No slides.

Follow Coronel Slides

Add web architecture and security when appropriate

Note: ssh to www.cs.dartmouth.edu, then

```bash
tail -f /var/log/httpd/access_log
to watch www traffic.
```

Skip XML in first lecture and cover cloud. Then come back to XML (and JSON) in second lecture.

**XML**

Standard for data rep and exchange (originally Internet)
XML IS a LANGUAGE INDEPENDENT data model
Similar to HTML
Tags in HTML describe how to present the data, while XML tags talk about the format of the data.

Look at the XML data

[W3Schools Example](http://www.w3schools.com/xml)

Display and leave up

3 components
* tag elements - opening tag and closing tag
  * can be nested
  * The whole document is a tag element pair
* Attributes
  * Each element may have within its opening tag some attributes
    * attributeName = value
  * unlimited # of attributes, as long as they are unique
* Text - sort of the leaves of the tree
Relational vs. XML
* tables vs. hierarchical
* fixed schemas vs. flexible schemas, “self-describing”, extensible (“X”), optional
- Tough to have variable numbers of attributes in the Rel Model
- optional attributes lead to NULLs in Rel Model
* rel algebra and SQL are simple nice query languages, while XML is not as straightforward
* Rel Model is not ordering, the data in the table is unordered; XML seems to imply an ordering, especially with multiple attributes
* Rel Model is fully mature and efficient; XML often just on the surface, non-native models.

Well-formed XML
Single root element
Matched tags, proper nested (no interleaving)
attribute names within an element are unique

Tools exist for verifying that XML is well-formed

Displaying XML on the web
no good to just print it!
Use XSL to translate it into reasonable HTML for rendering
Extensible Stylesheet Language

“VALID” XML must adhere to Well structured XML rules
Valid XML follows two other rule files: Doc Type Descriptor (DTD) and XLM Schema Definition (XSD)
There are tools to establish XML validity (DTD and XSD)

DTD is a grammar like language that describes
what tags the XML can contain,
the ordering of tags,
the nesting of tags
the number of attributes a tag can have
The DTD also provides anchor-like pointers.

If you use a DTD/XSD, the appl program can assume VALID XML… saving code.
Also good to specify data interchange formats
Documentation also uses XML to specify detailed formats

DTD straightforward
* uses some grammar like constructs
- (Product+) meaning 1 or more
- P_INDATE? meaning optional
- #PCDATA meaning standard string data
- #REQUIRED means just that

Our systems have `xmllint` utility (as does OSX) which can validate XML and a whole lot more.
```
xmllint --valid --noout file.xml
xmllint --htmlout file.xml
```
Open XMPlify and show note.xml and then ProductList.dtd and xml
**Querying XML**

XPATH is the old standard, newer alternatives like XQuery are there. Since this is just an overview, we'll try XPATH.

Start XMPlify

Find the

XPATH queries: /bookstore/book/title

---

**JSON (“JASON”)**

This section closely follows a lecture by J. Widon of Stanford

JAVASCRIPT Object Notation

Use Github bookstore example

also http://www.w3schools.com/json/json_files.asp

Like XML it’s a LANGUAGE INDEPENDENT data model.

Good for semi-structured data.

New tech, thus far fewer tools outside or Prog Langs, but they're coming

Originally meant to represent object in a program in a human-readable form in files.

Used for data exchange, between programs.

Also used for representing software configuration info

Most languages have JSON parsers available.

**Language**

numbers, strings, T/F, NULL

Composites are objects and arrays

- Object is a set of label-value pairs enclosed in `{ }`
- Arrays are enclosed in square brackets, elements separated by commas

Like XML, objects can have optional members.

**Comparing to Rel Model**

JSON based on sets and arrays, not tables

Like XML, the JSON Schema isn’t required in advance, self describing like XML

XML is more verbose and more complex than JSON

* This could be due to its maturity
queries are hard, esp with the poor tooling
JSON is typically manipulated using programs.
QUery LAnguages are coming.

JSON is also used in some NOSQL Systems
* a format for input/output with those systems
* Document management systems, where documents may have JSON inside

Lots of overlap between XML and JSON, biggest difference is maturity of tools.

Syntactically valid JSON fit the label-value pairs and arrays

Look at JSONW3example.html and view it in browser and more.

```json
{"widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}}
```

the same stuff expressed in XML

```xml
<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
```

```xml..."Unknown (200): http://www.w3schools.com/xpat...
Auto-validate on refresh
```
1. Lecture notes based on texts by Coronel, Widom, Ullman, Jukic, and Silberschatz.