# Data submission hubs – without a giant standard

**Arnon Rosenthal**

**arnie@mitre.org**

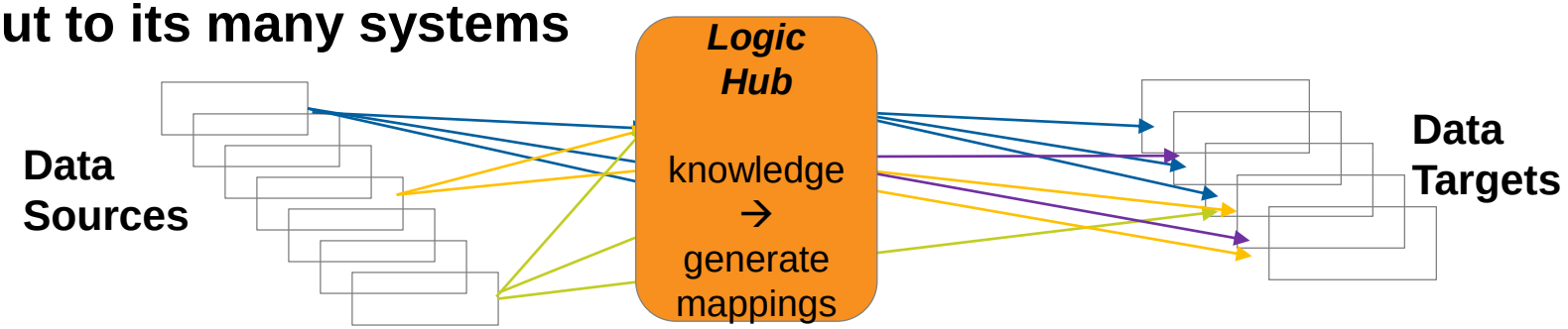
**Cambridge Semantic Web Meetup**

**October, 2016**

DISCLAIMER: The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

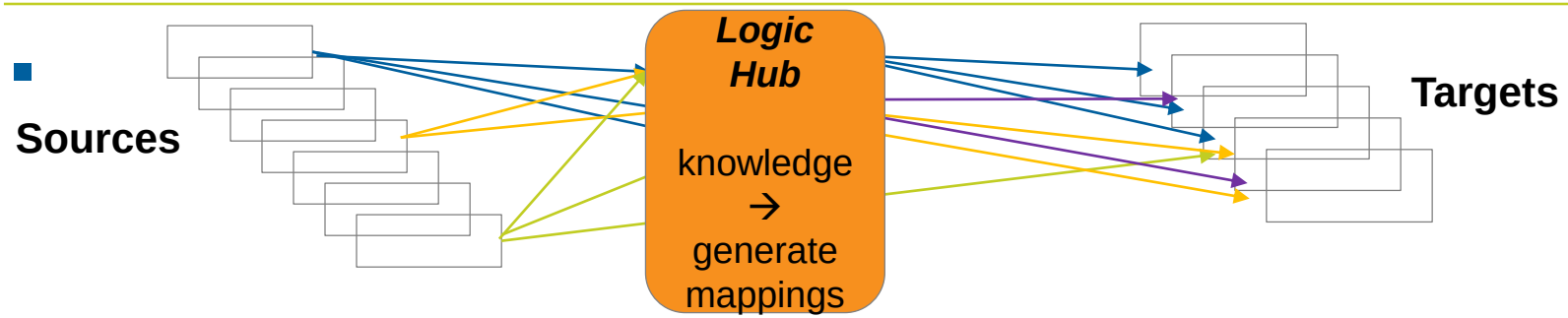
# The Problem

- **Help *many* submitters to submit data to an organization, and out to its many systems**




- **Example: CDC (Centers for Disease Control and Prevention) has**
  - 100+ topic-specific systems for submitting health information
  - Feeds from 50 states, plus others (often out of good will, not funding)
  - Many internal organizations, and many target systems in each organization
    - The main funding is within the line centers, not with HQ. They cannot *separately* afford major investment in interoperability skills and tools

