SWIB 2013

Tutorial

on

Metadata Provenance

Slides: http://bit.ly/swib13-provenance

Metadata Provenance

Part 1: Linked Data Provenance

*"How can we **identify** RDF data, statements within RDF data, Linked Data, ... in order to provide provenance?"*

Part 2: The PROV Ontology

*"How can we **represent** the provenance of resources?"*

Speakers

Part 1: Linked Data Provenance

Dr. Kai Eckert

Mannheim University
Data and Web Science Group

Part 2: The PROV Ontology

Prof. Magnus Pfeffer

Stuttgart Media University

Agenda

13:00 Introduction and Foundations

Introduction to Provenance and Metadata

RDF and RDFS (very short)

Metadata (RDF) Provenance (What is the problem?)

13:45 Identification of RDF data

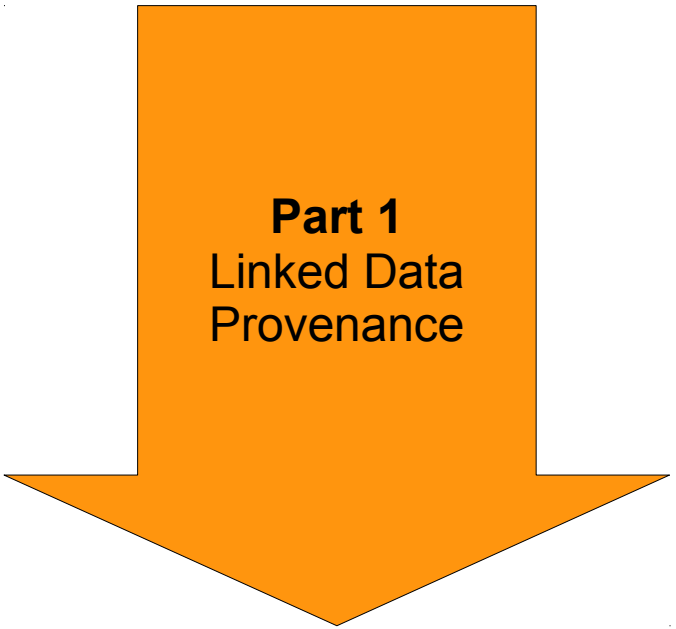
What's in the standards? A brief review of Reification.

Linked Metadata (Use the LD Principles)

Named Graphs

RDF 1.1

14:30 Short Break (15 min)



Agenda

14:45 Metamodels in Practice

- OAI-ORE
- The Europeana Data Model
- OAI-ORE "vs." Named Graphs
- Linked Data Publishing with VoID

15:30 Coffee Break 

16:00 Linked Data Publishing and Provenance

- State-ful or State-less Data, Versioning
- Identity and Provenance Context

Part 1
Linked Data
Provenance

Agenda

16:45 Modelling Provenance 1

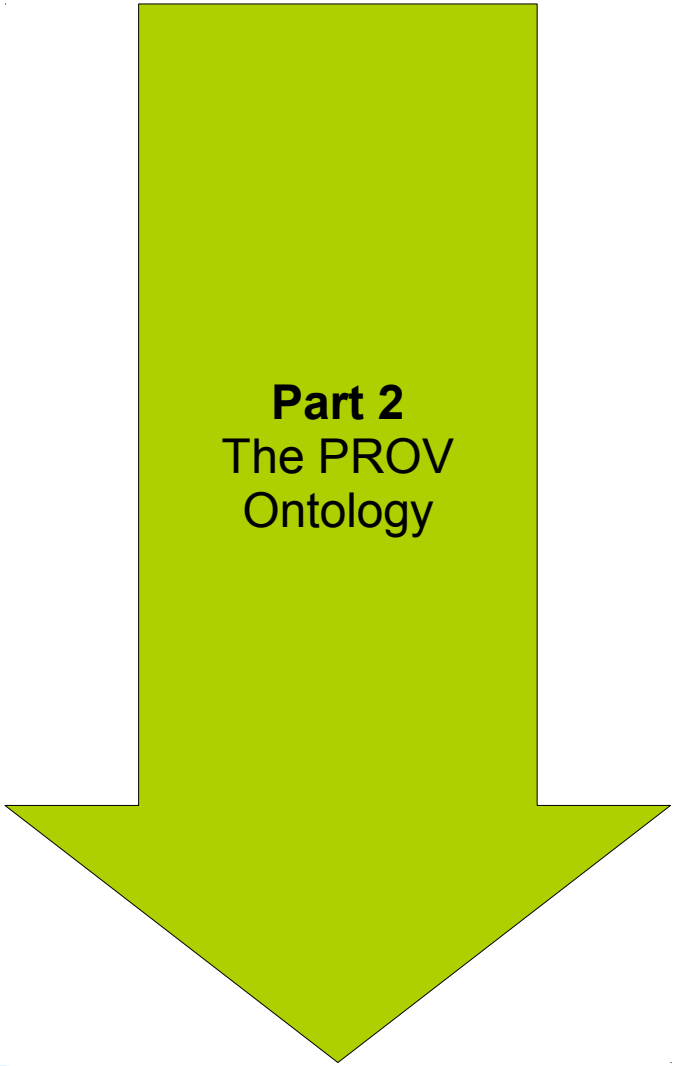
- A data model for provenance information
- Introducing the PROV ontology
- Extending the basic elements of PROV

17:30 Short Break

17:45 Modelling Provenance 2

- Qualifying relations in PROV
- Mapping DC provenance information to PROV

18:30 End



Slides, Further Readings

Eckert, Kai

Metadata Provenance in Europeana and the Semantic Web

Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft
Number 332, Berlin : Institut für Bibliotheks- und Informationswissenschaft der Humboldt-Universität zu Berlin, 2012, ISSN 14 38-76 62

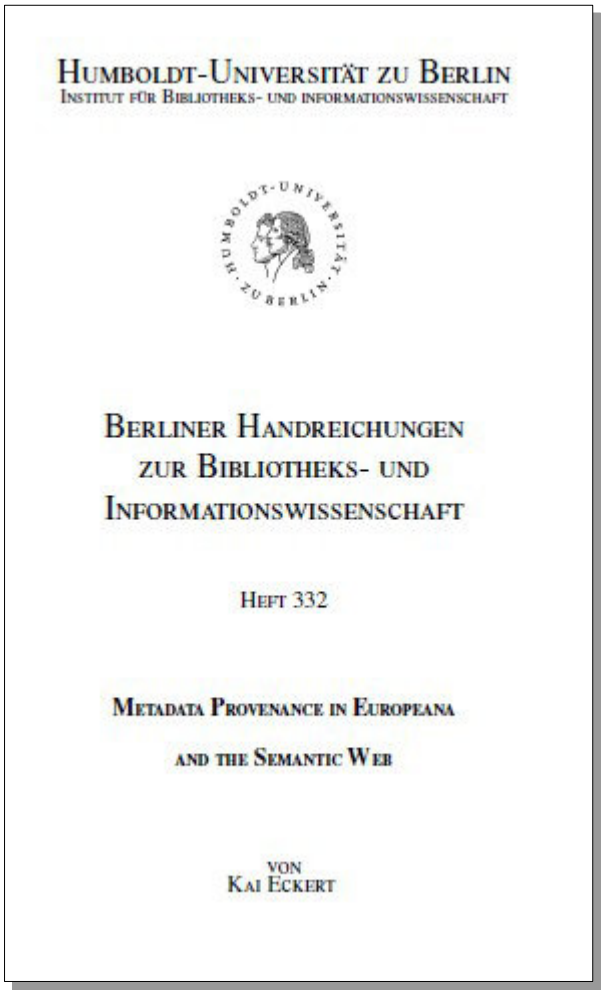
http://edoc.hu-berlin.de/series/berliner-handreichungen/2012-332

Eckert, Kai

Provenance and Annotations for Linked Data

Proceedings of the International Conference on Dublin Core and Metadata Applications 2013 (DC-2013), Lisbon, Portugal

http://dcevents.dublincore.org/IntConf/dc-2013/paper/view/154



http://bit.ly/swib13-provenance

Foundations

Agenda

Introduction to provenance and metadata

RDF and RDFS

Metadata (RDF) Provenance (What is the problem?)

Provenance

Not only ownership!
Not only artworks!

(But yes, my data is a valuable object.)

AN ENCYCLOPÆDIA
NICA COMPANY

erriam-
ebster

m-w.com

prov·e·nance *noun* \'präv-nən(t)s, 'prä-və-ˌnän(t)s\

Definition of PROVENANCE

1 : ORIGIN, SOURCE

2 : the history of ownership of a valued object or work of art or literature

See provenance defined for English-language learners »

Examples of PROVENANCE

- Has anyone traced the *provenances* of these paintings?
- The artifact is of unknown *provenance*.

Origin of PROVENANCE

French, from *provenir* to come forth, originate, from Latin *provenire,* from *pro-* forth + *venire* to come — more at PRO-, COME

First Known Use: 1785

Definition: Provenance

Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its *quality*, *reliability* or *trustworthiness*.

W3C Provenance Working Group (2013)

Metadata

AN ENCYCLOPÆDIA BRITANNICA COMPANY

Merriam-

Metadata is „About-Data“, not data about data...

metadata

Save Popularity

meta·da·ta *noun plural but singular or plural in construction* \-'dā-tə, -'da- *also* -'dä-\

Definition of METADATA

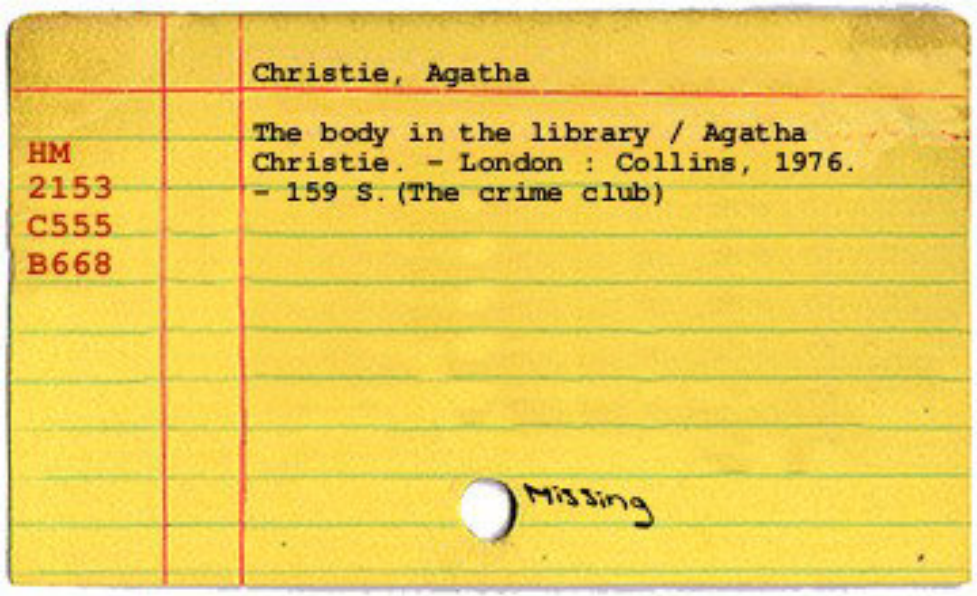
: data that provides information about other data

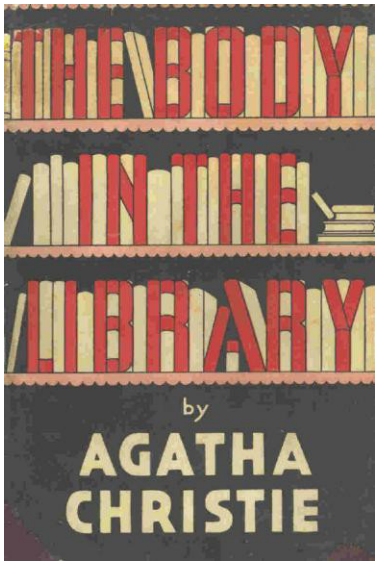
First Known Use of METADATA

1983

Definition: Metadata

Metadata is structured data that is used to describe the properties of a resource.



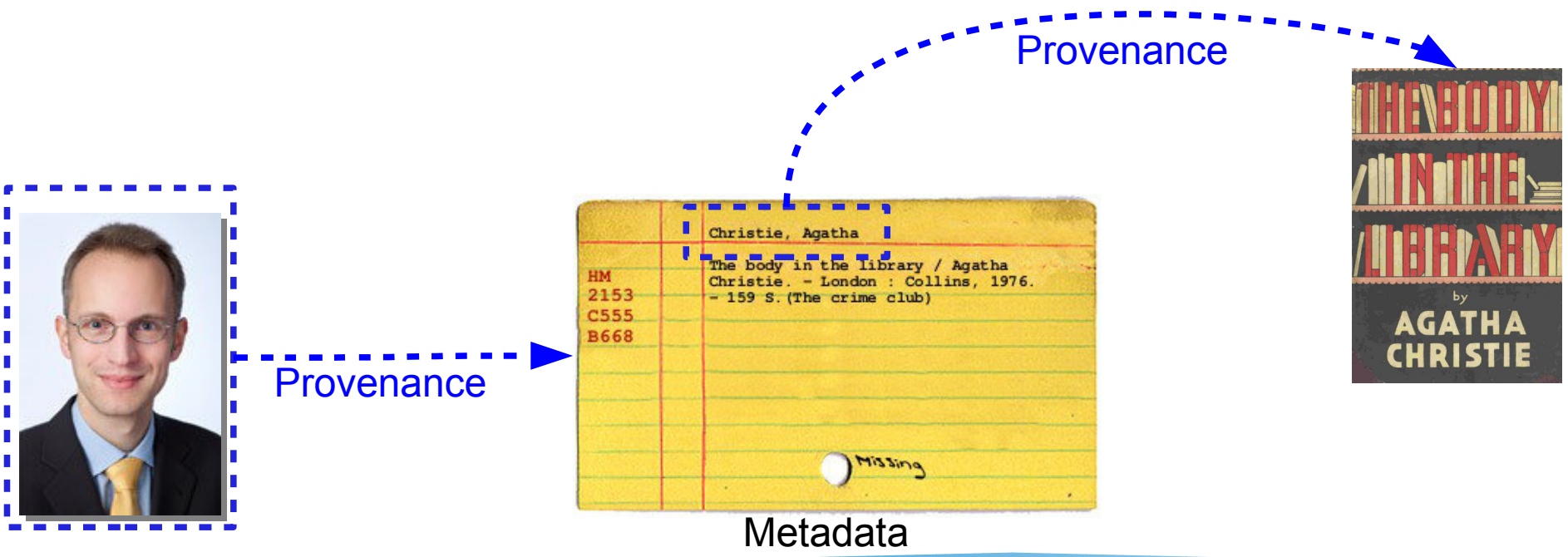


Metadata, Provenance and Metadata Provenance

Provenance data **is** metadata (Provenance metadata).

