The Emerging Grid

- grid: "...emergence of a new infrastructure upon which first science, and then the whole economy, will be built" (Larry Smarr)
- do for computing what Internet did for documents
- aiming at worldwide governance & standards body analogous to IETF: http://www.gridforum.org

Evolution of Web Services

- limited to HTML
- improve function via 'active pages'
  - those that require server to run command parser on requested HTML file and perform computing as needed
  - but still limited:
    - to what can be embedded in near-HTML pages
    - what if could generate pages containing tags not previously defined?
    - i.e., tags specific to particular application needs?

Working the Web

- current web pages written in html
- browser/client renders html
- html tags:
  - understood by both client to have particular meaning
  - who defines?
  - can you have new ones whenever you want?

Working the Web

- html tags:
  - understood by both client to have particular meaning
  - who defines?
  - W3C
  - can you have new ones whenever you want?
  - not really
  - html tags only describe layout of document contents

Beyond HTML

- web purchase example: buy a hard disk
- go to vendor site and
  - navigate through their web pages to find item
  - maybe you don’t like their price
  - go to another vendor site
  - navigate through their web pages to find item
  - you like price, but they’re out of stock
  - go to another vendor site...

Better Than Before...

- better way?
- software agent on your computer:
  - presents a form where you say what you want
  - it returns a summary of vendors that meet criteria, automatically
  - cannot (practically) use html to do this
  - how come?
...but need new language

- html only specifies layout
- to do order form: need tags that apply to content of document
  - e.g., identify it as a vendor query
  - e.g., indicate what part describes sought part
- need something that extends html capabilities:
  - eXtensible Markup Language, xml

xml

- xml, like html, uses tags mixed in with document content
- xml lets you define and create any tag you want in a document, e.g.,
  - xml: <parttagsquery>
    - type: computer hard disk
    - capacity: 60 Gb
    - speed: 7200 rpm
  - </parttagsquery>
- but who will understand these tags?
- who can you send this to?

xml

- xml tags used to describe content of document
  - not really used to describe layout
  - so how can you see the content of an xml document?
  - some html browsers have simple xml rendering ability
  - need another document that describes how to render the tags
  - the same content can thus be rendered differently depending on the style sheet that describes how to render
- what do these style sheet documents look like?

xml and xslt

- rendering is a transformation operation
  - e.g., from 'abstract' xml to plain text
- xslt can be used for any such transformation
  - e.g., from one xml document to a different xml document
- ability to transform depends on having an 'understanding' of the source xml document's structure
  - internal working representation: tree constructed from the xml

xml parsing

- an xml parser builds such a representation
  - e.g.,
    - Computer Hard Disk
    - part: Query
    - type: Hard Disk
    - size: 60 Gb
    - speed: 7200 rpm
xml parsers

- two popular parsers:
- Domain Object Model (DOM):
  - from W3C
  - passes through entire xml document, builds representation in memory (e.g., tree)
- Simple API for XML (SAX):
  - from XML-DEV
  - issues callbacks as tags are encountered during parsing
- freely available parsers of both kinds available for many platforms

xml: missing link

- have:
  - arbitrary xml tags describing content of document
  - ability to parse document into structure
  - can use xslt to transform into other structures
  - don't have:
  - anything that knows about
data types
  - legal/illegal values for tags we've invented
  - what if we name our xml tags the same as someone else's in a different document
  - where they don't mean the same thing?

xml: missing link

- need a guide to describe what our tags mean and what they are/are not allowed to have for values
- an xml schema describes:
  - data types appearing in xml document
  - content, e.g., values that are allowed
  - structure
  - allowed elements
  - what about DTDs?

schema vs. DTD

- Document Type Definition (DTD):
  - defines tags appearing in a document
  - any SGML, including html
  - does not provide info re.
    - values a tag may have, may not have
    - structure (e.g., hierarchy) of data
  - must be global
  - tags described in a DTD have that meaning everywhere
  - can't have arbitrary xml tags

xml namespace

- schema:
  - provides info about structure and content of data and tags
  - allows for tags to be qualified by namespace
    - so tag 'product_query' can appear in two different xml documents and be different in structure and content
  - qualified by, e.g.,
    - acme.com/product_query
    - ozm.com/product_query
Other Web Services Components

- so you have an xml document + schema...
- who do you send it to?
- how do you know if a particular target is able to understand and process your document, and send you a reply?
- how do you get your document there?

Other Web Services Components

- how do you know... destination’s capabilities?
- use Web Services Description Language (WSDL):
  - provides standardized way for a site to make known formats & protocols its service accepts

Other Web Services Components

- how do you know...
- use Web Services Description Language (WSDL):
  - provides standardized way for a site to make known formats & protocols its service accepts

Other Web Services Components

- how do you get your document there?
- use Simple Object Access Protocol (SOAP):
  - defines ‘envelope’ for web services communication
  - envelope mappable to http and other transport protocols
  - one-way msg protocol, allows for intermediaries to process or add to msg

Getting The Word Out

- suppose you have a special interest in x
  - e.g., news, sports, technical updates, AV equipment
  - how do you stay current?
  - discover/learn which web sites have content
  - visit them often
  - sites might prefer to be able to push their wares?
  - member subscription; they send email
  - specialized software ‘agent’ goes mining web site

Getting The Word Out
A Newer Model

How Build This?

- use specialized content agents
  - like crawlers, but targeted
- content-providing sites provide API for content access

How Build This?