# Objectives (long term)




- **Desired technical outcomes**
  - **Agility --** Empower domain experts to handle many cases of adding new sources, consumers, and data elements
    - **Auto-generate mappings** from *reusable* knowledge (+ COTS)
  - **Avoid giant "compromise" standards.** Each partner selects mini-vocabularies natural to them and extends ontologies as needed
- **Desired "best practice" and business outcomes**
  - Start organizations at using metadata-driven *data integration COTS*
  - Cut years-long delays
  - Cut submitters' costs *and agencies' costs to feed backend systems*

# Why it matters

- ***Many* organizations face the "many submissions" challenge, some for structured data from external partners, some for their data lake**
  - E.g., CDC, FDA, SEC, DHS, …
- **With current approaches, less data is available to consumers, and expenses are high for IT staff**
  - Self-service data analysts spend ~60-80% of the time on data prep
- ***Progress on improving data engineering and data integration has been glacial***
  - The usual data engineering practice is MS Office (Excel, Word, PPT)
  - Government adoption of metadata repositories (knowledge bases) and COTS data sharing tools is very limited
  - Submission hubs are a great *initial* place for tools –one stakeholder receives ROI for simplifying many data exchanges

# Background: How is it done today?

## *At many (most?) agencies, it's still 1998*

**Technology insertion of COTS data tools has been glacial**

- **Use of few data control and mapping tools (other than DBMSs) that are driven by knowledge about systems and their relationships**
  - *Humans* process the knowledge in Excel, Word, PPT, etc.
- **Little incentive to invest in *agility* via knowledge capture and tools (even though 70% of costs are "maintenance")**
- **Exchange is via physical data exchange standards– often a big XML schema**
  - Long negotiations, one size fits all. Lose specificity
  - Model formalisms capture too little (semantics, codesets)
- ***Many* wrappers, each hand coded**
  - Each submitter creates at least one
  - *Agency* creates a wrapper to each target system
- **Consequences: Costs are high for submitters and agency**
  - Change is resisted because it's expensive, and takes years

# Typical mapping approach:
# Create a big standard, and all map to it

***Knowledge structure***

- **One global standard (XML or ontology) covers everything**
- **Each area is modeled once within the ontology, e.g., Events, Diagnoses, Places, ….**
- **Everyone integrates using the same standard, *or else* they develop a wire format for each set of content**
- **Each change requires coordinating many partners**

- **Often uses XML schema as the (very poor) modeling formalism**
  - XML schema does not describe relationships or specializations

- **NIEM provides a bit of decentralization** **(see slide 8)**

# Summary of problems with BIG standards for BIG communities

- **Slow and costly to develop a standard**
- **2+ year change cycles**
  - Each change requires long negotiation
- **Needs of small subcommunities (and agile piloting) are not met**
  - Large committee won't tackle additional areas, nor will they provide extra specificity (80% rule)

- **If one codes wrappers manually (as our customers do), huge costs and delays till they are recoded**
  - Even power users can't do even the smallest extension
  - XML approaches don't express or exploit (X generalizes Y)

# More state of the practice: NIEM distributes authority to large domains, not to small groups

- **NIEM (National Information Exchange Model)**
  - A tree of vocabularies, plus tools for managing them
  - Use these to create an XML schema when exchange is needed

***Good (green)*** ***Bad (red)***

- **Has substantial buy-in – organizations reuse NIEM definitions**
  - Decentralizes vocabulary creation a bit
- **Splits the world into still-gigantic pieces (e.g., Justice, Health), managed by a heavyweight committee process**
  - These are hardly small, easily learned, agile units
  - Definitions are far from the user communities
  - No IS-A among concepts
- **Tooling creates UML models and wire formats for exchanges, but gives no help in defining or wrapping databases in systems**

# What we want, instead

- ***Avoid the effort and rigidity of giant vocabularies that require community buy-in***
  - Promote use of familiar localized vocabularies
  - Enable local extensions/evolution
- ***Empower domain experts* to do routine descriptions** (replacing programmers who code data mappings)
- **Generate mappings between systems, largely automatically**
  - (Yosemite does this between *standards)*
- **Break the adoption logjam**
  - For submission hubs, interoperability is central, and *many* exchanges are constructed. So agencies have incentives to invest
  - Once the tools are licensed and the metadata collected, others can use it