Metadata (typically) contains provenance information.

Metadata provenance is the provenance of metadata.



Resource Description Framework (RDF)

All things described by RDF are called ***resources***, and are instances of the class rdfs:Resource. This is the class of everything. All other classes are subclasses of this class.

Information about resources is expressed in ***statements*** about the resource.

A statement...

... is a **triple** of subject, predicate, and object,

... generally describes one **property** of one identifiable resource by assigning a value.

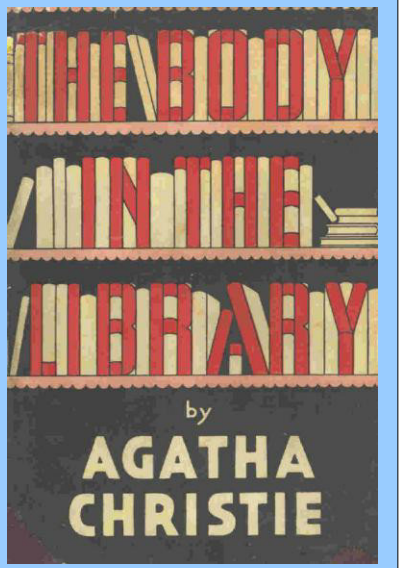
The **subject** is always a resource.

The **object** can be another resource or a literal.

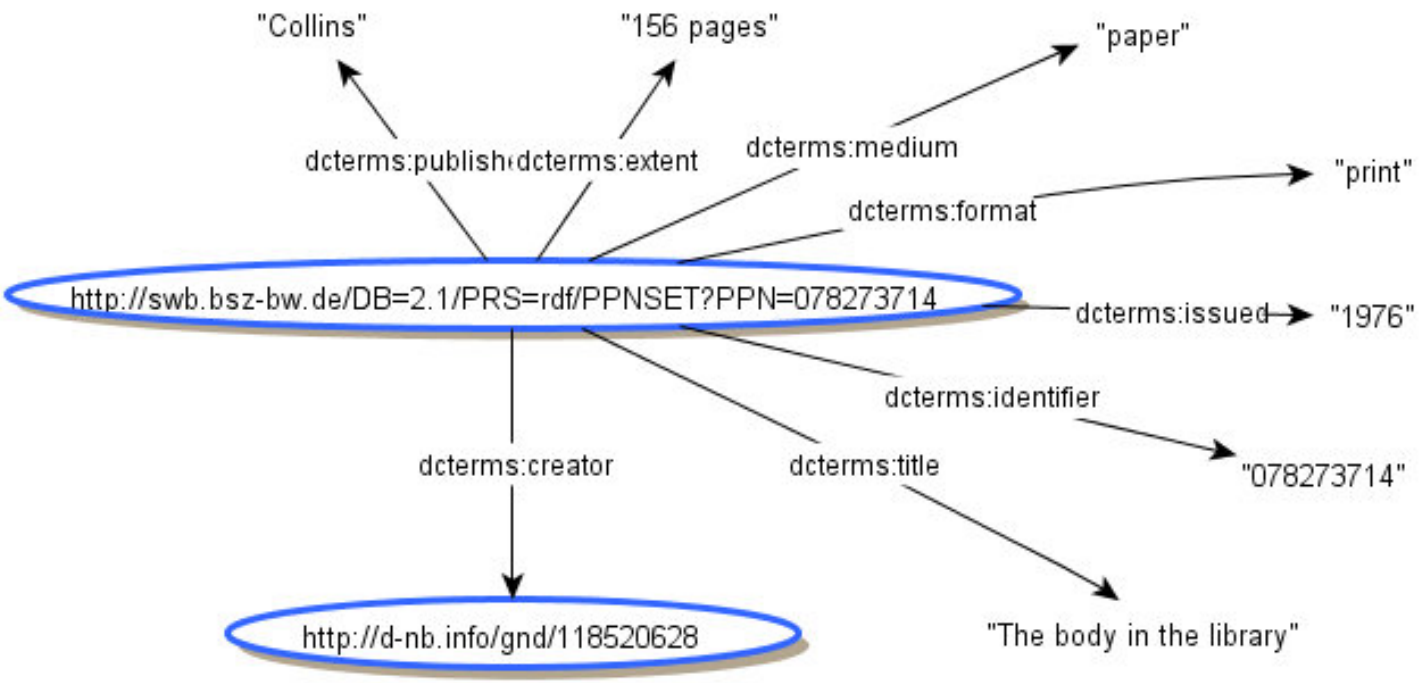
Example

```
@prefix dcterms: <http://purl.org/dc/terms/>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix swb: <http://swb.bsz-bw.de/DB=2.1/PRS=rdf/PPNSET?PPN=>

swb:078273714
    a rdf:resource ;
    dcterms:title “The body in the library”;
    dcterms:creator http://d-nb.info/gnd/118520628;
    dcterms:issued “1976”;
    dcterms:publisher “Collins”;
    dcterms:format “print”.
```



That's all folks!



RDF is a very simple and abstract graph-based model that supports links between resources and relations between resources and literals.

No graph boundaries, no records.

Yes, there are (named) graphs... we come to that.

Linked Data

Linked Data Principles:

1) Use URIs as names for things.

2) Use HTTP URIs so that people can look up those names.

3) When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).

4) Include links to other URIs, so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData.html

Linked Data

Information resources

Resources that are delivered via the Web:

Web pages, images, PDF files, ...

Non-information resources

Resources that are not on the Web:

Books, concepts, persons, ...

Linked Data

Dereferencing a URI from RDF data

- Non-information resources
  - Using http redirects (303 redirect)
    - Delivers information on the resource in RDF format

- Information resource
  - Depending on *content negotiation* and using http redirects
    - Delivers the resource itself
      - or
    - Delivers information on the resource in RDF format

Metadata in a linked data environment

Now metadata on a given resource...

... can come from many sources,

... can contain redundant statements,

... can contain false or contradictory statements,

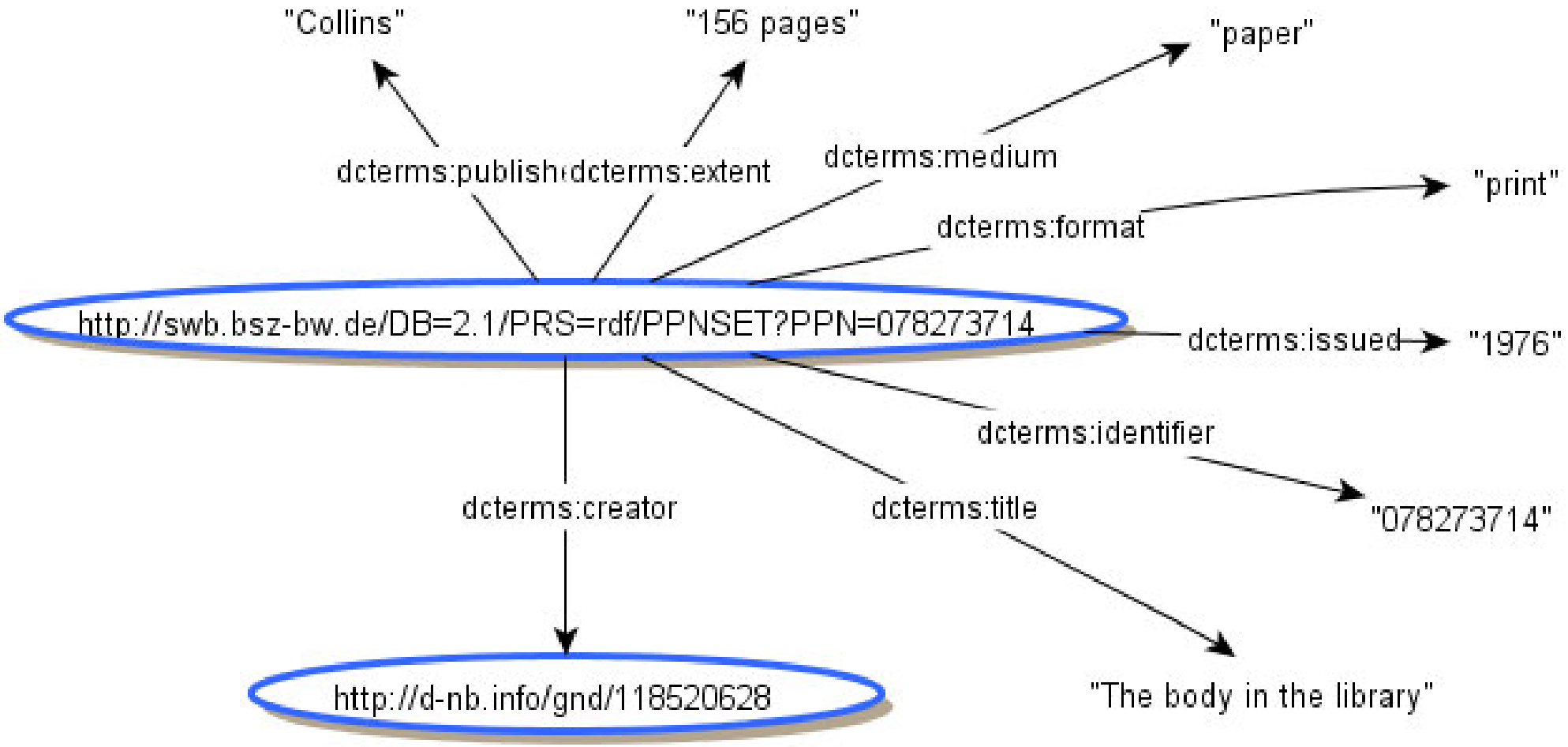
... can be created by many means and processes.

One would like to keep track of those statements

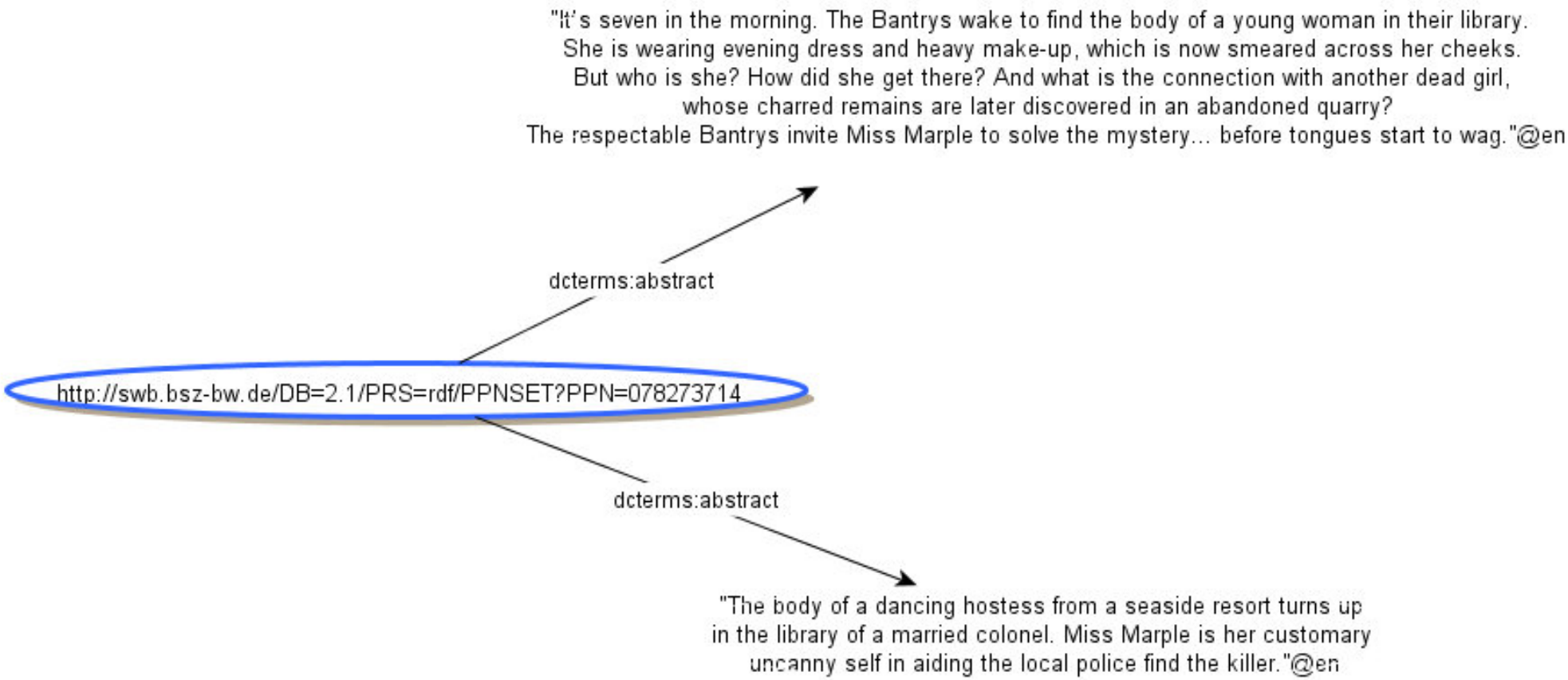
But provenance – as defined – only deals with resources.

