Uses of XML or JSON

Supporting arms-length relationships

- Exchanging information across software components, even within an administrative domain
- Storing information in nonproprietary format
- Representing semistructured descriptions:
  - Products, services, catalogs
  - Contracts
  - Queries, requests, invocations, responses: basis for Web services
  - System configurations

Compare with Lisp

List processing language

- S-expressions
- Cons pairs: **car** and **cdr**
- Lists as nil-terminated s-expressions
- Arbitrary structures built from few primitives
- Untyped
- Easy parsing
- Regularity of structure encourages recursion

Exercise

Produce an example XML or JSON document corresponding to

- An invoice from Locke Brothers for 100 units of door locks at $19.95, each ordered on 15 January and delivered to Custom Home Builders
- Factor in certified delivery via UPS for $200.00 on 18 January
- Factor in addresses and contact info for each party
- Factor in late payments

What is Metadata?

Literally, data about data

- Description of data that captures some useful property regarding its
  - Structure and meaning
  - Provenance: origins
  - Treatment as permitted or allowed: storage, representation, processing, presentation, or sharing
- Markup is metadata pertaining to media artifacts (documents, images), generally specified for suitable parsable units

Motivations for Metadata

Mediating information structure (surrogate for meaning) over time and space

- Storage: extend life of information
- Interoperation for business
- Interoperation (and storage) for regulatory reasons: supporting organizational coherence
- General themes
  - Make meaning of information (more) “explicit”
  - Enable reuse across applications: *repurposing* (compare to screen-scraping)
  - Enable better tools to improve productivity

Reduce need for detailed prior agreements

Metadata History

What kind and how much of prior agreement do you need?

- No markup: significant prior agreement
- CSV, Comma (likewise Tab) Separated Values: no nesting
- Ad hoc tags
- SGML (Standard Generalized Markup L): complex, few reliable tools; used for document management
- HTML (HyperText ML): simplistic, fixed, unprincipled vocabulary that mixes structure and display
- XML (eXtensible ML): simple, yet extensible subset of SGML to capture *custom* vocabularies
  - Machine processible
  - Comprehensible to people: easier debugging

Meaning of Information on the Web

Need to represent meaning to enable automatic processing

- Challenge: how can we produce representations that are rigorous yet comprehensible?
  - Humans rely on meanings of names (lexical tokens) for understanding
  - Computers work on uninterpreted symbols: the words don't matter but their interconnections do
- Relational DBMSs work best for rigidly structured information
  - But rely on column names for meaning
- Web information can be less rigidly structured
  - Same problem: reliance on names for meaning
  - Better opportunities to organize richer meaning representations
- Represent metadata through specification of a *vocabulary*, i.e., names organized through standardized relationships

Naming Conventions

Ways to systematically generate names

- MAC addresses
- Postal and telephone codes
- Vehicle identification numbers
- IP addresses and domains as for the Internet
- On the Web, use URIs for uniqueness

Namespaces on the Web

Essential for interoperation of heterogeneous resources

- Problem due to custom vocabularies and interoperation
  - Difficulty in identifying meaning
  - Risk of name collision
- A namespace is a set of names
- Namespaces must be identical or disjoint: no partial overlaps
  - Crucial to support independent development of vocabularies
  - Rely upon and provide a naming convention

Uniform Resource Identifier: 1

- URIs serve these main purposes
    - Identify resources we wish to access
    - Identify metadata of the resources
    - Identify namespaces using which the metadata is constructed
- URIs are abstract
- What matters is their (purported) uniqueness
- URIs have no proper syntax per se
- Kinds of URIs include
    - URLs, as in browsing: not used in standards any more
        - Formal syntax
        - A locating architecture: a way to resolve to a resource
    - URNs, which leave the mapping of names to locations up in the air
        - Formal syntax

Uniform Resource Identifier: 2

Good design requirements

- Ensure that the identified resource can be located
- Ensure uniqueness: eliminate the possibility of conflicts through appropriate organizational and technical means
- Prevent ambiguity
- Use an established URI scheme where possible

Web Architecture

Principles and constraints that characterize Web-based information systems

- URI: Uniform Resource Identifier
- HTTP: HyperText Transfer Protocol
- Metadata must be recognized and respected
  - Enables making resources comprehensible across administrative domains
  - Difficult to enforce unless the metadata is itself suitably formalized