- use specialized content agents
  - like crawlers, but targeted
- content-providing sites provide API for content access
  - content-providing sites provide data dumps from their content

A Better Way To Build This

- use standardized ‘blurb’ format embodied in a technology called RSS
  - Rich Site Summary
  - a.k.a. Really Simple Syndication
- like syndicating: publish material to some number of locations
  - e.g., like comics in newspapers, some TV shows
- RSS uses XML

An RSS Item

- RSS provides a set of items within a channel to interested readers
- an item looks like:

```xml
<titles>RSS Resources/titile</titles> <link>http://www.wiredreference.com/authoring/languages/xml/rss/</link> <description>
If you use XML, the Rich Site Summary (RSS) format has quickly become a de-facto format for distributing headlines of new content, tips and tutorials you need to get started using RSS.
</description> <pubDate>...</pubDate> <guid></guid> ...
```
RSS Channel

- channel provides set of items in some way related
  - e.g., most recent, same topic
  - up to 15 items per channel
- RSS element may contain at most 1 channel
- each channel must contain tags:
  - title channel’s title
  - description: brief text description of channel
  - link: an HTML URL to channel’s web site
  - language: language encoding for channel (e.g., en-us)
  - item: from 1 to 15 items

Too Good...

- RSS developed by Netscape for a service it wanted to provide: version 0.90
- then UserLand Software did work to simplify original: version 0.91
- further UserLand refinements: 0.92, 0.93, 0.94
- RSS-DEV adopted 0.90 and evolved it into version 1.0
  - based on RDF
  - UserLand most recently produced 2.0
  - most versions mutually incompatible

Using RSS: source-side

- to provide RSS feed from your site to others:
  - need XML file defining your RSS channel
  - need your httpd server to know about this file
  - can be served on demand
  - need to keep RSS file up-to-date as content changes on your site
  - can do by hand
  - many tools to semi- or fully automate

Using RSS: client-side

- RSS-clients
  - receive and render XML of the feed
  - maintain local ‘tracking’ info so can know which feeds user subscribes to
  - and (perhaps) also status of feeds
  - may integrate into browser
  - many current browsers will render the XML file as plain text; ugly, but readable, sort of

Sample RSS ‘page’

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<rss version="2.0" xmlns:content="http://purl.org/rss/1.0/modules/content/">
  <channel>
    <title>NPR News: Top Stories/title>
    <link>http://www.npr.org/topics/topic.php?topid=33</link>
    <description>NPR News: Top Stories</description>
    <language>en-US</language>
    <copyright>Copyright 2004 National Public Radio</copyright>
    <lastBuildDate>Sun, 28 Nov 2004 18:46:37 EST</lastBuildDate>
    <image>
      <url>http://www.npr.org/feeds/npr_news_22x22.gif</url>
    </image>
    <generator>NPR RSS Generator 1.0 (generator)</generator>
    <item>
      <title>NPR News: Top Stories</title>
      <link>http://www.npr.org/feeds/npr_news_22x22.gif?url</link>
      <pubDate>Sun, 28 Nov 2004 18:46:37 EST</pubDate>
      <description>
        generation of NPR RSS Generator 1.0 (generator)
      </description>
    </item>
  </channel>
</rss>
```
Better RSS XML Handling

- RSS ‘viewer’
- top picks as recommended by blogspace.com:
  - for Macintosh: NetNewsWire
  - for Windows: SharpReader
  - for Linux: Straw
  - for web: Bloglines
- common to use Python as part of receiving/rendering process
- a scripting language similar in some ways to Perl and Tcl

Aggregation

- RSS on server does publishing or syndication
- on client-side, collect feeds from multiple sources
  - ...aggregators collect news, weblog and other feeds over the web and aggregate them so the news items are readable from a single place, regardless of their source.”
  - http://www.nongnu.org/straw
- a site performing aggregation may issue the aggregate as its own RSS feed

Some RSS Feeds...

- recent earthquake information:
  - NEWS: http://www.qpr.org/rx/index.html
  - tech news: http://slashdot.org/index.rss
- very widespread use with blogs
- fastest growing need type of feed?

Some RSS Feeds...

- recent earthquake information:
  - NEWS: http://www.qpr.org/rx/index.html
  - tech news: http://slashdot.org/index.rss
- very widespread use with blogs
- fastest growing need type of feed?
  - the podcast
  - simple RSS-pushed feed
  - uses RSS 2.0 “enclosures”
- can be downloaded independently of main msg

RSS Standard?

- current RSS users should provide support for 1.0 and 2.0
- what about a single standard?
  - e.g., what’s the IETF say?

RSS Standard?

- what about a single standard?
  - e.g., what’s the IETF say?
  - IETF:
    - has “atompub” working group
    - no RFCs yet
    - but has Internet-drafts for proposed atom standard
      - The Atom Syndication Format
      - The Atom Publishing Protocol
      - Atom Feed Autodiscovery
Atom

from "draft-ietf-atompub-format-03."

<feed version="1.0" encoding="utf-8">
  <title>Example Feed</title>
  <link href="http://example.org"/>
  <updated>2003-12-31T18:30:00Z</updated>
  <author>
    <name>John Doe</name>
  </author>
  <entry>
    <title>Type-1</title>
    <link href="http://example.org/2003/12/1/atom03"/>
    <id>urn:uuid:2003-12-31T18:30:00Z</id>
    <updated>2003-12-31T18:30:00Z</updated>
  </entry>
</feed>