Thus: We need a notion of metadata as a resource.

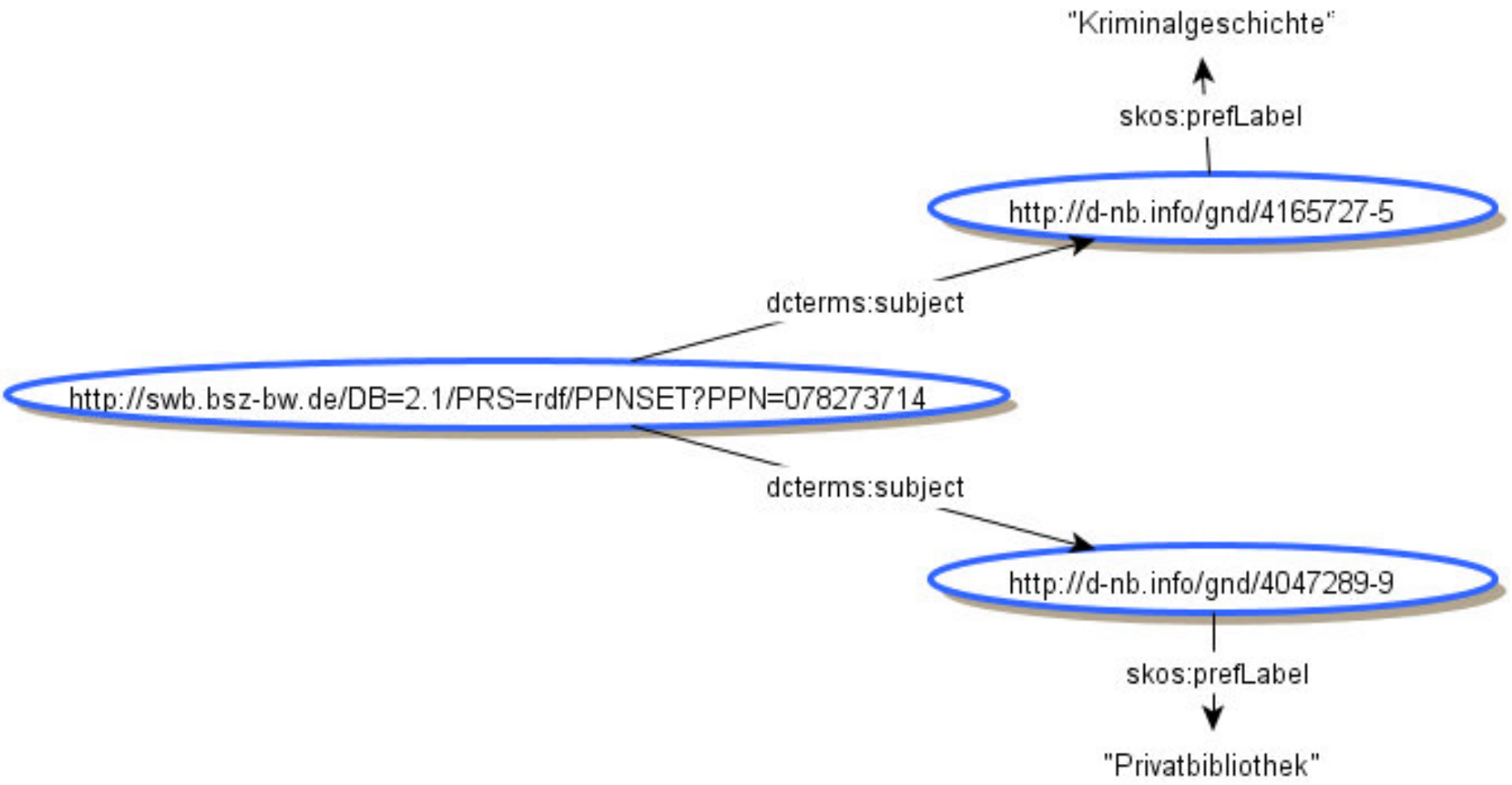
Example: Data enrichment



Add different abstracts



Add subject information



Metadata in a linked data environment

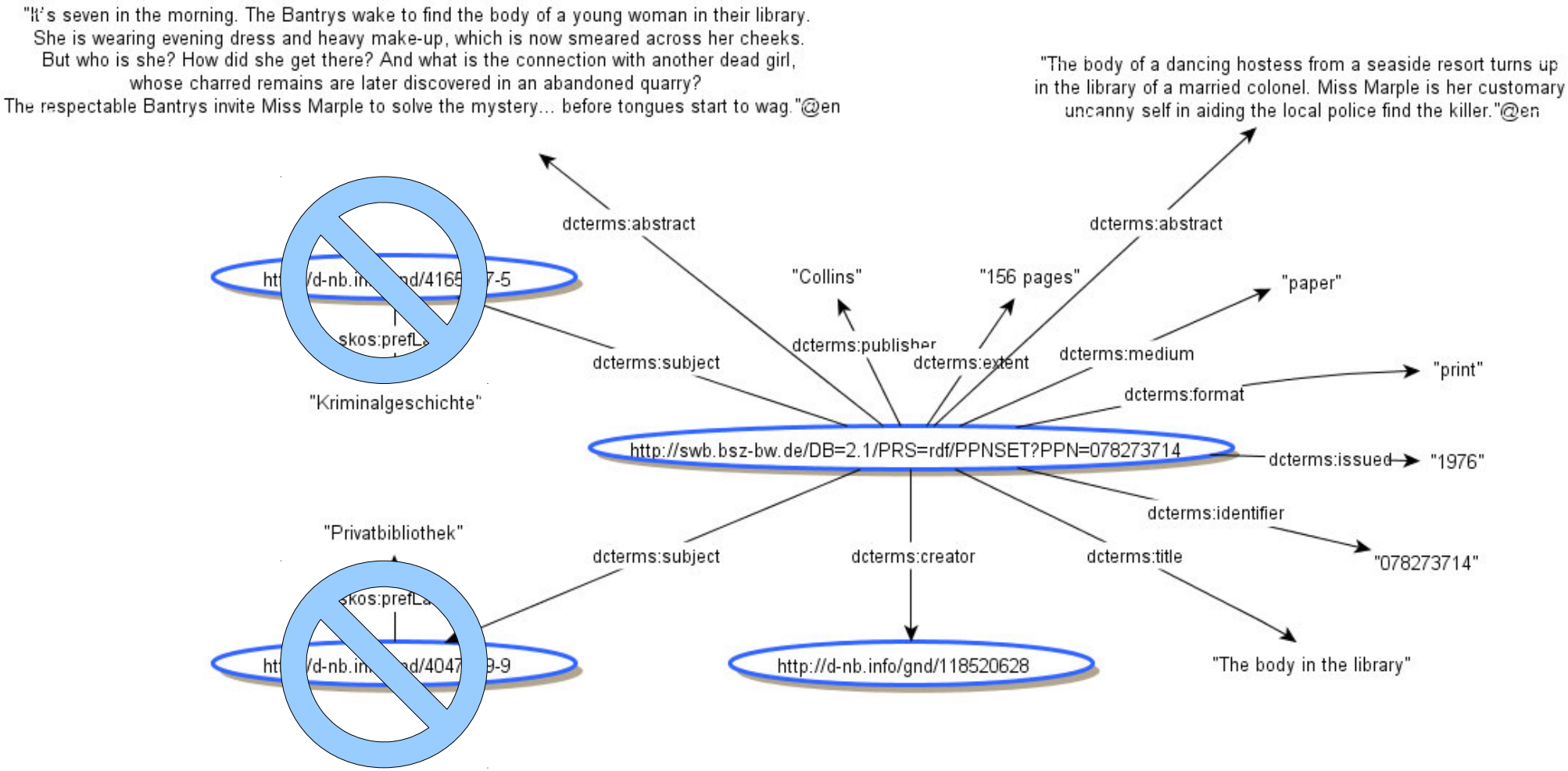
One would like to keep track of those statements

But provenance – as defined – only deals with resources. Is RDF data also a resource?

We need metadata provenance:

What dataset does a given statement belong to?
Who (or what) is responsible for it?

Example


"It's seven in the morning. The Bantrys wake to find the body of a young woman in their library. She is wearing evening dress and heavy make-up, which is now smeared across her cheeks. But who is she? How did she get there? And what is the connection with another dead girl, whose charred remains are later discovered in an abandoned quarry? The respectable Bantrys invite Miss Marple to solve the mystery... before tongues start to wag."@en
"The body of a dancing hostess from a seaside resort turns up in the library of a married colonel. Miss Marple is her customary uncanny self in aiding the local police find the killer."@en
dcterms:abstract
dcterms:abstract
"Collins"
"156 pages"
"paper"
dcterms:publisher
dcterms:extent
dcterms:medium
dcterms:subject
"print"
dcterms:format
"Kriminalgeschichte"
http://swb.bsz-bw.de/DB=2.1/PRS=rdf/PPNSET?PPN=078273714
dcterms:issued
"1976"
dcterms:identifier
"Privatbibliothek"
dcterms:subject
dcterms:creator
dcterms:title
"078273714"
http://d-nb.info/gnd/118520628
"The body in the library"

The Linked Data Gap

Linked Data publication is often one-way.

Linked Data as an export from the „real“ data.

Linked Data as a source for new data.

The connection easily gets lost!

Bridge the gap from YOUR data to Linked Data

Part 1: Linked Data Provenance

Identification of RDF Data

Metamodels in Practice

Linked Data Publishing and Provenance

Identification of RDF Data

Agenda

What's in the standards? A brief review of Reification.

Linked Metadata (Use the LD Principles)

Named Graphs

RDF 1.1

Expressing provenance in RDF

RDF offers a way to describe statements: Reification

- New resource to represent a statement
- Subject, predicate and object as properties of this resource
- Additional information using additional properties

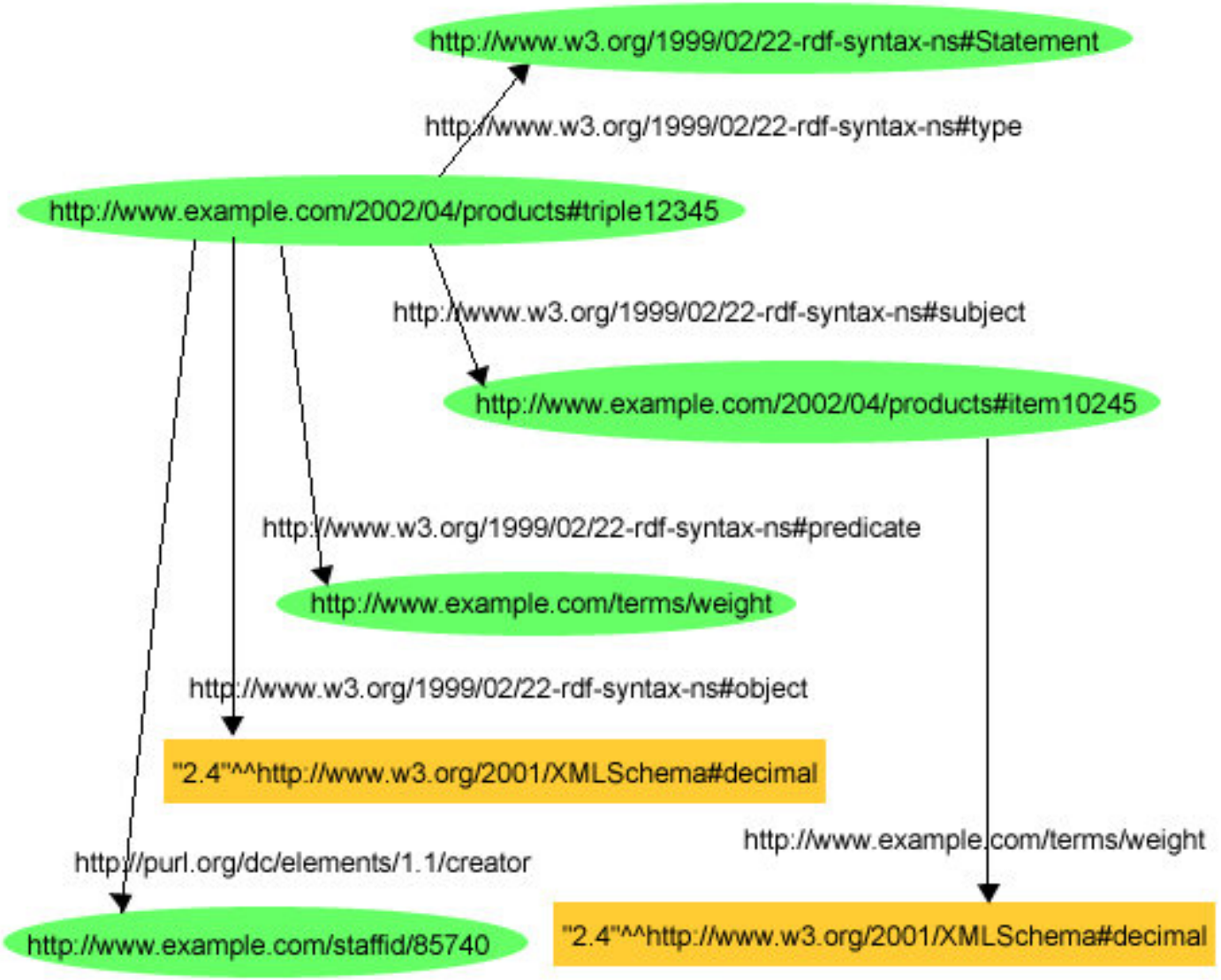
Example

```
exproducts:item10245    exterms:weight    "2.4"^^xsd:decimal .
```

```
exproducts:triple12345    rdf:type          rdf:Statement .
exproducts:triple12345    rdf:subject       exproducts:item10245 .
exproducts:triple12345    rdf:predicate     exterms:weight .
exproducts:triple12345    rdf:object        "2.4"^^xsd:decimal .
exproducts:triple12345    dc:creator        exstaff:85740 .
```

Source: RDF Core Working Group. (2004)

Example



Source: RDF Core Working Group. (2004)

Limits

No link between statement and reification:

- Only by matching subject, predicate, object.

