  - Buoy near Pembroke
  - Wind speed
HTML

- **Hypertext Markup Language (HTML)** is the language of the Web
  - **Hypertext** because the Web is a hypermedia system
  - **Markup** because documents are encoded using **text**
  - **Language** because HTML is used for **communications**
- **Markup Languages** are different from most file formats
  - many computer formats are binary encoded and not just text
  - *markup* allows structured documents to be encoded as just text
- Web data formats use markup as well as other encodings
  - **HTML** and **XML** are markup languages
  - **JavaScript** is also exchanged textually (but it's not markup)
  - images and other multimedia content is encoded as binary files
Text

- `<h1>`-`<h6>` are different levels of headings
- `<p>` contains paragraph text
  - whitespace and line wrapping are ignored
  - paragraphs are set as boxes containing a number of lines
- Text inside paragraphs can use additional markup (phrase markup)
  - `<em>` for emphasized text
  - `<strong>` for text with a strong emphasis
  - `<sub>` for subscript text
  - `<sup>` for superscript text
  - `<q>` for quoted text (try nesting quotes)
  - `<code>` for code examples
- rendering of all these elements is built into the browser
  - more sophisticated issues probably are more browser-dependent
More Advanced Text

- Quotations can be explicitly marked up as such
  - `blockquote` for block-level quotations
  - `q` for inline quotations (part of a block)
  - `cite` provides support for pointing to the source

- Preformatted text allows text formatting in the HTML source
  - `pre` leaves whitespace intact and usually uses monospaced fonts
  - `word` wrapping may be turned off by default
Lists and Tables

- HTML supports three kinds of lists
  - `<ul>` for **unordered lists** containing li
  - `<ol>` for **ordered lists** containing li
  - `<dl>` for **definition lists** containing dt/dd

- Tables are the most complex visual structure in HTML
  - `<table>` represents a table as a sequence of rows
  - `<tr>` represents a *table row* as a sequence of cells
  - `<td>` represents a table cell containing *table data*
  - `<th>` is a special cell containing *header data*
Images

- The Web is an open hypermedia system
  - *hyper* refers to the term hypertext for linked content
  - *media* refers to the fact that multiple media types are supported
- For a long time, the Web only supported text and images
  - images can be used in a variety of formats (GIF, JPEG, PNG)
  - audio and video are possible today, **but not part of the Web**
- Images are not part of a Web page, they are included by markup
  - *img* is an empty element for including images
  - *src* is a URI pointing to the image (often a relative URI)
    ```html
    <img src="../img/portrait.png" alt="Portrait">
    ```
Links

- Links are the most important feature of the Web
  - conceptually, the Web is one large hypermedia document
  - links are based on Web identifiers, the *Uniform Resource Identifier (URI)*
- `<a>` is a link *anchor* and links to a URI (the *link target*)
  `<a href="http://www.cems.uwe.ac.uk" title="CSCT UWE">CSCT</a>`
- URIs can have various forms
  - `http:` points to resources available on Web servers
  - `https:` is the same but uses encrypted connections
  - URIs can use a variety of other *[URI Schemes]*
  - URIs can be relative (in the same was as file names)
  - relative URIs are evaluated relative to the URI of their occurrence
  - relative URIs can use path segments such as `/` and `..`