PhiloWeb panel
“Philosophy” of the Web

Larry Masinter
April 17, 2012

Some credentials:
Chair, URI Working Group, IETF, 90’s
Chair, HTTP working group, IETF, 90’s
Chair, Scientific Advisory Board, RealNames Corporation
W3C TAG

Some PhiloWeb Observations

- URIs aren’t identifiers
- “Resources” don’t exist
- Persistence = meaning
- Naming is printing money
- Resources are angels, URIs are pins
- Languages ≠Specifications ≠ Implementations

Need better theories

- Meaning

  Need to talk about security, privacy, provenance:

  - Use communication model, not semantic model
  - Ontologies are backwards
- Identity
  - Individuals
  - Organizations
- Persistence

Economics and Meaning

- Economics
  - Economy of ideas
  - Ownership of names
  - Ownership of ideas
  - IPR
  - Indirect monetization
  - “Nation” => “Internet group”

Economics of naming

- ~~Being able to name something gives you power over it…~~
- Being able to tell people what a name means gives you power to control access to it
  - People think they're buying names, but buying a SLA to be the authority that people will use
    - domain names
    - DOIs
    - Selling certificates
  - Search Engines usurp name ownership
  - SOPA, PIPA force name resolution

MIME gives the **web**: **persistent names** for **languages**

- – “persistent”
- – “name for”
- – “language”

Language, File Format, Protocol, Interface

- A language is a way of giving meaning to data
  “Given some data, what does it mean?”
- “**File format**”: a kind of language
  (binary) languages
- Languages have **syntax** & **vocabulary**
- Languages usually use other languages
  – **protocol element** (a little language)
  – **abstract language** (defined in terms of structure)
  – **layer** (SVG on XML on Unicode)
- “URI” is a language, JavaScript, CSS are languages

What is a name?
How does MIME name languages?

- A name is protocol element
  - with some structure
  - used in other languages, protocols, apis, interfaces
  - Which has some meaning
- Meaning of MIME types
  - “which language should be used to interpret this data”

Persistent names

- languages change: how can names be **persistent?**
- With no evolution, updates, extensions to languages used in the web: no problems

CORE

- ***How do languages change?***
- ***What are problems with MIME during evolution?***

Languages and Implementations

- Languages (as with protocols, protocol elements, file formats, APIs) are used between systems to communicate
- Systems using a language should mean the “same” thing
- Need agreement between the systems that are communicating

Interoperability is a property of implementations, not specifications

Languages and Specifications

- **Specifications** are documents that describe a language and rules for implementations
  - How implementations should “understand” the language/API/protocol/protocol element’

Implementations to guide and validate single-user

- *Many* specifications used to define a single language
- What happens as those evolve?

Standards for Languages

- **Standards** represent agreements among implementations (in the form of a specification)

Persistent names for languages

- What is persistent about the name for a language?
- What is it that the name of a language identifies?
- How do languages evolve, grow, change over time?
- How can the name be persistent when the meaning changes?

Persistence and Evolution

- When a language evolves, it keeps its name
- A new language, even if it isn’t very different, would get a different name

Wait…

- How do languages evolve?
- What happens to systems that use those names with evolving meaning?

“language” is over-simplification

- Languages (file formats, protocols, protocol elements) are defined in terms of others
- Complex structure of interrelationships between components
- Each component can evolve independently

Implementations evolve

- The language is “as spoken”, not “as defined”
- Concrete and abstract languages
- References to other specification
- Syntax and parsing

specifications describe Languages

- References in specifications: how do rules apply when referenced specification is updated
- Editions, version numbers

More complexities

- Content negotiation
  - Different “representations” for same “resource”
- Polyglot
  - Same content in multiple languages
- “multi-view”
  - Same content, different views, treated differently

Registry

- A way of naming something
  - Organization to manage registry
  - Key role of registry is to manage updates
    - When there are compatibility requirements
    - When there are requirements
- Ontology
  - A kind of dictionary / registry
  - Attempt to be proscriptive

Persistent name problems

- Forking (HTML)
- Versioning (javascript)
- References
- Compound languages (HTML + RDFa/lite + SVG + MathML)
- Layering

- Generalization: other “persistent names”:
  - Charset (addition of Euro)
  - Other web names (codes, URLs)

Content negotiation

  - *Which languages do you understand?*
  - *Which languages can you speak to me?*
- MIME types don’t help much
  - Wrong level of granularity
  - Ambition of reader implementers doesn’t match conservative requirements of senders

Persistent names and versions

- “version” parameter requires future proofing
- In-band version identifiers might be preferable
  - Except for “quirks mode” failure cases
- Users would like “version of language”
- Best a specification can give is “version of specification”
- Specifications and languages often don’t evolve in sync

Being able to name language = control over language

- Politics / Economics of standards

- “Owning” the standard
    - Keep others from disadvantaging your products or services
    - Perhaps allows you to advantage your products and services over others

Wealth of Nations

- Boundary of nations
  - Internet communities transcend
  - Social organizations over the Internet
  - Governance in a global community