No grouping possible:

- Excessive numbers of statements, e.g. identical creator for 100 statements leads to 500 additional statements.

Reification can be used to talk about specific statements (we'll come to this again, later), but is not practicable to provide the provenance of a whole dataset.

Linked Metadata

Linked Data Principles

1) Use URIs as names for things.

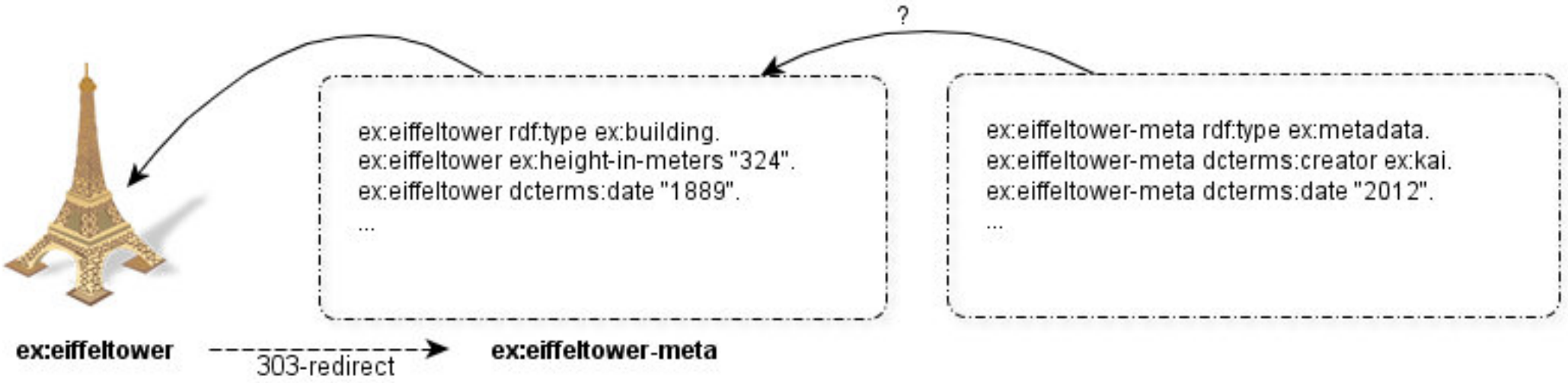
2) Use HTTP URIs so that people can look up those names.

3) When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).

4) Include links to other URIs, so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData.html

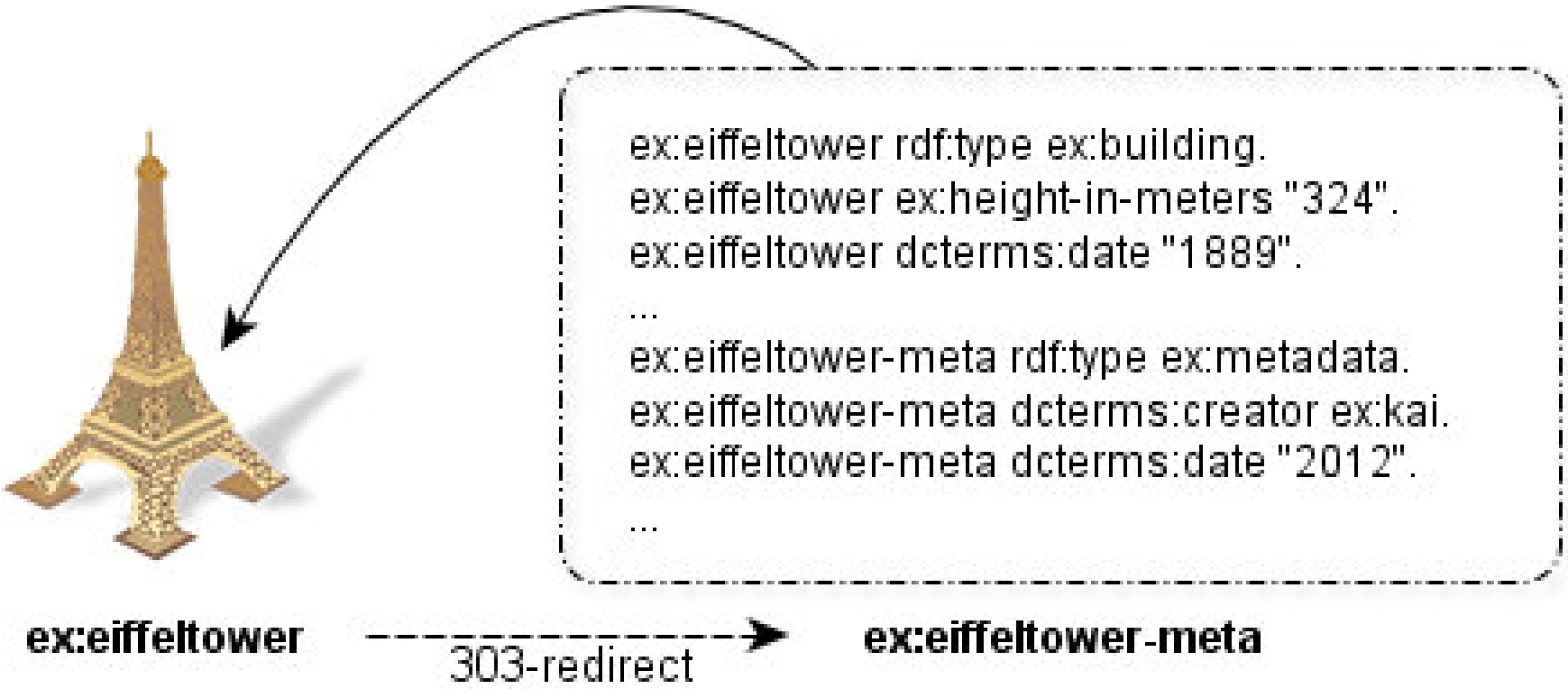
Linked Metadata



How do we get the metadata provenance?

Usual best practice: deliver it with the metadata.

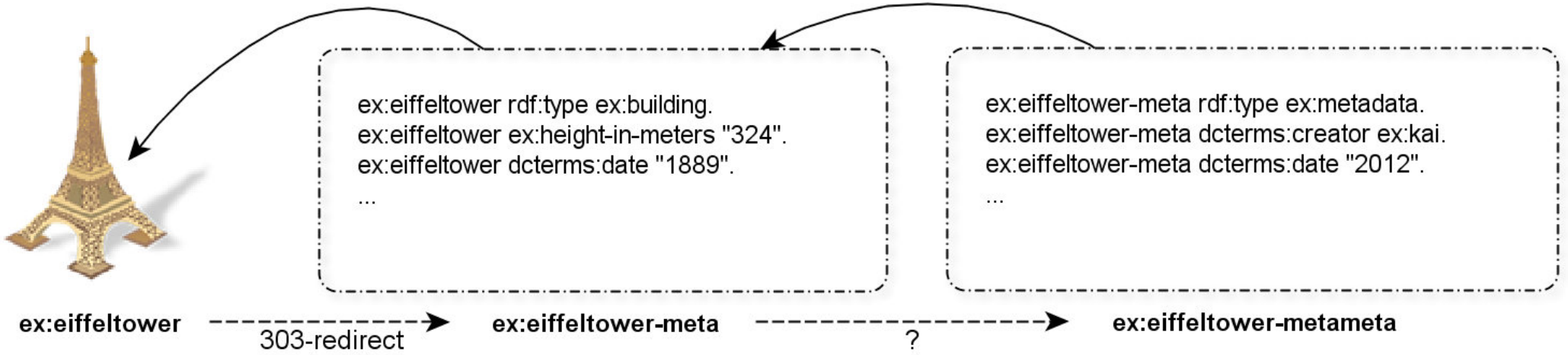
Embedded Linked Metadata (Method 1)



Drawback:

What about the provenance of the provenance?
There is no URI for the metadata provenance.

Linked Metadata



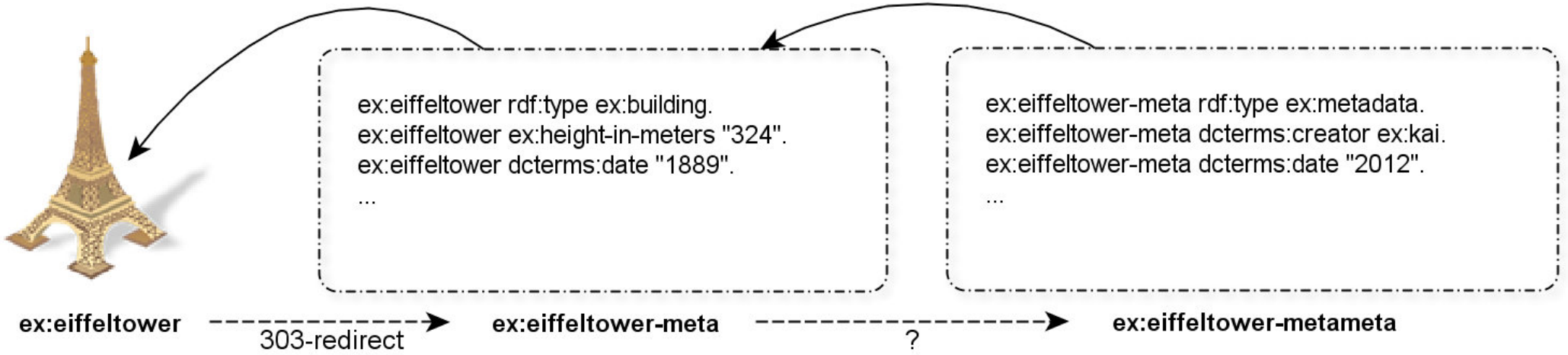
Then we give the metadata provenance a URI!

Problem: How to tell that we want the provenance.

Content negotiation is not working any more, as both contents are RDF.

Missing: A **request header** that asks for provenance.

The Link Header (Method 2)

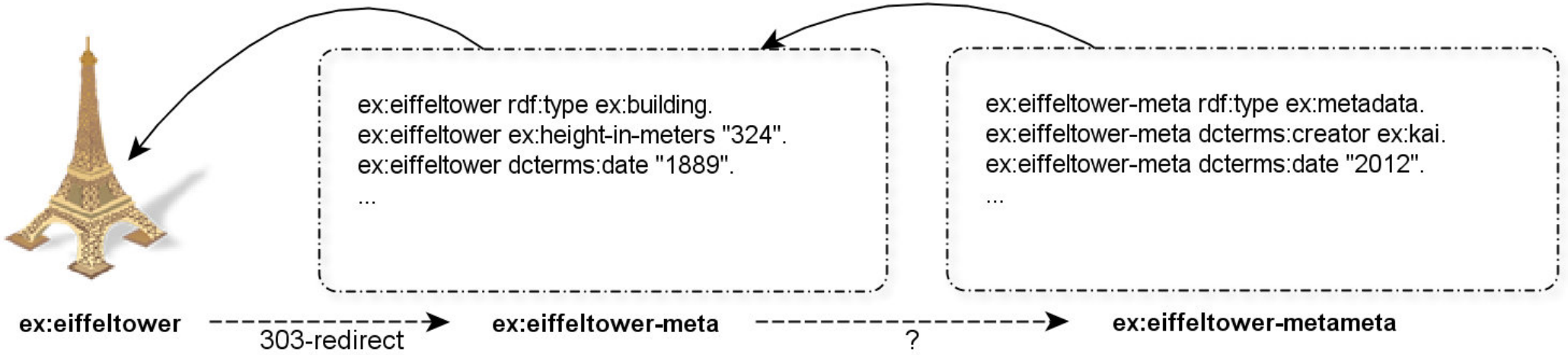


Response header sent by ex:eiffeltower-meta:

```
Link: <http://example.org/eiffeltower-metameta>; rel=meta
```

Drawback: Additional (head) request needed.

Additional Statements (Method 3)



Provide a **reference** to the provenance data:

```
ex:eiffeltower-meta rdfs:seeAlso ex:eiffeltower-metameta.

ex:eiffeltower-meta
   prov:has_provenance ex:eiffeltower-metameta.
```

Drawback: rdfs:seeAlso very general. PROV is very new, but should be preferred, especially if PROV is used.

The new URNs

IETF Working Draft:
http://tools.ietf.org/html/draft-saintandre-urnbis-2141bis

Replaces RFC 2141 (URNs)

Section 6.1: *"If a **query component**, **fragment identifier component**, or both have been appended to the assigned URI, they **MUST be ignored** for purposes of determining equivalence."*

*Section 4.3: "This specification does **not define** the applicability and **semantics** of the query component or the fragment identifier component in URNs."*

Possible use-case: **`urn:example:54321`**`?metadata`

What about #? ?

Query:

`http://www.example.org/shop/showDetails?product=2652917`

Fragement Identifier:

`http://www.example.org/blogposts/2013-11-25/the-tutorial.html#TableOfContents`

`http://www.w3.org/2000/01/rdf-schema#label`

Problem: Neglecting query and fragment identifier for URI equivalence violates WWW (and Linked Data!) practice.

PROV-AQ: Provenance Access and Query

HTTP header:

```
Link: <provenance-URI>;
   rel="http://www.w3.org/ns/prov#has_provenance";
   anchor="target-URI"
```

Provenance Query Services:

```
<http://example.com/prov/service>
   a prov:ServiceDescription;
   prov:describesService _:direct .

_:direct
   a prov:DirectQueryService ;
   prov:provenanceUriTemplate
      "http://www.example.com/provenance/service?
      target={uri}" .
```

Linked Metadata Summary

+ Based on Linked Data Principles.
+ Current "best practice."

- Not suitable for provenance on statement level.
- Requires full control over web server.
- No URI for provenance information, **or**
- provenance retrieval requires HTTP information: is this "follow your nose"?

