Making groups a first-class abstraction of PIDs

Daniel S. Katz
Assistant Director for Scientific Software & Applications, NCSA
Research Associate Professor, CS
Research Associate Professor, ECE
Research Associate Professor, iSchool
dskatz@illinois.edu, d.katz@ieee.org, @danielskatz
What is a PID?

- Handles (IETF RFCs 3650, 3651, 3652) have prefix (naming authority) and suffix (local name)
  - Prefixes registered in Global Handle Registry
  - Opaque: encodes no information about underlying resource; provides only means to retrieve metadata about the resource
  - Information about resource (e.g., access protocol, host name, port, etc. for server holding the resource) are in metadata
  - Doesn’t require DNS, but can use proxy servers (https://doi.org/NA/LN)
  - No required model for persistence of objects
- DOIs (ISO/DIS 26324, ANSI/NISO Z39.84-2005)
  - Handles with special rules, including persistence commitment
- ARKs
  - Explicitly separates naming authority into name mapping authority and name assigning authority, local name similar to handle (http://NMAH/ark:/NAAN/Name)
  - Is a URL; use domain names from NMA
  - Returns object; add ? to return metadata, add ?? to return commitment statement
- PURLs
  - URLs that point to other URLs
Common elements of PIDs

• Naming authority & local ID split
• “Persistence” of IDs, not contents
• Formal process, fully run by professionals
• For objects that are intended to be static or “permanent”

• But what about objects that change?
  • Such as software – open source, with frequent commits
  • Authors working on a project – with frequent additions
  • Organizations – with frequent hiring and moves
  • Data – with new data frequently added

• All involve the idea of a group
Software citation today

- Software and other digital resources currently appear in publications in very inconsistent ways
- Howison: random sample of 90 articles in the biology literature -> 7 different ways that software was mentioned

<table>
<thead>
<tr>
<th>Mention Type</th>
<th>Count (n=286)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cite to publication</td>
<td>105</td>
<td>37%</td>
</tr>
<tr>
<td>Cite to users manual</td>
<td>6</td>
<td>2%</td>
</tr>
<tr>
<td>Cite to name or website</td>
<td>15</td>
<td>5%</td>
</tr>
<tr>
<td>Instrument-like</td>
<td>53</td>
<td>19%</td>
</tr>
<tr>
<td>URL in text</td>
<td>13</td>
<td>5%</td>
</tr>
<tr>
<td>In-text name only</td>
<td>90</td>
<td>31%</td>
</tr>
<tr>
<td>Not even name</td>
<td>4</td>
<td>1%</td>
</tr>
</tbody>
</table>

- Studies on data and facility citation -> similar results
Software citation principles

• FORCE11 Software Citation group (2015-2017)
• ~55 members (researchers, developers, publishers, repositories, librarians)
• Reviewed existing community practices & developed use cases
• Drafted software citation principles document
  • Started with data citation principles, updated based on software use cases and related work, updated based working group discussions, community feedback and review of draft, workshop at FORCE2016 in April
  • Discussion via GitHub issues, changes tracked
• Content
  • 6 principles: Importance, Credit and Attribution, Unique Identification, Persistence, Accessibility, Specificity
  • Motivation, summary of use cases, related work, and discussion (including recommendations)
• Published version
  • Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group.(2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86 and https://www.force11.org/software-citation-principles
Discussion: Unique identification

• Recommend DOIs for identification of published software

• However, identifier can point to
  1. a specific version of a piece of software
  2. the piece of software (all versions of the software)
  3. the latest version of a piece of software

• One piece of software may have identifiers of all 3 types
• And maybe 1+ software papers, each with identifiers

• Use cases:
  • Cite a specific version
  • Cite the software in general
  • Link multiple releases together, to understand all citations
Discussion: Unique identification (cont.)

• Principles intended to apply at all levels
• To all identifiers types, e.g., DOIs, RRIDs, ARKS, etc.
• Recommend when possible use DOIs that identify specific versions of source code
Discussion: Identifier resolves to ...

- Identifier that points directly to software (e.g., GitHub repo) satisfies Unique Identification (3), Accessibility (5), and Specificity (6), but not Persistence (4)
- Recommend that **identifier should resolve to persistent landing page that contains metadata and link to the software itself, rather than directly to source code**
- Ensures longevity of software metadata, even beyond software lifespan
- Point to figshare, Zenodo, etc., not GitHub
Problem: software citation vs paper citation

• Three relevant steps for paper citation
  1. Creator (aka author) submits paper to “publisher”
  2. [review+], then publisher publishes paper & assigns identifier, often DOI
  3. To refer to paper within another work, cite paper metadata, often including DOI

• Fixed order, discrete steps

• For software today
  • Creator develops software on GitHub, released at different stages (versions) during its development
  • Someone who uses that software will likely not cite it, but if they do, they will cite the repository
  • No step 2
  • Partial step 3, because there is no clear metadata or identifier for the software that was used

• Software citation principles inserts step 2
Software Heritage as a partial solution?

