Overview of PID Systems for Digital Objects: Some High Level Considerations

August 2016

Views About PID Systems:
Training Course

Larry Lannom

Corporation for National Research Initiatives
http://www.cnri.reston.va.us/
http://www.handle.net/

PIDs for Research – Why Bother?

- Managing increasing amounts of primary and secondary data on the Net over long periods of time
- Managing increasingly complex data relationships on the Net over long periods of time
- When the attributes of that data such as location(s), responsible parties, and the underlying systems may change dramatically over time
- Science builds on past work and increasingly relies on collaboration within virtual distributed communities
- All of this absolutely requires reliable, long-term persistent references to bind together the distributed data, processes, and parties involved

PID Considerations – Big Picture

- No lack of unique identifiers in the world – that part is easy
  - Unique identification is NOT a technical challenge (U.S. SS# - 1935)
- Strength in numbers – at this point you would need a very good reason to start yet another PID scheme
  - Smaller independent schemes will be more fragile and vulnerable to a small group moving on in any fashion, i.e., less persistent
  - Reliable well-run systems will tend to grow (nobody gets fired for assigning DOIs?)
  - If there is some aspect of a current widely used scheme that doesn't work for your case, talk to that community
- What problem are you trying to solve?
  - Don't start with deciding on a scheme, start with defining the requirements
- Resolution Systems – basic decision point
  - Single authoritative resolution system (≠ single point of failure): DNS, Handle
  - No single authoritative system (but controlled minting): ISBN, SS#

Requirements: Identifier String

- Not based on any changeable attributes of the entity
  - Location
  - Ownership
  - Any other attribute that may change w/o changing data itself
- Opaque, preferably a ‘dumb number’
  - A well known pattern invites assumptions that may be misleading
  - Meaningful semantics invite IP wars, language problems
- Unique
  - Avoid collisions, referential uncertainty
- Nice to have
  - Human-readable
  - Cut-able, paste-able, embeddable
  - Fits common systems, e.g., URI specification
- All of the above contribute to persistence

Requirements: Identifier Resolution System

- Reliable
  - Redundant, no single points of failure
  - Fast enough to not appear broken
- Scalable
  - Higher loads managed with more computers, not new software
- Flexible
  - Adapt to changing computing environments
  - Useful to new applications
- Trusted
  - Resolution/Administration must be trusted
  - Organization must be committed to the long term
- Open Architecture
  - Leverage efforts of a community in building apps on your infrastructure
- Transparent
  - Users knowing the id/infrastructure NOT a good feature
- Persistence, again

Using a Resolution System with Existing Identifiers

- No lack of identifiers in the world
- ISBN mapped to DOI
  - Example: 10.97812345/99990
  - The syntax specification, reading from left to right, is:
    - Handle System DOI name prefix = "10."
    - ISBN (GS1) Bookland prefix = "978." or "979."
    - ISBN Publisher prefix = variable length numeric string of 2 to 8 digits
    - Prefix/suffix divider = "/"
    - ISBN Title enumerator and checkdigit = variable length numeric string of 8 to 2 digits

Persistence is (primarily) an Organizational Issue

- No technology runs itself
- Organizations need to commit to persistence
- Organizations need the resources to keep their commitments
  - Size helps
  - Business model needed (profits not required, but funds are)
- Organizations need dedication to persistence
  - Conflicts of interest, e.g., if profit is the motive (not the case in any major system of which I am aware) then lack of profit will be a problem
- Regional organizations will have difficulty growing
  - International organization is best, even with the accompanying political and cultural issues

Persistence is an Organizational Issue (but don't make it harder than it needs to be)

- Do not bake changeable attributes into the string, such that users and developers operate with mistaken assumptions
  - Ownership if ownership can change
  - Organizational names (count the orgs that are > 100 years old and still have the same name)
- Assume resolution will change over time
  - Usage can and will change
  - Computing/networking environments that seem eternal will change
  - Indirection is a good thing
    - ID string will exist as a static set of bits in various formats while the computing and usage environments shift
    - Disconnect the string from those things that will change
    - New functions will evolve over time – don't make it difficult to connect the ID to those new functions
    - Best example of the problem – broken URLs
    - An example of a solution – adding functionality to DOIs
      - Going from 1-to-1 to 1-to-Many
      - Adding linked data as a resolution option

Final Word - PID Advantages

- Persistent Identity via Indirection
  - Static references into fluid systems over time
    - Data on networks moves
    - Ownership/responsibility change
    - Formats change
  - Embedded Ids
    - For data object in hand – current state data
      - Updates
      - New related entities
  - Networks of Persistent Links
    - Data / metadata links
    - Provenance chains
    - Inheritance across a broad set of entities

Final Word - PID Disadvantages

- Extra level of effort / cost on creation
  - Analysis – what to identify / granularity
  - Coordination across organizations
  - Maintain resolution system
- Persistence requires sustained effort
  - Organizational discipline
  - Technology necessary but not sufficient
- Analyze cost/benefit ratio
  - Don’t start unless its worthwhile
  - Is your data worth it?