Despite the drawbacks: a good starting point, as every provenance mechanism has to fit with the linked data principles.

Named Graphs

Named Graphs

A Named Graph is an RDF graph with an assigned URI as name.

Serialization in TriG:

```
ex:eiffeltower-meta {
      ex:eiffeltower rdf:type ex:building.
      ex:eiffeltower ex:height-in-meters "324".
      ex:eiffeltower dcterms:date "1889".
      ...
   }
```

Named Graphs will be part of the RDF 1.1 standard, and are supported in SPARQL.

Named Graphs in RDF Stores

RDF-Stores today are usually **quad-stores.**
(not triple-stores, even if we call them that way)

Each triple is assigned to a graph via the **fourth** quad element.

If the fourth element contains a URI, the URI is interpreted as the **name of the graph** that contains all triples with the same graph URI.

Named Graphs and SPARQL

SPARQL supports Named Graphs:

```
SELECT ?origin ?p ?o WHERE {
   GRAPH ?origin {
      :MonaLisa dc:creator :LeonardoDaVinci .
   }
   ?origin ?p ?o .
}
```

This retrieves all statements about graph URIs containing a certain statement (e.g., provenance).

Named Graphs and Linked Data

A client that fetches linked data via a URI **usually** stores this URI as graph URI in a quad store.

This is **great**, because this way we can talk about the fetched RDF data and store provenance in our RDF store.

This is **only half way there**, because we can not reexpose the provenance information easily.

Because it is not (yet) part of RDF.

RDF 1.1

RDF WG

Mission:

Update the 2004 RDF Recommendations, extending RDF to include features desirable and important for interoperability, but without a negative effect on deployment.

Required Feature (Charter) among others:

Support for **Multiple Graphs** and Graph Stores.

Standardize the Turtle RDF Syntax. Either that syntax or a related syntax should also support **multiple graphs**.

http://www.w3.org/2011/01/rdf-wg-charter

Named Graphs in RDF 1.1 (Work in Progress!)

From RDF 1.1 Concepts and Abstract Syntax
(W3C Candidate Recommendation 05 November 2013):

> An RDF Dataset is a collection of RDF graphs and comprises [...] zero or more named graphs.
>
> Each named graph is a pair consisting of an IRI or a blank node (the **graph name**), and an **RDF graph**.
>
> Note:
>
> > The graph name does **not** formally denote the graph.
> >
> > RDF does **not** place any formal **restrictions** on **what resource the graph name may denote**, nor on the **relationship between that resource and the graph**.

RDF Graphs

What is an RDF Graph?

An RDF graph is a set of RDF triples.

That means that a (named) RDF Graph does not contain other (named) graphs.

Consequences:

You can reexpose graphs with names (e.g., with TriG),
but: no directions how to interprete the graph URI,
and: when the TriG file is fetched, no possibility to store the graphs inside another graph with the URI of the TriG file.

Summary

Half way there,
but still enough room
for own decisions and
developments.

Positive thinking ;-)

Metamodels in Practice

Infrastructure vs. Data Model

Retrieval URL, Content Negotiation, Link Header, Query Services all belong to the **infrastructure**.

A **data model** forms the basis of your data. You want to be able to retrieve your data, to store your data, to publish your data completely – wasn't that the idea of RDF in the first place? Any important information (like provenance) must be part of the data model.

If you use RDF as model for your data model, and triple stores as databases, you are limited by their limits. Need not to be a problem, but in any case, ***be aware of these limits****.*

Metamodels

Metamodels are based on RDF, but provide means to talk about RDF data on a metalevel.

We briefly introduce the follwing:

- OAI-ORE and Europeana Data Model
- VoID

OAI-ORE and EDM

OAI-ORE

Open **A**rchives **I**nitiative - **O**bject **R**euse and **E**xchange

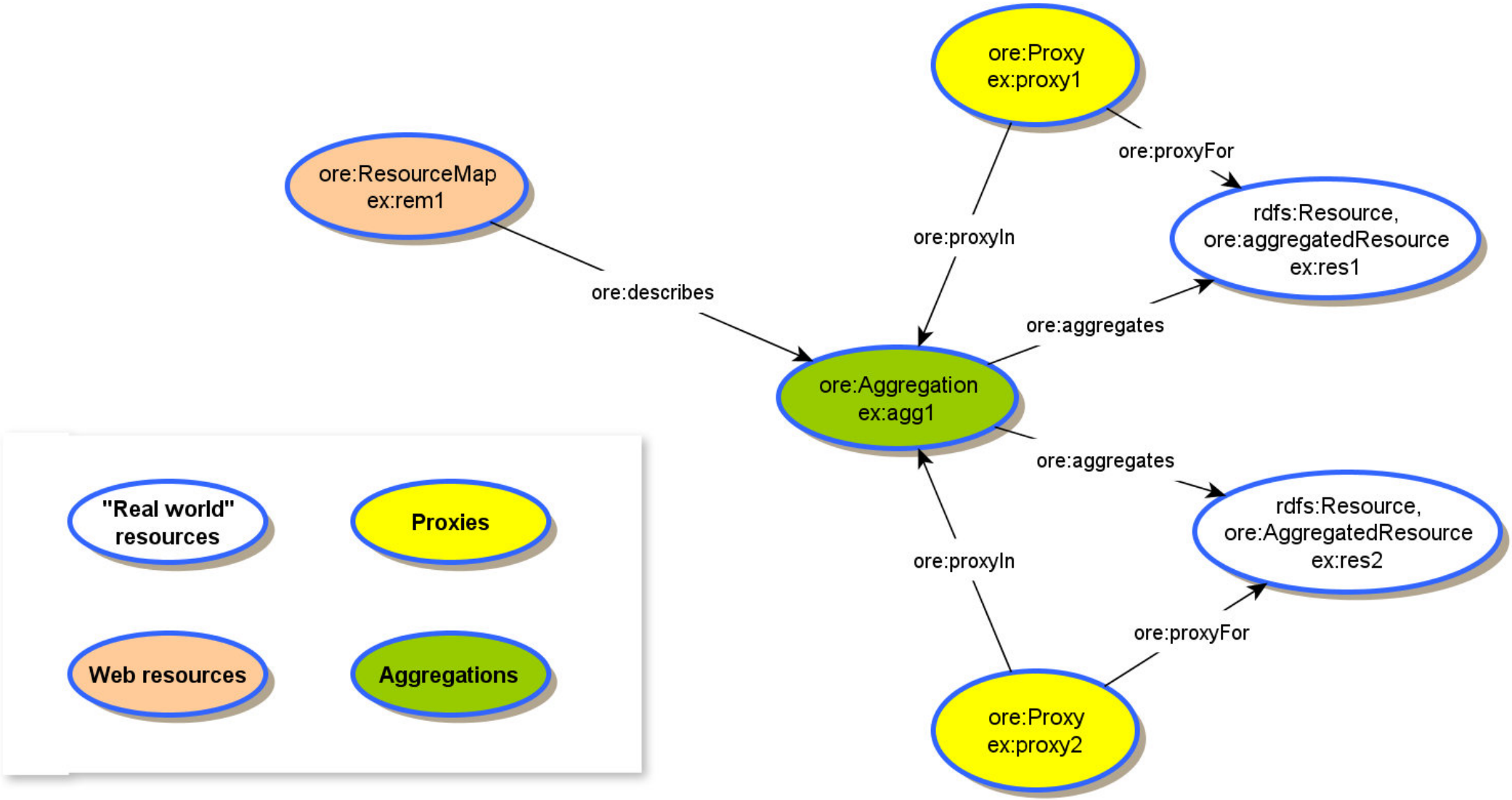
Originally addresses another problem that lacks a solution in RDF:

> How to make a statement about a resource that is only valid in a special context?
>
> Example: The ordering of resources in an aggregation, like the ordering of articles in a bibliography.

Adaption for provenance:

> All statements are provided within such a context, the context can be identified and further described by provenance statements.

OAI-ORE Graph



OAI-ORE and Linked Data

The Resource Map is just a web resource with an own URI.

The Resource Map is connected to the Aggregation via ore:describes.

The Aggregation and the Proxies provide the scaffolding for the statements that are made in the context of the Aggregation.

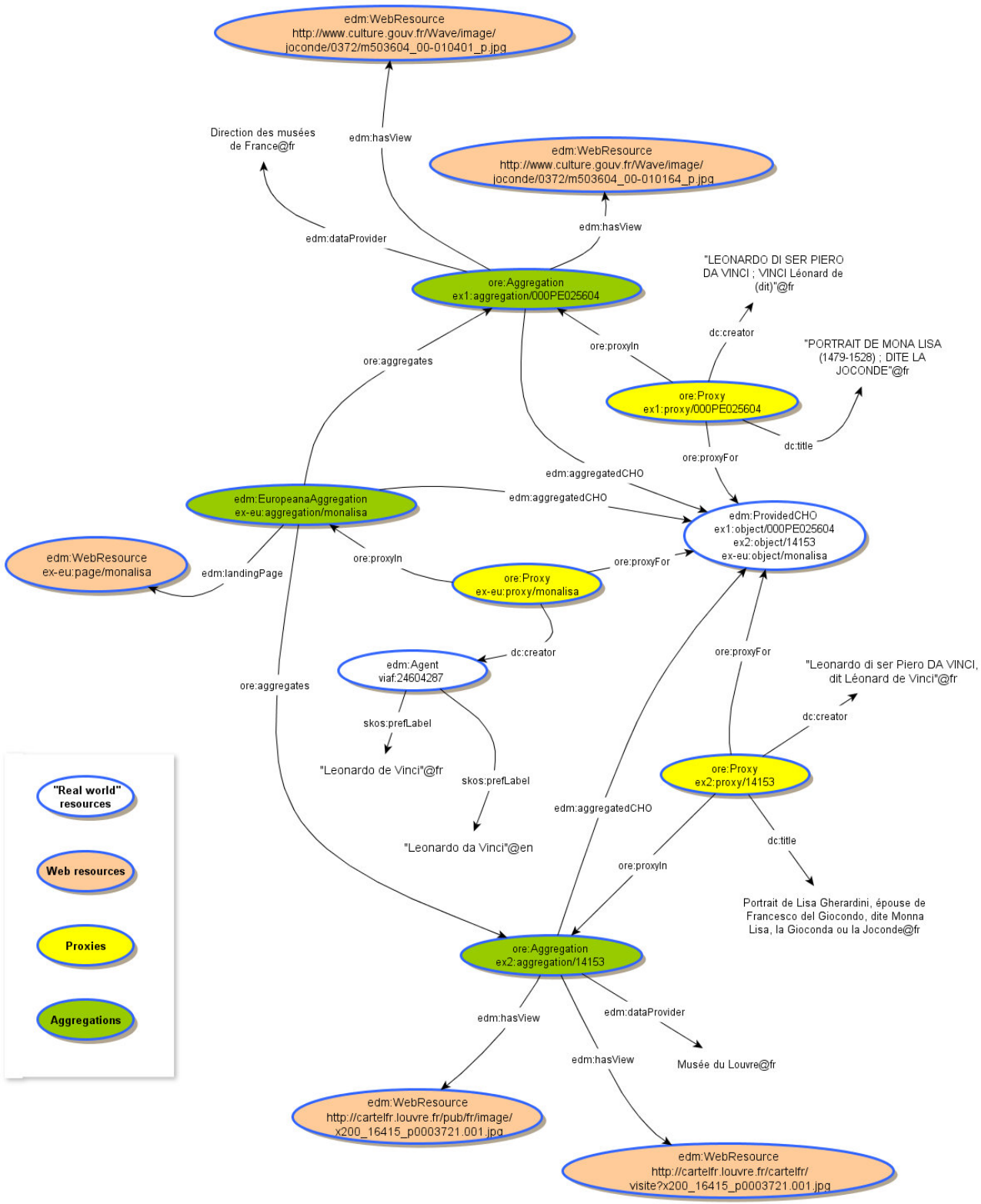
Drawback: An application has to be "ORE-aware" to make sense of all this, as the concept of a proxy resource is not known in RDF.

Europeana Data Model

Europeana provides data about **cultural heritage objects (CHO)** from CH institutions all over Europe.

Provenance requirement: Distinguish metadata from different institutions talking about the same (owl:sameAs) resource.

The Europeana Data Model (EDM)



Provenance realized by means of OAI-ORE.

Problems?

Users have to understand **Proxies**.

Users have to understand **Aggregations**.

How are proxies and aggregations used?

What is an aggregation?