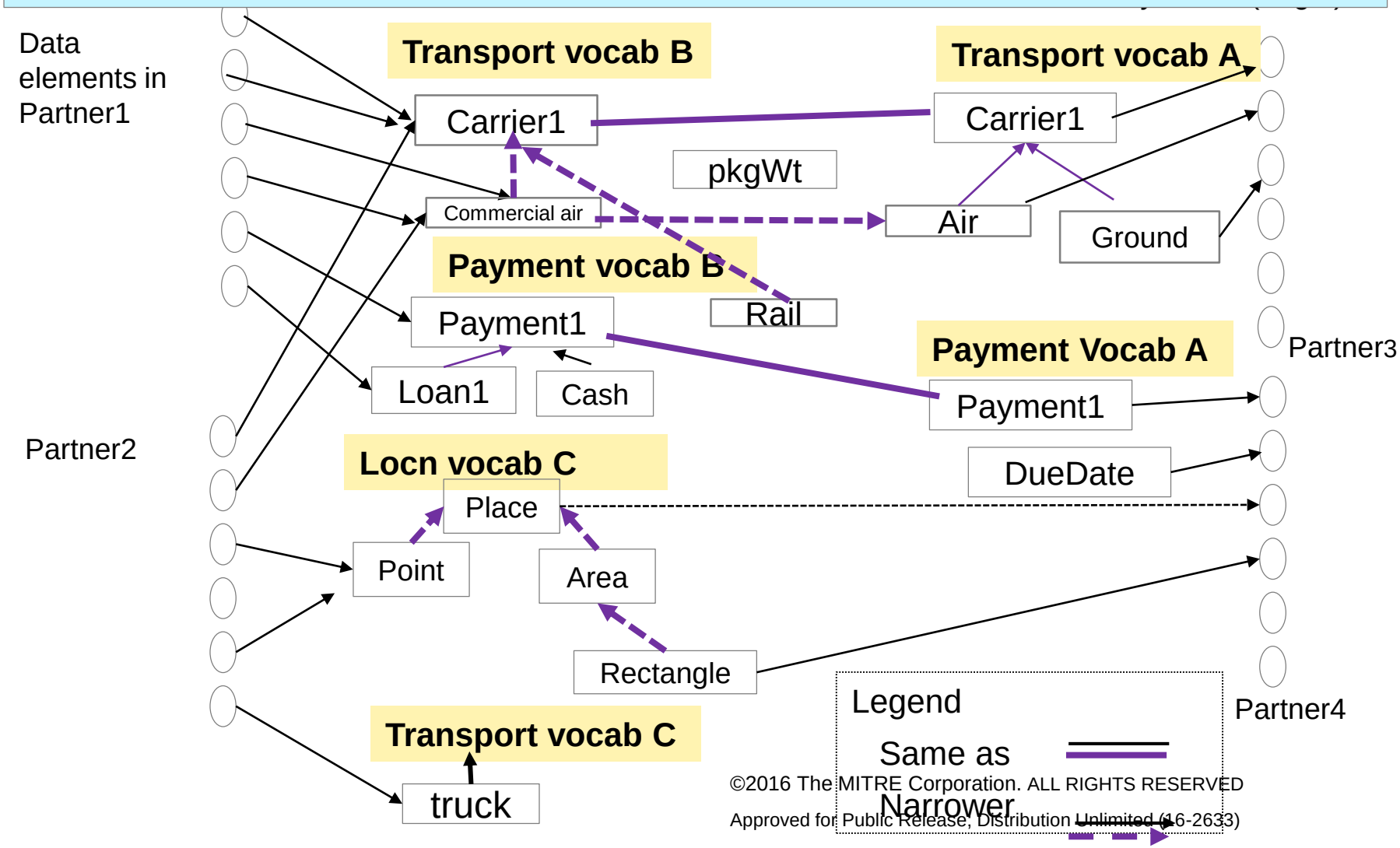
# How to do it?

