Practical Challenges in Designing Digital Object Identifiers for Data

Adam Clark, Tim Ahern, Chad Trabant, Robert Newman, Rick Benson

IRIS Data Management Center, Seattle, Washington

IN13B-1563

Abstract

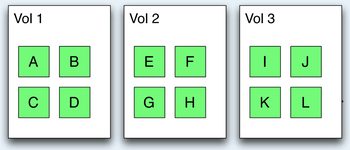
While the basic idea of Digital Object Identifiers (DOIs) is fairly simple, a robust implementation must consider a number of factors that are not necessarily obvious at first glance. This is especially true when attempting to apply the DOI framework -- which was designed for use with published works -- to datasets for which concepts like identity, provenance, and lifecycle may be much more complex or ambiguous.

The IRIS Data Management Center (DMC) recently undertook an effort to build DOIs for various parts of its data holdings, and we share some of the particular challenges and solutions that emerged from this process.

The DOI system is designed for simple identifiers

The DOI system was designed around the model of a published journal, in which a publisher is responsible for the dissemination of a relatively small number of discrete articles. The challenges of designing a DOI system for data generally stem from the mismatch between the characteristics of a given dataset and this basic design.

The intended model looks something like this:



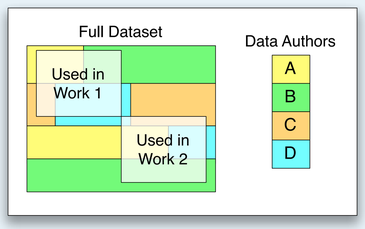
- Although articles may be split across volumes, they are easy to consider as a single list.
- Each article is a discrete entity -- there are clear boundaries between articles, and each can be easily treated in isolation.

DOIs serve multiple (potentially conflicting) roles

Citations serve a number of purposes which tend to be **conflated**, but good DOI design requires **recognition that there are a few different parties who may have diverging interests**:

- Data Authors (whose work is being cited) *want DOIs that reflect their contributions to a dataset.*
- Researchers (who add citations to their work) and **Readers** (who view those citations) *want DOIs that reflect the data that was used in a particular work.*

Since the patterns of data usage **may not correlate** with the patterns of authorship, DOI design **may have to choose which group's interest it prioritizes.**

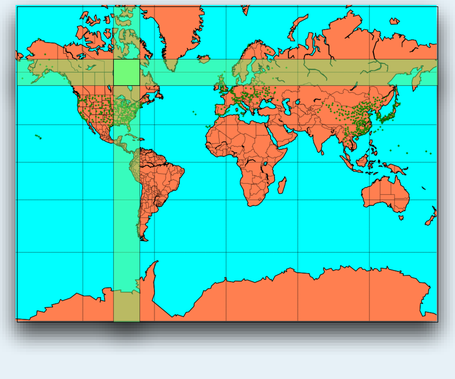


Data usage often **does not** align with data authorship.

Identifiers tied to authorship would provide credit *but may not provide useful information about the data that was actually used*.

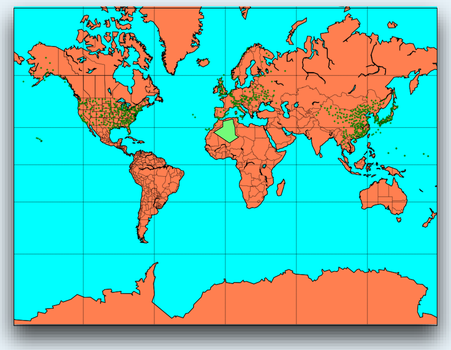
Example: How to identify a location?

This is a simple example of one fundamental challenge of designing identifiers for data. Locations are **multidimensional** (they can be defined by two physical dimensions, or by administrative, geographical, or other boundaries) and those dimensions may be **continuous** (latitude and longitude have no natural units, so any subdivisions are necessarily arbitrary).



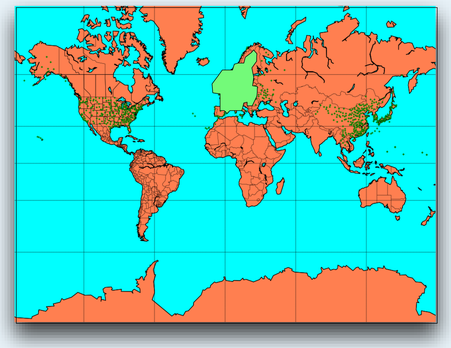
"53N112W"

Identify by **latitude** and/or **longitude**. Granularity is arbitrary, and higher granularity generates exponentially more identifiers:
- 1°x1° units: 65,000 identifiers
- 1'x1' units: 233 million identifiers



"ALGERIA"

Identify by **administrative boundaries**. These are not stable or consistent.



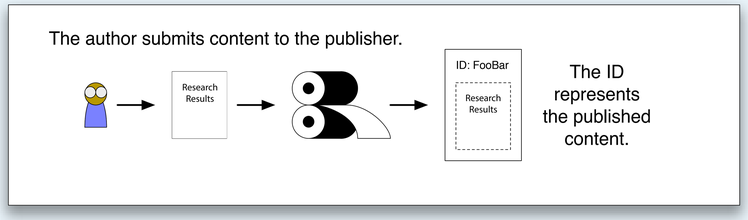
"36: NORTHWESTERN EUROPE"

Identify by **Flinn-Engdahl region**. An existing domain-specific identification system may be a good solution, or at least suggest a practical approach.

Data centers are publishers but not owners

In the traditional publishing model, an author submits an article to a publisher, at which point the publisher is, if not the owner of the article's content, at least the owner of a **canonical presentation** of the article.

In this model, the DOI can be viewed as an identifier of the published version of the content, that is, it identifies something that belongs to the publisher.



A data center may be tasked with curating and managing data, but it does not have the same kind of ownership that a traditional publisher does. This can greatly complicate the task of managing DOIs:
- Any DOI assignment requires coordination with the original author.
- Authors of historical data may be difficult or impossible to reach.
- The data center may have little control over variations in scope and granularity in DOIs from different authors.

DOI Target URL Design

A DOI is commonly used as a **permanent URL**; the DOI record includes a target URL, and a DOI resolution service will redirect requests to that URL.

Ex: `http://dx.doi.org/10.1111/T12345`
The server at `dx.doi.org` (a DOI resolution service) looks up the target URL for doi:10.1111/T12345 and redirects to its registered target URL.

The target URL can be updated (unlike the DOI string itself, which is immutable), but a dataset may encompass millions of DOI entries, and batch updates can be difficult., so for maintainability the target URL should be designed as **semi-permanent**.

An easy way to do this is to register a generic proxy URL, which redirects to the actual URL at the target.

Actual URL -- exposes lots of implementation details:
`http://www.iris.edu/mda/network.php?net=IU`
name of the tool^ file extension^ ^method for passing data

Generic target -- this is much less likely to require changing later:
`http://www.iris.edu/doi/network/IU`

Using pattern matching, a webserver or proxy can easily redirect all requests from the generic target to the actual URL.

Apache redirect for the pattern above:
`RedirectMatch ^/doi/network/(.*)$ /mda/network.php?net=$1`