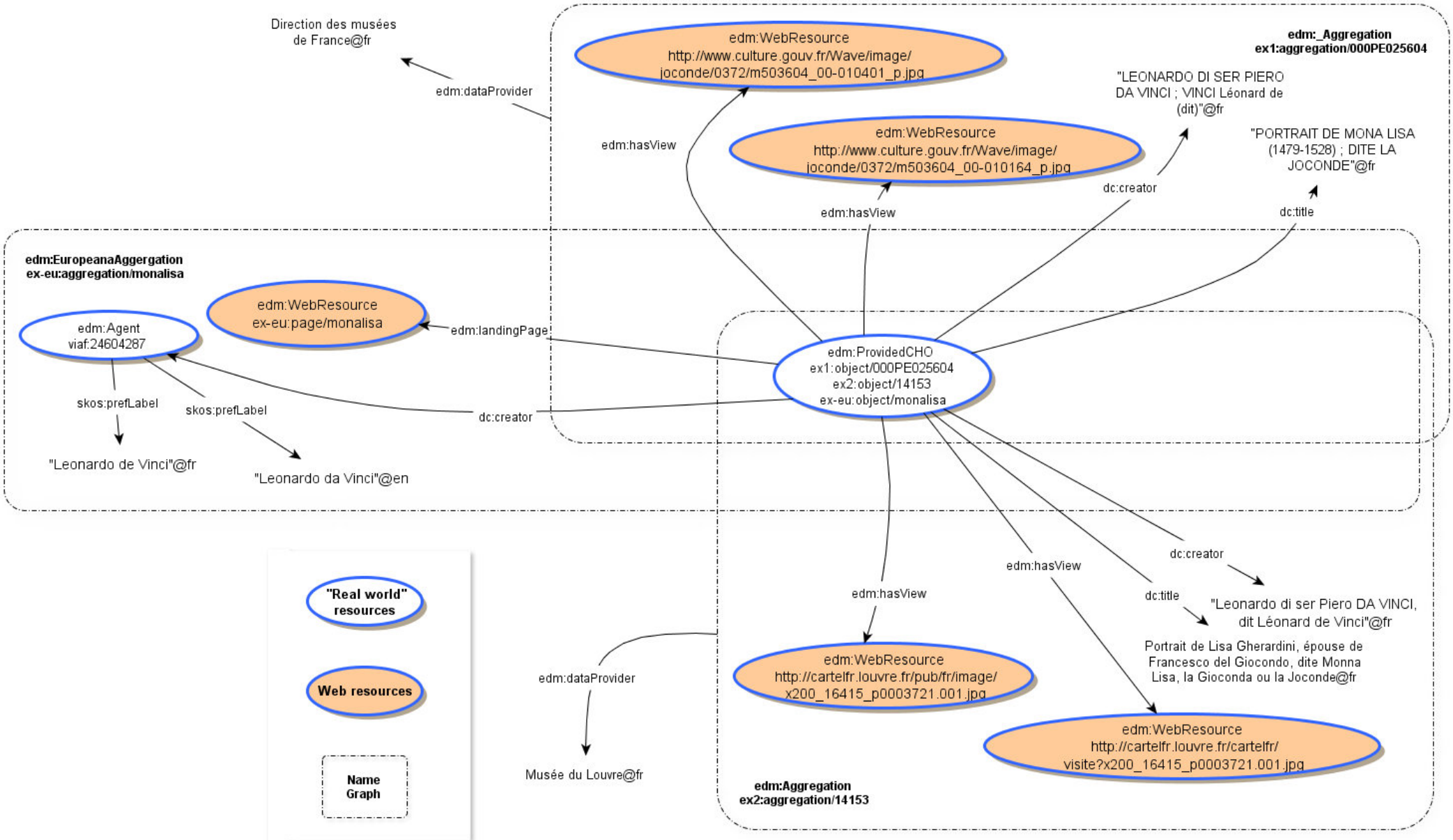
*"Aggregations are used in Europeana to **represent the complex constructs** that are provided by contributors. An aggregation **is associated to the object that it is about**, by the property edm:aggregatedCHO."*

Level of aggregation:

1 aggregation per providedCHO.

EuropeanaAggregation aggregates other aggregations (from data providers).

Overlapping Resource Descriptions: We want to talk about Graphs!



The **Vo**cabulary of **I**nterlinked **D**atasets
VoID

What's inside our store?

RDF Datasets, organized in named graphs.

NG URI scheme:

http://example.org/dataset/[provider]/[datasetId]/[version]

VoID (http://www.w3.org/TR/void/):

Each named graph is a `void:Dataset`.

Additional provenance statements for each dataset.

Make it available

Web documents (with URI) deliver RDF, provenance is included as statements about the URI.

Each Web document is a `foaf:Document`.

Each Web document contains a statement that links to the `void:Dataset`:

```
ex:doc1 void:inDataset ex:dataset1 .
```

Example for a RESTful API (Web documents)

http://example.org/...

... **dataset**/[provider]/[datasetID]/**[version]**
=> (Provenance) information about the dataset

... **resource**/[provider]/[identifier]
=> 303 to latest version

... **dataset**/[provider]/[datasetID]/**[version]**/[identifier]
=> data about a single resource

... **linkset**/[provider]/[linksetID]/**[version]**
=> additional links from a different source

Hint: Documents contain a **[version]**.

Provenance in Documents

Generated from provenance information about datasets:

dc:creator => Data provider
dc:date => Timestamp

ex:version => version number
ex:nextVersion => link to next version of the document
ex:previousVersion => link to previous version
ex:links => link to a linkset

PROV statements for full provenance chain.

Version means always the version of the underlying dataset.

Consuming the data

Linksets and data enrichments are managed as seperate datasets.

All statements in a dataset share the same Provenance.

Applications have to combine the data as needed.
=> Preservation of provenance is left to consumer.

Storing the data

How should the data be **organized on client side**?

1. A named graph per retrieved URL, as usual?
2. Or a named graph per dataset, which would replicate the organization on the server?

Both is possible, but depending on the application one or the other way might be preferred.

Summary

Many different approaches:

1) Reification
2) "Simple" application of Linked Data principles.
3) Named Graphs
4) OAI-ORE, VoID
5) Own models and extensions

In practice, we have to combine them to create flexible solutions. Unfortunately, the full understandability of linked data provenance is not (yet) guaranteed.

Linked Data Publishing and Provenance

Agenda

State-ful or State-less Data

Versioning

Identity and Provenance Context

State-ful data

Content on web pages can change, they are usually **state-less**.

Example for a state-less URL:

http://example.org/weather/lisbon

By commitment, the content of a URL can be kept stable, the URL represents a specific state,
it is **state-ful**.

Example for a state-ful URL:

http://example.org/weather/lisbon/2013-09-02

Example: Wikipedia

en.wikipedia.org/wiki/Provenance

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

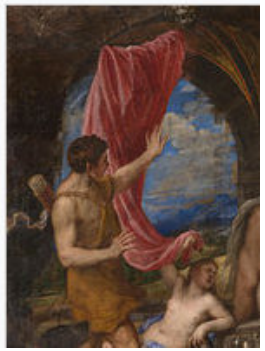
Interaction
Help
About Wikipedia
Community portal
Recent changes

Article | Talk | Read | Edit source | Edit beta | View history | Search

Provenance

From Wikipedia, the free encyclopedia

For other uses, see Provenance (disambiguation).



Provenance, from the French *provenir*, "to come from", refers to the chronology of the ownership, custody or location of a historical object.[1] The term was originally mostly used in relation to works of art, but is now used in similar senses in a wide range of fields, including archaeology, paleontology, archives, manuscripts, printed books, and science and computing. The primary purpose of tracing the provenance of an object or entity is normally to provide contextual and circumstantial evidence for its original production or discovery, by establishing, as far as practicable, its later history, especially the sequences of its formal ownership, custody, and places of storage. The practice has a particular value in helping authenticate objects. Comparative techniques, expert opinions, and the results of scientific tests may also be used to these ends, but establishing provenance is essentially a matter of documentation.

In archaeology (particularly North American archaeology and anthropological archaeology throughout the world), the term **provenience** is used in related but a subtly different sense to *provenance*. Archaeological researchers use *provenience* to refer to the three-dimensional location or **find spot** of an artifact or feature within an archaeological site,[2] whereas *provenance* covers an object's complete documented history. Ideally, in modern excavations, the provenience or find spot is recorded (even videoed) with great precision, but in older cases only

Provenance: Revision history

View logs for this page

Browse history

From year (and earlier): 2013 From month (and earlier): all

For any version listed below, click on its date to view it. For more help, see Help:Page history
External tools: Revision history statistics · Revision history search · Contributors · Use

(cur) = difference from current version, (prev) = difference from preceding version, **m** = minor
(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions

- (cur | prev) 12:55, 27 August 2013 Johnbod (talk | contribs) . . (24,233 bytes) (-95
- (cur | prev) 12:50, 27 August 2013 Johnbod (talk | contribs) . . (24,328 bytes) (+1
- (cur | prev) 18:41, 7 July 2013 GrindtXX (talk | contribs) **m** . . (24,139 bytes) (-4) .
- (cur | prev) 20:20, 24 June 2013 Rjm at sleepers (talk | contribs) . . (24,143 bytes
- (cur | prev) 19:44, 24 June 2013 GrindtXX (talk | contribs) **m** . . (24,077 bytes) (0)
- (cur | prev) 18:26, 24 June 2013 128.238.160.227 (talk) . . (24,077 bytes) (+156) .
- (cur | prev) 22:15, 24 May 2013 Beorhast (talk | contribs) **m** . . (23,921 bytes) (+9
- (cur | prev) 16:05, 23 April 2013 AnomieBOT (talk | contribs) **m** . . (23,830 bytes)

Wikipedia provides a stable URL for every version of an article, the content of these URLs won't change, the URLs are **state-ful**.

en.wikipedia.org/w/index.php?title=Provenance&oldid=570390671

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help

Article | Talk | Read | Edit source | Edit beta | View history | Search

Provenance

From Wikipedia, the free encyclopedia

This is the **current revision** of this page, as edited by **Johnbod (talk | contribs)** at 12:55, 27 August 2013. The present address (URL) is a **permanent link** to this version.

(diff) ← Previous revision | Latest revision (diff) | Newer revision → (diff)

For other uses, see Provenance (disambiguation).

Provenance, from the French *provenir*, "to come from", refers to the chronology of the ownership, custody or location of a historical object.[1] The term was originally mostly used in relation to works of art, but is now used in similar senses in a wide range of fields, including archaeology, paleontology, archives, manuscripts, printed books, and science and computing. The primary purpose of tracing the provenance of an object or entity is normally to provide contextual and circumstantial evidence for its original production or discovery, by establishing, as far as practicable, its later history, especially the sequences of its formal ownership, custody, and places of storage.

State-ful RDF and provenance

State-ful URLs make provenance-life easier.

The URL represents the data, so it can be used to identify the fetched data in local systems without problems.

State-less URLs are no show-stopper.

But the fact that the data might have changed in the source should be indicated:

1. Use a local state-ful URL for your data.
2. Link to the state-less URL as source, e.g., via `dct:source` or `prov:wasDerivedFrom`.

Versioning

Data always changes. Most applications with state-ful URLs will therefore need versioning.

The necessary links to other versions can be included with the data.

Versioning vocabulary

previousVersion: links to the previous version of this dataset.

firstVersion: links to the oldest available version of this dataset.

version: serial number of this version, starting with 1.

versionName: provides a human-readable name for this version.

nextVersion: links to the next version of this dataset.

latestVersion: links to the latest available version of this dataset.

availableVersions: number of available versions of this dataset.

Avoid changing properties in your data

~~**nextVersion:** links to the next version of this dataset.~~

Replace with a link to a state-less generic URL:

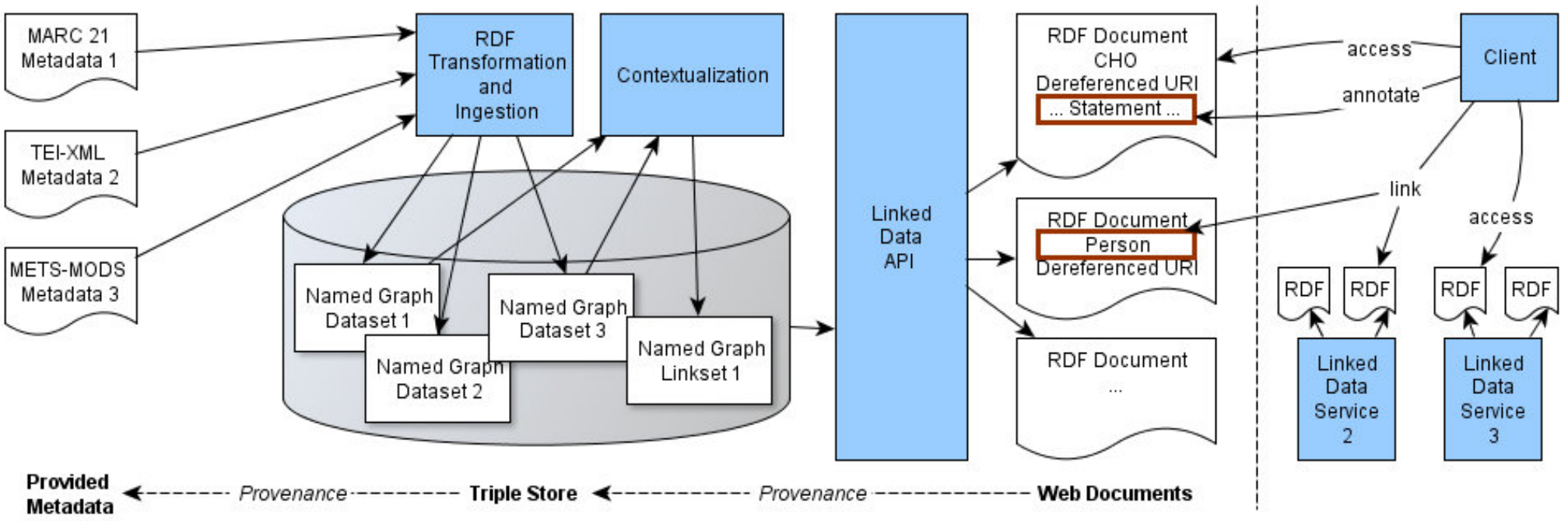
```
ex:doc1/version1 ex:isVersionOf ex:doc1
```

The following information is then linked to the **generic URL:**

latestVersion: links to the latest available version of this dataset.

availableVersions: number of available versions of this dataset.