- **Create a decentralized set of small-domain vocabularies (mini-ontologies) and links among their terms**
  - Right-sized -- manageable with local simplicity
  - Evolvable
  - Suited to each system (e.g., choose your favorite *Place* domain)
- **Curators extend and link the vocabularies. The union is a (redundant) over-arching ontology**
  - Can COTS or Protégé handle this?
    - Govt. agencies don't want to develop and sustain their own tools
- **Reasoners generate the data mapping, as best they can**

- **To break the tool-adoption logjam, focus on organizations where integration is a *critical* pain point**

# Scenario of usage (animated, see notes)

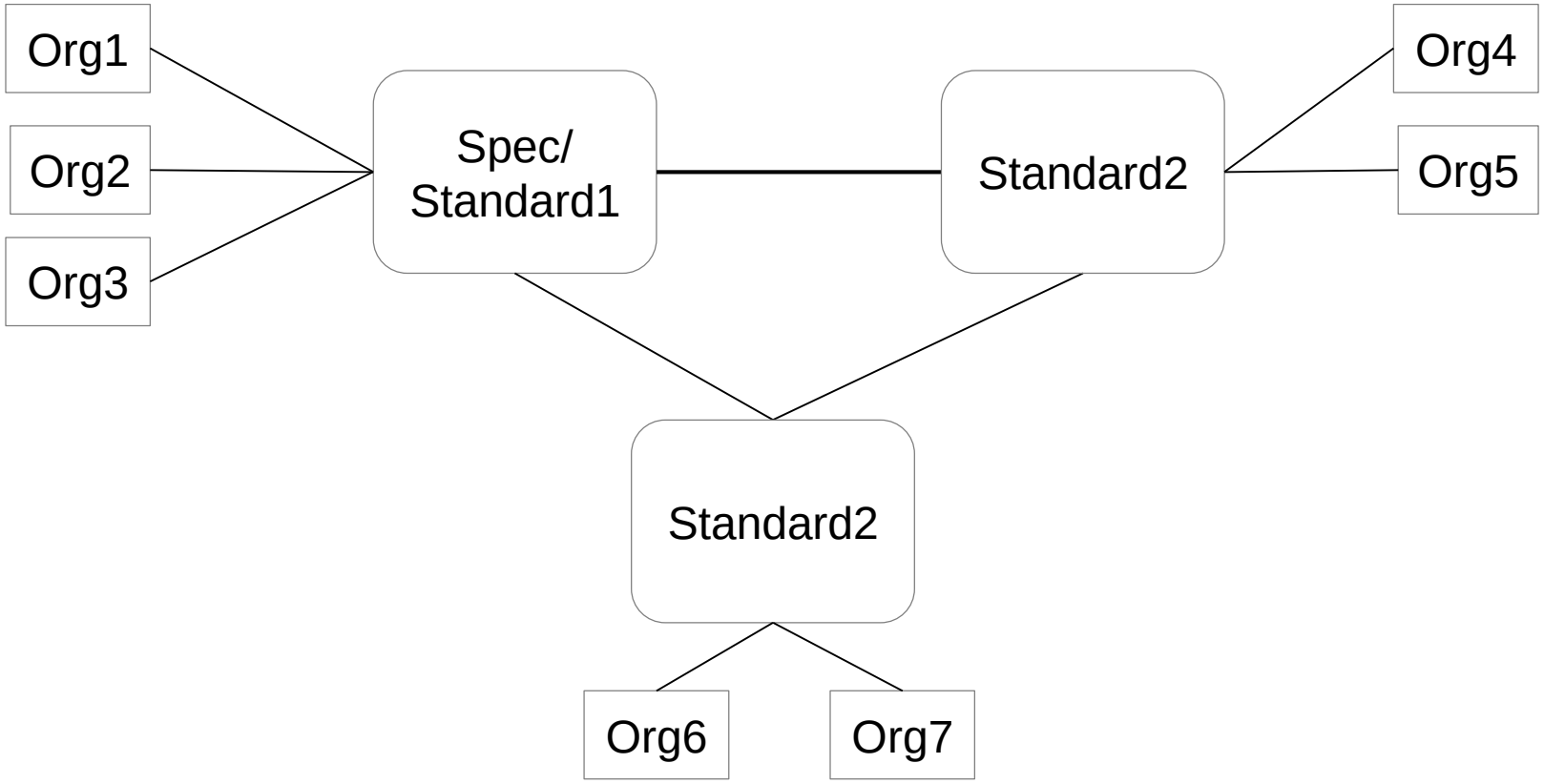
Conventional (central standard) approaches perform every knowledge-capture step shown here – but with farther-away vocabularies and no capture for reuse