• Mission: to collect, preserve, and share all software that is publicly available in source code form
• Source code stored in a Merkle tree, with hashed pointers

• To cite unpublished software, need to find hash to version used
  • Tool to go from GitHub commit to SH hash could be developed

• Need ID for SH hash
  • Maybe https://softwareheritage/id
How to refer to versions of software

• With Software Heritage and CodeMeta as “standards” (building blocks), have all the needed data
• But don’t have tools/methods to use it
• Use cases:
  • Cite a specific version ✔
  • Cite the software in general ?
  • Link multiple releases together, to understand all citations ?
Use RelationType?

  - isNewVersionOf
  - IsPreviousVersionOf
  - IsPartOf
  - HasPart

- Added in https://schema.datacite.org/meta/kernel-4.1/
  - HasVersion
  - IsVersionOf
RAiD (Research Activity Identifier)

- RAiD (https://www.raid.org.au): an identifier for research projects & activities
- A RAiD
  - A handle (string of numbers), minted via the RAiD API
  - Persistent, can have other Identifiers associated; e.g., ORCIDs & DOIs
  - Can be integrated into data management records of any type to trace all researchers, institutions, outputs, tools & services
  - Suitable for any kind of Research Activity
- Service Points
  - Define activities or projects as needed, e.g., long term projects, short term projects, sub-projects, single experiments
  - Mint RAiDs
  - Update and manage RAiDs that they mint
- More in next session (noon) at “Stage 2”
Scholarly Commons WG3 is building a system for decision trees:
- Decision trees are objects with many formats (PDF, coded markup, etc.)
  - Somewhat similar to papers, software, data (everything is data)
- They are improved over time (multiple versions); authors can be added
- How to identify (and publish) them leads to problems
  - Which contributions – and by what entities – does the relevant community recognize?
  - Automatically associating projects with awards with people should be easy
- Current problems include:
  - RAiD as currently envisaged to be used by institutions to track their resources and outputs; no clear funder or project member functionality
  - We initially thought we could use https://share-my-id.orcid.org/ to help with this, since it allows a group to form, but it can only be used to collect ORCiDs; the collection can not be used as an object
You've been invited to add your ORCID iD to the collection:

The Scholarly Commons Subworking Group 3
Created by Fiona Murphy. http://orcid.org/0000-0003-1693-1240 on 2017-06-12

We’re working on a project to operationalise The Scholarly Commons - a set of research practices that are Open by Design, FAIR, and Citable. Part of this project involves publishing Decision Trees. We have decided that Persistent Identifiers are a critical element of ‘commons-ing’ so are collecting this group of IDs in order to have a persistent record of the authors of the project and its outputs.

Add my iD

Your iD and public name will be displayed below.

Don't have an ORCID iD? Register for an iD and add it to this collection at the same time: Click 'Add my iD' above, then 'Register now' on the next screen.
Not familiar with ORCID? Learn more

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>ORCID ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fiona Murphy</td>
<td><a href="http://orcid.org/0000-0003-1693-1240">http://orcid.org/0000-0003-1693-1240</a></td>
</tr>
<tr>
<td>2</td>
<td>Daniel S. Katz</td>
<td><a href="http://orcid.org/0000-0001-5934-7525">http://orcid.org/0000-0001-5934-7525</a></td>
</tr>
<tr>
<td>3</td>
<td>Ian Bruno</td>
<td><a href="http://orcid.org/0000-0003-4901-9936">http://orcid.org/0000-0003-4901-9936</a></td>
</tr>
<tr>
<td>4</td>
<td>Daniel Mitchen</td>
<td><a href="http://orcid.org/0000-0001-9488-1870">http://orcid.org/0000-0001-9488-1870</a></td>
</tr>
<tr>
<td>5</td>
<td>K. Jane Burpee</td>
<td><a href="http://orcid.org/0000-0002-4847-071X">http://orcid.org/0000-0002-4847-071X</a></td>
</tr>
<tr>
<td>6</td>
<td>Michael Bar-Sinai</td>
<td><a href="http://orcid.org/0000-0002-0153-8465">http://orcid.org/0000-0002-0153-8465</a></td>
</tr>
<tr>
<td>7</td>
<td>Danny Kingsley</td>
<td><a href="http://orcid.org/0000-0003-3636-5939">http://orcid.org/0000-0003-3636-5939</a></td>
</tr>
<tr>
<td>8</td>
<td>Martyn Rittman</td>
<td><a href="http://orcid.org/0000-0001-9327-3734">http://orcid.org/0000-0001-9327-3734</a></td>
</tr>
<tr>
<td>9</td>
<td>Maryann Elizabeth Martone</td>
<td><a href="http://orcid.org/0000-0002-8406-3871">http://orcid.org/0000-0002-8406-3871</a></td>
</tr>
</tbody>
</table>
We recognize concerns about how groups should be managed
  • Ethically; who decides who has made what contribution; what is the bar to entry to a particular group; etc.
  • However, we’ve come to: “the community” or “communities” need to be able to decide where the credit goes: to the individual, working group, or to project itself
  • Researchers should be able to do the “right” thing via infrastructures that facilitate them
  • Platforms that allow groups to form also should provide a means to associate IDs, provide a stable identify for the group, and provide an API so that groups can assert authorship
  • If a group changes regularly, the group itself needs to be versioned, just like any other scholarly object