Linked Data Publishing

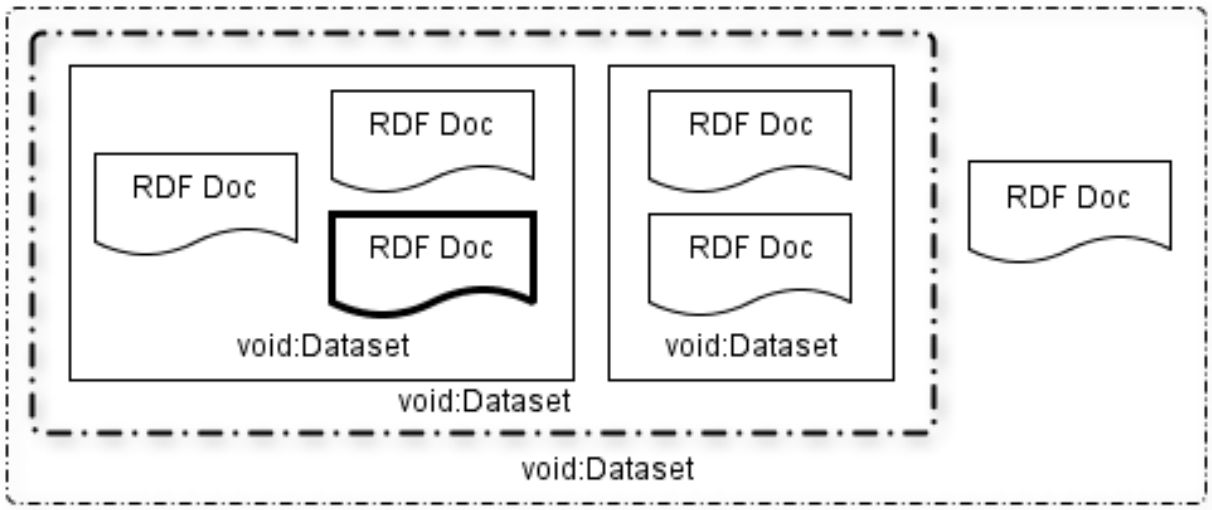


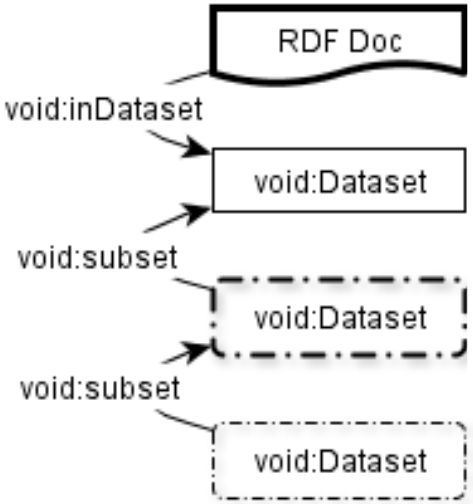
Too many options?

Web document URLs can be preserved as named graphs in a local triple store.

So can VoID datasets.

VoID datasets can be nested...





Triple Identity

Several sources can make the same statement. No distinction within RDF.

Statements (Triples) can be retrieved and become part of a new dataset.

A statement has no identity.

Can we establish triple identity?

What determines identity?

Philosophical Question.

Proposed Answer:

The provenance of a resource determines its identity.

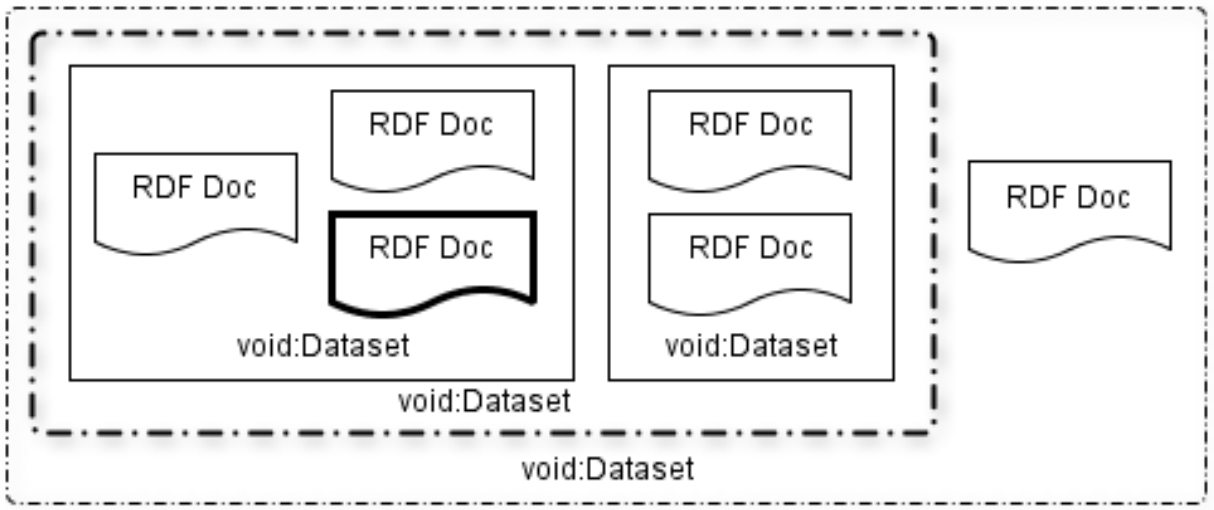
„If we want to preserve the identity of the statements in our data, we have to preserve their provenance.“

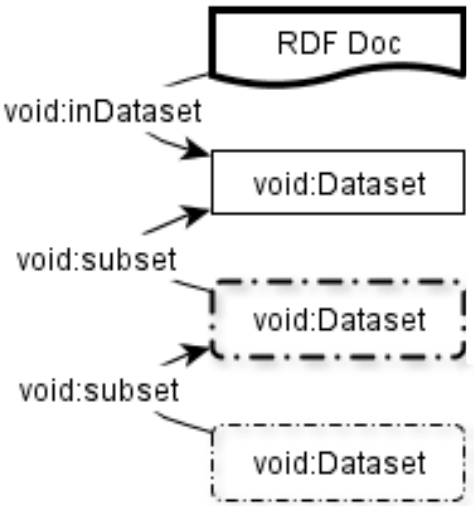
Provenance Context

One of our nested graph boundaries (hopefully) was created to provide provenance information.

✳ Provenance Context

To enable preservation of triple identity, we indicate the Provenance Context.





Definition

"A Provenance Context is a set of RDF triples that share the same provenance, identified by a URI."

Web documents (foaf:Document)

VoID Datasets (void:Dataset)

Named Graphs

ORE Resource Maps (ore:ResourceMap)

...

Determination of the Provenance Context

Per default, the Provenance Context of a triple is the **document** identified by the URL it is retrieved from **or** the **Named Graph** that contains the statement.

If the document or the Named Graph is related to a void:Dataset via void:inDataset, the Provenance Context is the **void:Dataset**.

The Provenance Context can be stated explicitly using the property **dm2e:inProvenanceContext**.

Consequences

✳ There must always be one and only one Provenance Context for each statement.

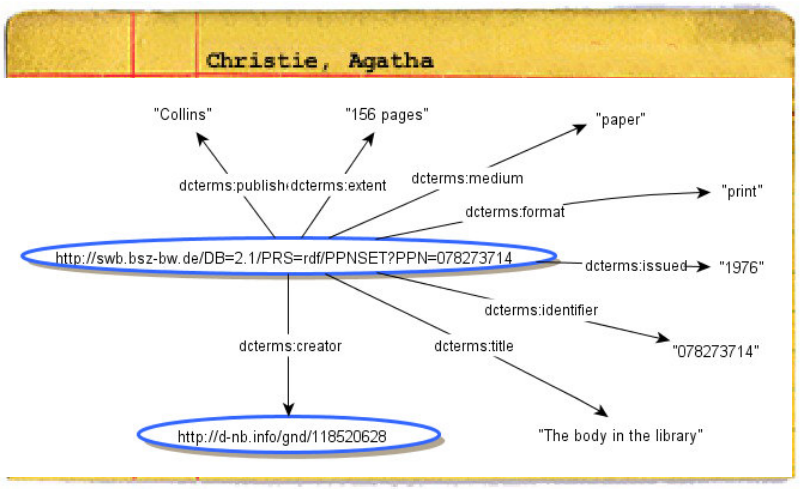
✳ Every RDF graph either **is** a Provenance Context or it is contained **completely** within one Provenance Context.

✳ The Provenance Context determines the maximum permissible set of RDF statements that are published together.

The Provenance Context in DCAM

Description Set: Logical Boundary that creates identity.

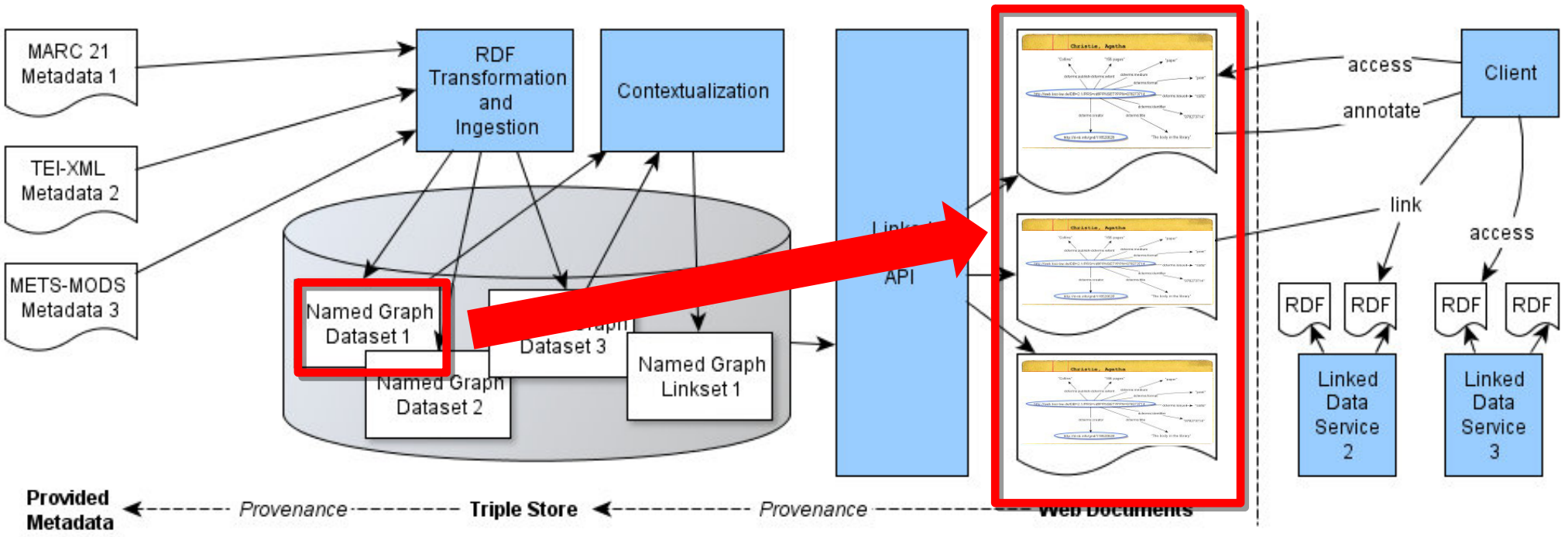
Record: Physical embodiment of a Description Set.



DCAM and Linked Data

Any RDF publication is a Record containing a Description Set.

These Description Sets are part of a larger Description Set, the Provenance Context.



Triple Identification in Linked Data

Idea: Use an XPointer-style way to point to statements within a Provenance Context.

```
<scheme name>:<hierarchical part>[?<query>][#<fragment>]
```

Fragment: spo=subject,predicate,object

Example

```
http://example.org/provcontext1?spo=%3Chttp%3A%2F
%2Fexample.org%2Fdata%2Fdoc1%3E,%3Chttp%3A%2F
%2Fpurl.org%2Fdc%2Fterms%2Fcreator%3E,%3Chttp%3A%2F
%2Fexample.org%2Fpersons%2Fkai%3E
```

Statement:

```
<http://example.org/data/doc1>
   <http://purl.org/dc/terms/creator>
   <http://example.org/persons/kai>.
```

within the Provenance Context:

```
<http://example.org/provcontext1>
```

What does this mean?

The fragment URIs can be created and interpreted on the fly.

But semantics in the URI are an anti-pattern.

So let‘s explain what the URI represents.

Contextual Reification

```
<http://example.org/provcontext1?spo=%3Chttp%3A%2F
%2Fexample.org%2Fdata%2Fdoc1%3E,%3Chttp%3A%2F%2Fpurl.org
%2Fdc%2Fterms%2Fcreator%3E,%3Chttp%3A%2F%2Fexample.org
%2Fpersons%2Fkai%3E>

        a  rdf:Statement ;
        rdf:subject <http://example.org/data/doc1> ;
        rdf:predicate <http://purl.org/dc/terms/creator> ;
        rdf:object <http://example.org/persons/kai> ;
        dm2e:context <http://example.org/provcontext1> .
```

Dereferencing the URI explains the meaning. It is a Statement (Reification), connected to a specific Provenance Context.

Provenance Context and Contextual Reification

- Provenance-tracking for data requires data identity.
- For the preservation of data identity, we need guidance.
- The Provenance Context abstracts from technical details and indicates the bounday that defines data identity.
- Furthermore, we can use it to connect statements about statements (annotations) to a concrete context.
- Technical issues (length!) with the fragment URIs still have to be investigated.

Practical Implications

No publishing of merged statements from different sources.

Leave the merging to the consuming application.

Pedantic Web: Do not publish the provenance statements together with the data, if they do not share the same provenance.

Break these rules if you have to ;-)

Summary

The problem of metadata provenance is the **stable identification** of data.

The problem gets worse if the data starts to move around, i.e., **when it is consumed and republished**.

There are **limitations** for clean solutions **resulting from the web architecture**.

If you know these limitations, you can create applications that work **perfect** for you...

... and reasonable **well** for all others (i.e., they follow common practices).

