Course review

Lecture 27

CS 638 Web Programming

Most important areas

- Web documents
- Handling data
- Large systems/applications
- Server-side programming
- Client-side programming
- Computer networking

Web documents (1)

- Web pages
  - Human-readable markup languages HTML, XHTML, (XML)
  - Tags, attributes, URLs, hyperlinks, forms
  - Style information and images separated in different files
- Style sheets
  - Text-based format: CSS (rules, selectors, attributes)
  - Can be shared by multiple pages (from same site)
- Images
  - Formats for photos (JPEG) or other graphics (GIF, PNG)
  - Use of lossy compression to reduce size
  - Backgrounds, logos, often shared by multiple pages
Web documents (2)

- Advantages of separating content from presentation:
  - Easier to adapt document to multiple target audiences
  - Easier to have teams of specialists working together
  - The control over the appearance of the document split between author and the user viewing it
    - Tables, absolute positioning, divs, font sizes
  - Extensibility of markup languages key to the evolution of web standards
    - Forward and backward compatibility
    - Non-standard extensions

Handling data (1)

- Relational databases represent data as tables
  - Both for sets of entities and relationships between entities
  - Key is unique for each row
    - "Foreign key" is key for another table
    - Can combine multiple tables using joins on keys
- Querying the database using SQL
  - Powerful declarative query language: joins, filtering out rows that do not match a condition, grouping & aggregation
  - User specifies what result she wants, not how to compute it
    - Database software finds efficient way of getting the results

Handling data (2)

- XML can represent any type of structured data
  - Well-formed of XML documents – syntax stricter than HTML
    - Proper nesting, explicitly closing tags, all attributes' values quoted, single root
  - All XML documents can be represented like a tree
    - Basis for DOM – objects representing XML document
  - DTD describes specific tags and structure for XML documents holding data of a given type
    - Valid XML document conforms to rules described by DTD
    - XHTML = XML documents conforming to DTD from W3C
Large systems/applications(1)

- Web applications have three-tier architecture
  - Persistent data storage handled by relational database
  - Processing of data (a.k.a. business logic) implemented by server-side code
  - Presentation of data implemented by (X)HTML with CSS
    - Client-side code typically part of presentation
- Applications (application server, browser) often have event-driven structure
  - Web page determines hierarchy of objects
  - Programmer handles events and/or defines new ones
  - Important to understand available libraries, structure of the application, and the meaning of events

Large systems/applications(2)

- Modularization is crucial for large systems
- Often use multiple languages, some better suited for a given task, some forced on us by circumstances
  - Object-oriented language with strong types (e.g. C# or Java on the server)
  - Scripting languages (e.g. JavaScript on the client)
  - Declarative languages (e.g. SQL for querying/updating databases, XSLT for transforming XML documents)
  - Pattern matching languages for processing data (e.g. regular expressions for strings, XPath for XML documents)
- Programmer must understand tools and frameworks
  - Programmer may have to understand older code (possibly written for different version of the platform)

Server-side programming(1)
Server-side programming (2)

- ASP.Net is a mature framework designed with programmer productivity in mind
  - Pages and code separated (.aspx & code-behind files)
  - Controls encapsulate HTML snippets, server-side code, and even client-side code
  - Complex event system make it easy to perform specific tasks and to extend framework by overriding events
  - Master pages allow better structuring of multi-page apps
  - Next lecture we will see Java-based alternatives
  - Web services allow enable remote procedure calls into server from web client or other servers

Client-side programming

- Client-side JavaScript allows direct interaction with user without posting data back to server
  - Strongly integrated with browser’s event model
    - Can define handlers for a variety of events
    - Can directly manipulate objects used by browser to represent page elements (DOM)
  - AJAX – interactive pages that behave more like desktop applications
    - Can make asynchronous requests to server
    - Individual elements of the page updated independently
    - Re-use of existing toolkits/libraries is convenient

Computer networking (1)

- The layered structure of the Internet
  - IP (network layer) delivers packets end to end
    - IP addresses identify endhosts, can use DNS names
  - TCP (transport layer) implements two-way, reliable, in-order delivery of bytes
    - Port numbers allow multiple services on same computer
  - HTTP (application layer) request-response protocol used to transfer web documents from server to client
    - Wide range of options encoded as header lines
    - Client can also submit data
Computer networking(2)

- Performance improvements supported by HTTP
  - Caching reduces delay
  - Compression reduces transmission time
  - Pipelining eliminates unneeded delays for multiple objects
- HTTP requests are independent (stateless protocol)
  - HTTP cookies stored on the client allow server to keep state and logically link client’s requests
  - Many cookies are stored on disk and survive reboots
  - Some privacy consequences
- HTTP authentication: credentials sent with each request, user prompted for password just once
  - Authentication can also be done with cookies

Computer networking(3)

- Protocols using cryptography can give various security guarantees
  - Privacy (protection from eavesdroppers)
  - Authentication (for server and client)
  - Integrity (unauthorized changes to documents detected)
- Cryptographic algorithms
  - Secret key (symmetric) cryptography
  - Public key (asymmetric) cryptography
  - Cryptographic message digests
- Cryptographic certificates are a source of trust
- TLS/SSL wedged between TCP and HTTP

Most important areas

- Web documents
- Handling data
- Large systems/applications
- Server-side programming
- Client-side programming
- Computer networking