Introduction to THOUGHTS on Persistent Resolution

Herbert Van de Sompel
Research Library
Los Alamos National Laboratory, USA
Acknowledgments

- Jeroen Bekaert at Ghent University & the Los Alamos National Laboratory
- Carl Lagoze, Sandy Payette, Simeon Warner at Cornell University
- Xiaoming Liu at the Los Alamos National Laboratory
Context

- An environment consisting of Digital Object Repositories with a Long Life Expectation:
  - Scholarly repositories
    - Institutional repositories
    - Discipline-oriented repositories
    - Publisher’s repositories
    - Dataset repositories
    - ...
  - Cultural heritage repositories
  - Preservation archives
  - Educational repositories
Context

• This Long Life Expectation comes with requirements regarding persistence that:
  o are different from those for the overall Web environment
  o go beyond a single generation of technical implementations

• We understand and accept this regarding digital preservation of Digital Objects
• We understand and accept this regarding persistence of identification of Digital Objects
• It also applies to resolution of identifiers of Digital Objects

=> Persistent Resolution
Persistent Resolution

- Need for an Identifier resolution system that we can carry into the future
  - Identifier resolution system needs to be able to deal with all kinds of current and future identifiers:
    - Various namespaces, actionable, non-actionable, resolvable, non-resolvable, …
  - Identifier resolution system must be deployable on the basis of current and future technologies
- Persistent resolution means *something* needs to come back in response to a resolution request

=> The Persistent Resolution environment proposed here is supposed to exists in parallel to resolution mechanisms for those existing identifier schemes that have built-in resolution.
THOUGHT 1: Persistence is a matter of policy

- Persistence of identification of a Digital Object is not guaranteed by picking a specific technology; it can be achieved on the basis of several identification systems (http, purl, urn, ark, info, …)
- Persistence of identification of a Digital Object is a matter of policy of the custodian of a Digital Object ~ policy of the Repository
- Persistence of identification Of a Digital Object is kind of a hollow concept without an associated persistence of resolution of that identifier into something

⇒ Make the Repository responsible for persistent resolution of the identifier into that something
⇒ Make the Repository express the commitment to persistence of identification/resolution
⇒ Repository centric resolution environment
THOUGHT 2: Identifier Resolution to *something*

- Resolution to the identified Digital Object does not make sense
- Resolution to a *Surrogate* for the identified Digital Object
- Commitment to persistence of identification/resolution by a Repository means commitment to bringing a *Surrogate* back
THOUGHT 2 (CONT'D.): Identifier Resolution to *something*

- **Surrogate:**
  - A representation of a Digital Object
  - Expresses properties and access points for the Digital Object
  - Uniform across the repositories: not tied to identifier-type & not tied to specific application domain
  - Would be great if it were not encumbered by IP issues
  - *Surrogate* expresses (level of) commitment to persistence
  - *Surrogate* contains the necessary information (providerInfo) to get a *Surrogate* back at a later point in time
  - the *Surrogate* that is returned at a later point in time may very well be very different

- **Long-term perspective:**
  - need an abstract Data Model for the representation of Digital Objects across Repositories
  - Data Model can be serialized into different *Surrogate* formats (all compliant with the Data Model) as technologies evolve
Candidate technology: Pathways Core Data Model for Surrogates

THOUGHT 3: Where is that resolution interface?

- Given an identifier, where can it be resolved into a *Surrogate*?
- Introduce providerInfo (part of the *Surrogate*)

<table>
<thead>
<tr>
<th>identifier</th>
<th>version</th>
<th>location of resolution interface</th>
</tr>
</thead>
</table>

- Long-term perspective: need indirection, i.e. need Registry of “identifiers of provider” listing actual locations of resolution interfaces

<table>
<thead>
<tr>
<th>identifier</th>
<th>version</th>
<th>identifier of provider</th>
</tr>
</thead>
</table>

- About providerInfo:
  - An *identifier* for Persistent Resolution purposes
  - Long term machine actionable citation
  - Variation on the theme “Digital Objects carry their own identifiers” => “Surrogates carry their own providerInfo == the way in which to obtain a(nother) *Surrogate* over time”
Thought 4: Resolution protocol

- Need abstract definition of identifier resolution protocol
  - Instantiate abstract protocol using different technologies as time goes by

```
from providerInfo

Obtain

identifier, version

service identifier, (request Surrogate)

Surrogate

contains providerInfo
```
Candidate technology: OpenURL Framework Standard


  An OpenURL Application is a networked service environment in which packages of information are transported over the network. These descriptions have a description of a referenced resource at their core, and they are transported with the intent of obtaining context-sensitive services pertaining to the referenced resource.
Candidate technology: OpenURL Framework Standard

Repository
Obtain interface

Resolver

Referent: identifier, version
ServiceType: request Surrogate

Transport

description of
Referent & context

ContextObject

networked resource
reference

about
Referent

Digital Object

Surrogate

services pertaining to Referent
**SUMMARY**: Abstract Persistent Resolution protocol

- **Service Registry**
  - Provider
  - Location provider Obtain interface

- **Obtain**
  - Identifier, version
  - Service identifier, (request Surrogate)
  - Surrogate

- **providerInfo**:
  - Provider
  - Identifier
  - Version

- **OpenURL ContextObject**
- **OpenURL Transport**
- **Data Model**
- **Abstractions**
SUMMARY: Concrete instantiation

http://my.repository.org/obtain?
url_ver = z39.88-2004 &
rft_id = info:doi/123.112454 &
svc_id = info:pathways/svc/pwc.rdf

providerInfo:
- provider
- identifier
- version

KEV
ContextObject

HTTP Transport

RDF/XML
Pathways Core
Surrogate

Concrete
QUESTIONS, COMMENTS, FLAMES