Credits:
SC WG3 Blog3: https://www.force11.org/blog/decision-trees-licenses-attribution-provenance-credit-and-glitches
Assign ORCIDs to sets of co-authors

I work with an institutional repository now depositing a paper with more than 300 co-authors. We're also talking about what we'd do with the paper reported in Nature News with more than 5,000 co-authors.

http://www.nature.com/news/physics-paper-sets-record-with-more-than-5-000-authors-1.17567

Here's a thought. Imagine that ORCID allowed a researcher to create a set of co-authors for a given paper by clicking ORCID checkboxes, roughly as Gmail users can create an email list by clicking individual Gmail contacts. The resulting set of authors would have a single ORCID. It might be used only once, or it might be used many times, depending upon how many times the same set of people co-author a paper. Any reader clicking on the ORCID profile would see complete metadata.

Hi,

Thanks for the suggestion, but the ORCID Registry is designed specifically for assigning identifiers to individual researchers. The Registry's functionality, our terms of use, and the basic design of how ORCID iDs are link to publications and other research outputs are all based on a single individual person associated with the ORCID iD. Redesigning everything to allow for iDs assigned to a group isn't feasible for us. While we agree that a solution to identify groups of authors is needed, that solution isn't ORCID identifiers.

Best,
-Catalina
ORCID Support
Data concerns

• Data collections and data sets are really groups of data elements or data streams
• Data sets can change (as new measurements are made or collected)
• Data streams are also groups of data elements
• Real issue: data is plural
Overall Problem

• Groups are hierarchies: group name / \{objects\}
• PIDS are single level: naming authority / object

Use cases:
• Distribution in organization
  • Groups is local (to a naming authority)
  • Group is distributed (beyond a naming authority)
• Distribution in time
  • Group is static – created at the same time as the objects in it
  • Group is dynamic – group is created, then objects are inserted or removed (beyond a time)
Idea 1 for group PIDs

• As done by figshare and others for versioning, add an extra level to PIDs
  • Naming authority/local group/local item
  • E.g.: Figshare adds .vN to the end of DOIs
    • https://doi.org/10.6084/m9.figshare.5328442.v3
    • Base DOI points to latest version
    • see https://figshare.com/blog/The_future_of_figshare/166
• Probably a bad idea to put semantic info in the DOI
• Doesn’t work when the distribution is beyond the naming authority (figshare)
Idea 2 for group PIDs

• Continue to use relationType
  • As done by Zenodo for versions
    • Unique DOIs for each version and for concept, with no semantic content
    • Concept DOI points to latest version or content landing page
    • See http://help.zenodo.org/#versioning

• Need to update indexes and related tools

• To support use cases
  • A specific version
  • The project/concept
  • The latest version
Idea 3 for group PIDs

- Provide split between naming authority (NA) and a more local authority (equivalent to lead author, let’s call LA)
- Naming authority sets up group and assigns LA
- LA can make changes to group
- Changes involving people may (need to) be verified by those people
Ideas 2 and 3 need support

• How to implement?
  • Assigning a local authority
    • Person who creates the first or concept DOI?
    • Standard of using RelationType IsCollection for concept DOI? Or need another RelationType?
  • How to verify/ensure changes are correct
    • Before acceptance?
    • After acceptance?
  • API for this, with naming authority
    • One or many?

• What are the shortcomings?
Credits

- Thanks to Arfon Smith and Kyle Neimeyer for co-leadership in FORCE11 Software Citation WG
- And Martin Fenner for help understanding identifier systems and metadata
- And Fiona Murphy and Maryann Martone in FORCE11 Scholarly Commons Working Group 3
- More of my thinking
  - Blog: http://danielskatzblog.wordpress.com
  - Tweets: @danielskatz