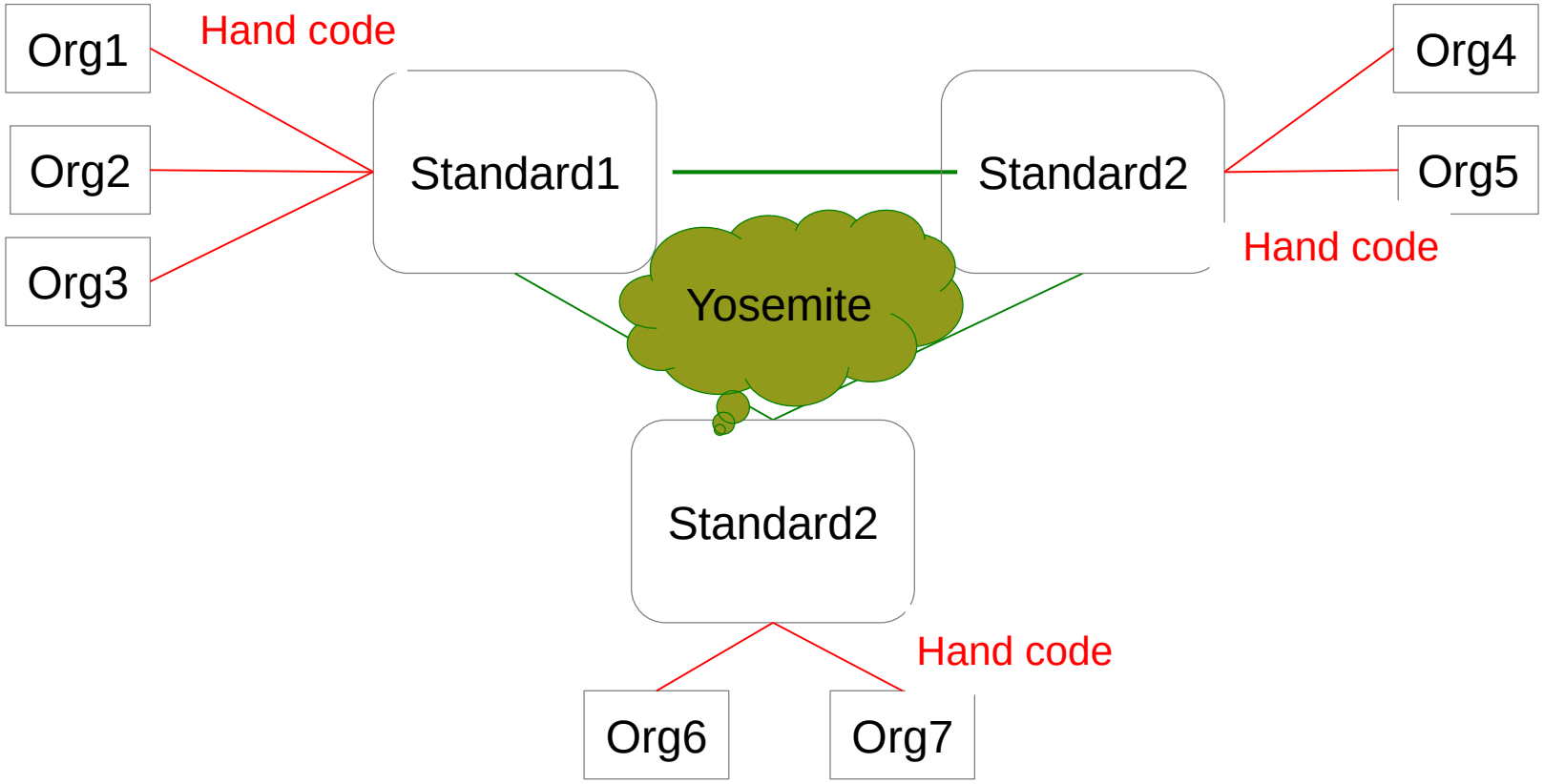
# Potential benefits

- **Domain experts can make small changes that satisfy local needs. This "killer app" can drive adoption**
  - E.g., add an attribute to an existing source
  - Bring in a new source that uses existing vocabularies
  - Add a term to a local vocabulary you understand. Relate it to partner's vocabulary
- **Not hostage to the standards committee** (which had limited coverage, too general for many needs, slow change)
  - Less effort for standards development, faster pace
- **Infer data mappings from item relationships, don't code them**
  - TopBraid, Anzo, Yosemite, IBM Infosphere all do this
  - Submission hubs offer promise as a place to get them adopted. Break the logjam; then reuse for other purposes

# Yosemite model



# Yosemite's contribution




- **Organizations want an end to end solution, to/from their own structures**
- **Yosemite generates data mappings among standards. But one still hand-codes between organizations and standards**

# Compare *Submission Hubs* to Yosemite

***A common aim: benefits of a central ontology, with few negatives***

- **Both: A web of ontologies and SKOS links among them**
  - A reasoner generates data mappings from concepts+links
- **Uses medium-big ontologies that represent existing standards (vs. small agile mini-onlologies that we control)**
  - They do well for systems that have been "tagged" w.r.t. *a* standard
  - We try for concepts that are easier to use for data not yet tagged
- **Yosemite connected *standards.* Our customers' aim is to ship data among *systems;* the story may not motivate them**
  - **Yosemite *principles* suffice to add the needed additional maps:** to reason from (source→standard, standard → target)
  - For creating a demo, public domain standard schemas were available

# Pros and Cons

## Submission Hubs

- **Mini-ontologies are scalable and changeable – it's natural to add more concepts**
- **Bleeding edge -- manage all mini-ontologies and links**
- **Can connect more data items, but must be opportunistic**
- **Takes data between *systems*, not just standards**
  - Query (pull) also works

## YOSEMITE

- **Existing standards can be awkwardly large, yet do not cover all data**
- **An established modeling approach, off the shelf tools**
- **Slow change makes behavior more predictable**
- **Goes between standards – to use it, must add mappings**
  **(source→ standard1)**
  **(target → standard2)**

# Next steps to advance the work

- **Examine existing ontology tools and model driven technology (e.g., Anzo, TopBraid)**
  - Add a separate declarative treatment of format and units
  - Generate code from the captured knowledge (using semantic COTS)
- **Identify first steps a customer could take -- *with positive ROI***
- **Create a paper, perhaps collaborating with Yosemite team?**

# Some research questions

- **Devise processes to extend and curate the mini-ontologies and links**
  - Manage change for an RDF property that spans mini-ontologies (e.g., Person *residesAt* Place)?
  - How to organize multiple name spaces, and "adopt" concepts across them, and who should see what changes?
  - Manage SKOS links as ontologies evolve
- **Create metrics for admin and coding labor, estimating with and without tools**
  - Do it for both initial setup and deltas
  - Estimate time to
    - Develop/extend mini-ontologies and gain adoption
    - Connect the mini-ontologies to systems and to each other

# Summary

- **Avoid the giant schema, and manually-coded wrappers**
  - Be faster, cheaper, and more flexible
- **Break the adoption logjam**
  - Those who adopt Submission Hubs can use the same metadata and tools in other integration scenarios
- **Enable power users to make modest extensions**
  - Build in some best practices. Curate more later

*Very* open to collaboration

# Backup

# **Where we fit** amid data integration aspects

- Invocation protocol (e.g., ODBC, REST)
- Data structure formalism (e.g., XML, SQL)
- Access controls
- **Concepts mappings (semantics)**
- **Value set mappings (e.g., zipcodes → cities)**
- **Value representation mappings (syntax, units)**
- Identity resolution (e.g.  Jon@example.com, Jsmith, John Smith)
- Data value merging (Height = 71”, 72”, 61”, 999999”)

***Weave element-mappings together into a mapping of whole datasets***

# A succession of industry approaches

- **Capture point to point element mappings, weave together (infer) a mapping of data structures (IBM)**
  - Walk in, capture the links, and demo
  - Uses sophisticated theory to weave together into a mapping of tuples
- **Create an ontology, link RDF representation of systems**
  - Potentially gives *many* connections, but the initial barrier (create ontology) makes it a much harder sell
- **Multiple linked ontologies (for standards), inference (e.g., Yosemite)**
- **Multiple evolving, overlapping ontologies and links**

- **Extend the semantic tool suite to handle**
  - Other configurations (peer to peer, rollup to a warehouse)
  - Format transforms
  - Data merging and cleansing

# A mediator is not magic

- **A mediator's job: The *information* is available … but *data* is not in desired structure, vocabularies, format, …**
  - Includes "desired info is automatically derivable", e.g., Area = L*W

- **In cases where a human couldn't write a derivation, a mediator won't either**
  - E.g., for city pairs, source might have "great circle" distance but not road distance