END

Acknowledgements

Tutorial provenance:

- Eckert/Pfeffer: Metadata Provenance Tutorial, SWIB 2012
- Kai Eckert: Metadata Provenance Tutorial, DC 2013, together with a PROV Tutorial by Daniel Garijo
- Eckert/Pfeffer: Metadata Provenance Tutorial, SWIB 2013

SWIB 2013

Tutorial

on

Metadata Provenance

Slides: http://bit.ly/swib13-provenance

Part 2:

Modelling provenance information using the PROV ontology

Agenda

- Modelling Provenance 1
  - A data model for provenance information
  - Introducing the PROV ontology
  - Extending the basic elements of PROV
- Short break
- Modelling Provenance 1
  - Qualifying relations in PROV
  - Mapping DC provenance information to PROV

A data model for provenance information

Motivation

Now that we have a handle to our data, we want to describe its provenance

Options for expressing provenance

Using existing generic vocabularies

Extending/creating a domain-specific vocabulary

Using/Creating a vocabulary specifically made for this purpose

Using a generic vocabulary: Dublin Core

- Dublin Core Metadata Initiative (DCMI)
  - Element set
    - 15 basic terms
    - No defined ranges (--> arbitrary values possible)
  - Terms
    - 55 granular terms (properties)
    - Well defined ranges

Example

Namespace

Element set --> dc:

Terms --> dcterms: or dct:

```
ex:doc1 dct:title "A mapping from Dublin Core..." .
ex:doc1 dct:creator ex:kai .
ex:doc1 dct:created "2012-02-28" .
ex:doc1 dct:publisher ex:w3c .
ex:doc1 dct:issued "2012-02-29" .
ex:doc1 dct:subject ex:dublincore .
ex:doc1 dct:replaces ex:doc2 .
ex:doc1 dct:format "HTML" .
```

Distinction

Some terms contain only information about the resource itself

But not how or when it was produced

→ Descriptive Terms

Some terms also contain information on the creation or derivation of the resource

→ Provenance Terms

Provenance in DC: Who?

Terms

- Contributor
- Creator
- Publisher
- RightsHolder

Range is dct:Agent

- a resource that acts or has the power to act
- Clearly influencing creation of a resource
- RightsHolder is ownership --> provenance in works of art

Provenance in DC: When?

Terms

- Available
- Created
- Date
- DateAccepted
- DateCopyrighted
- DateSubmitted
- Issued
- Modified
- Valid

Provanance in DC: When?

- Ranges
  - Date range
    - Available, valid
  - Single date
    - All others
- Dates are basic provenance information
  - Availability and validity often inherent to the resource
  - But: provenance related, if active change

Provanance in DC: How?

Terms

- IsVersionOf, hasVersion
- IsFormatOf , hasFormat

Derivation and Replacement

- References, isReferencedBy
- Replaces, isReplacedBy
- Source
- HasPart, isPartOf

Relations to other resources

- accrualMethod

Processes involved in creation

dcterms:provenance

Definition

> “statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity, and interpretation.”
>
> → ”classic” provenance of works of art

Summary

More than half of the DC terms deal with provenance related information

- Who?
- When?
- How?
- What?

Missing information

- Where?
- Why?
  - (only for the specific reason of replacement)

Extending a domain-specific vocabulary

Domain-specific vocabularies often deal with aspects of provenance

e.g. the SWAN Ontology (Semantic Web Applications in Neuromedicine) has a module dealing with "Provenance, Authoring and Versioning (PAV)"

→ Aspects, granularity and terminology differ between domains

Cross-domain data exchange becomes very hard

Example: PAV module of SWAN

properties

- importedBy - An entity responsible for importing the data from an external source
- importedOn - The date of the import of the resource
- importedFirstOn - The date of the first import of the resource
- importedLastOn - The date of the last import of the resource
- importedFromSource - The original source of the encoded information (PubMed, UniProt…)
- importedWithId - The unique identifier of the encoded information in the original source.

See http://swan-ontology.googlecode.com/svn/tags/1.2/pav.owl (latest version from 2008)

Example: PAV module of SWAN

properties

- sourceAccessedOn - The date when the original source has been accessed to create the resource.
- sourceFirstAccessedOn - The date when the original source has been first accessed and verified
- sourceLastAccessedOn - The date when the original source has been last accessed and verified

See http://swan-ontology.googlecode.com/svn/tags/1.2/pav.owl (latest version from 2008)

Extending a domain-specific vocabulary

Other ontologies have similar approaches

→ Aspects, granularity and terminology differ between domains

→ Cross-domain data exchange becomes very hard

Vocabularies for modelling provenance

Provenir

- Published in 2009

Open Provenance Model (OPM)

- Pulished in 2010

W3C Provenance Incubator Group (PROV-XG)

From 2009-2010

- Chaired by Yolanda Gil

"Provenance XG Final Report"

- http://www.w3.org/2005/Incubator/prov/XGR-prov/
- Overview of the existing approaches and vocabularies
- Proposes a dedicated W3C Working Group
  - Recommendation of an initial set of terms as a basis for further discussion

W3C Provenance Incubator Group (PROV-XG)

Discussion of requirements for provenance on the web

http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements

Mapping of provenance terms from existing vocabularies to OPM

http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings

Common use case scenarios for provenance

W3C Provenance Working Group

Active from 04/2011 to 07/2013

- Co-chaired by Paul Groth and Luc Moreau

Goal

- The mission of the Provenance Working Group [..] is to support the widespread publication and use of provenance information of Web documents, data, and resources. The Working Group will publish W3C Recommendations that define a language for exchanging provenance information among applications.

Main focus on linked data and the semantic web

W3C Provenance Working Group

Implementation of the PROV-XG recommendations

”A provenance framework should support:

the core concepts of identifying an object, attributing the object to person or entity, and representing processing steps;

accessing provenance-related information expressed in other standards;

accessing provenance;

the provenance of provenance;

reproducibility;

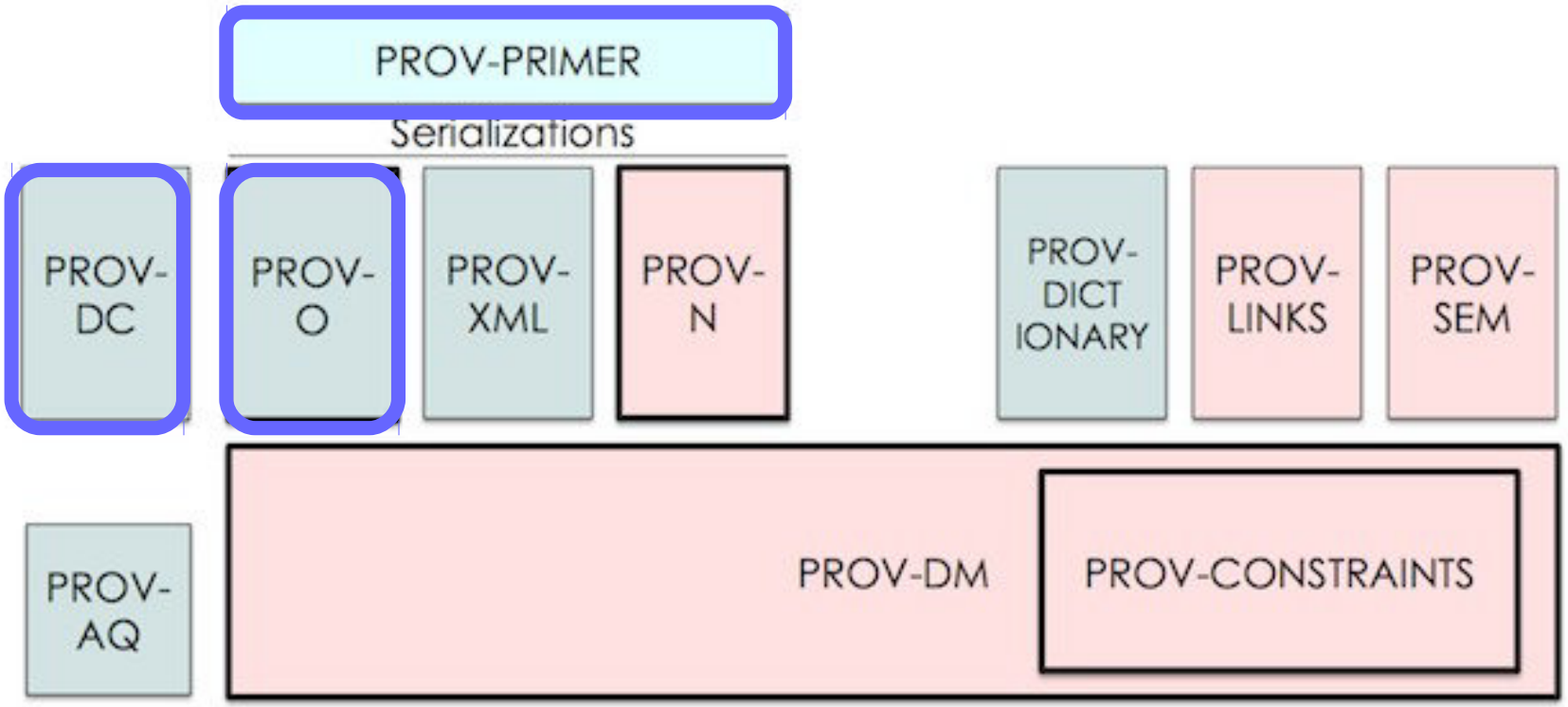
versioning;

representing procedures;

and representing derivation.”

Introducing the PROV ontology

PROV Ontology (PROV-O)



http://www.w3.org/TR/prov-overview/

Entities

PROV-O allows to record the provenance of entities

Entities are all kinds of things

- Physical: books, articles, reports, …
- Digital: pictures, text files, pdf documents, videos, …
- Conceptual/other: abstract concepts, ideas, theories, …

Provenance information can also include references to other entities

Activities

Model the dynamic aspects of the world

Occurs over a period of time and acts upon or with entities

- Includes consuming, processing, transforming, modifying, relocating, using, or generating entities

Examples

- Writing a report
- Translating a book
- Moving an online document to a new URL
- Generating web access statistics

Agents

Bear responsibility for

- an activity taking place
- for the existence of an entity
- for another agent's activity

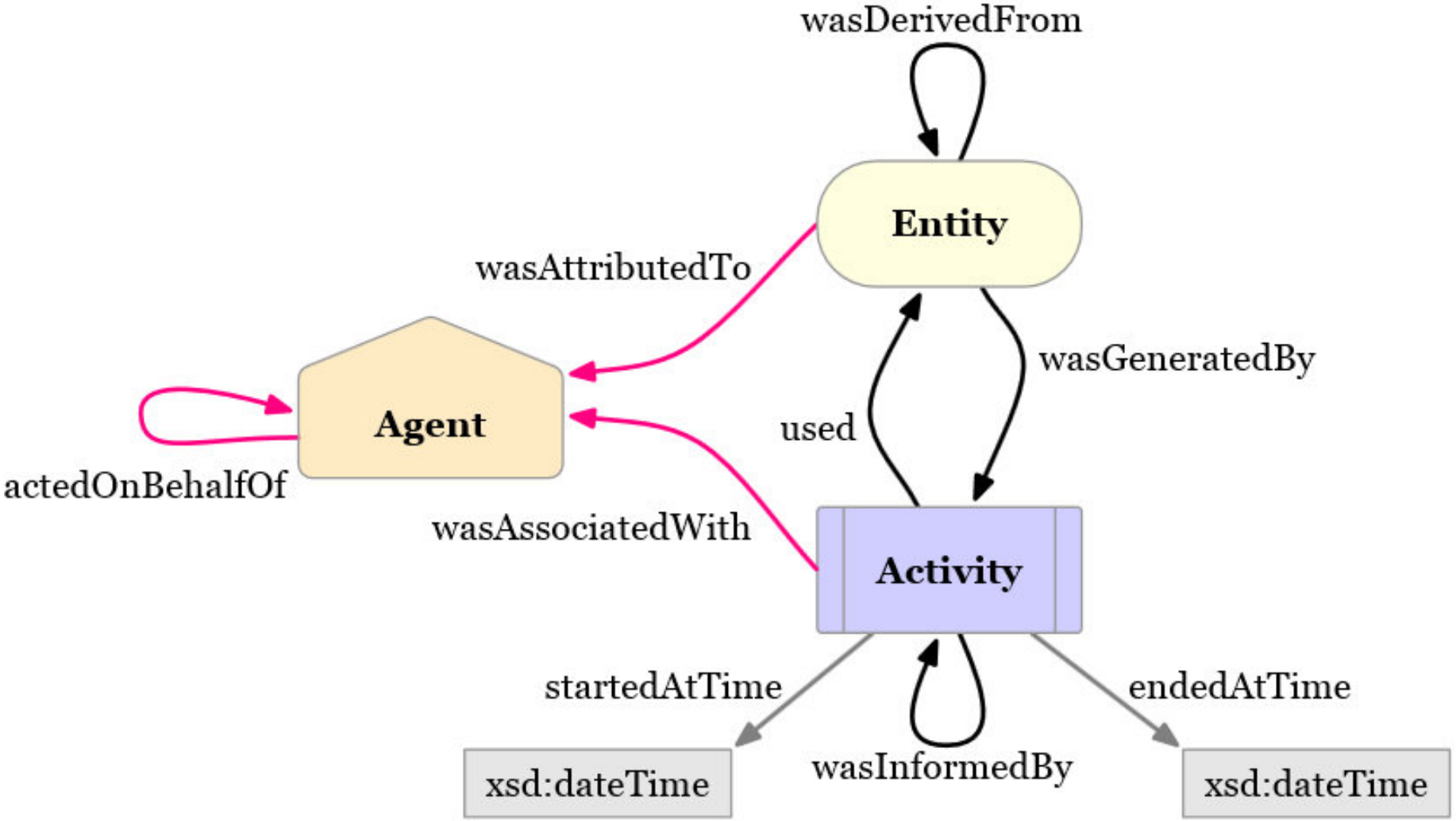
Examples

- Persons and organizations
- Inanimate objects
- Computer programs

Caveat:
One cannot describe the provenance of Agents.

To do so they have to be both Agents and Entities.

Prov-O Basic Elements



Source: http://www.w3.org/TR/prov-primer/

Starting properties

prov:wasAttributedTo

prov:wasGeneratedBy

prov:used

prov:wasAssociatedWith

prov:wasDerivedFrom

prov:startedAtTime

prov:endedAtTime

prov:wasInformedBy

prov:actedOnBehalfOf

Example: Provenance of a conference paper

The paper was written by a student

The final version of the paper is based on an earlier draft

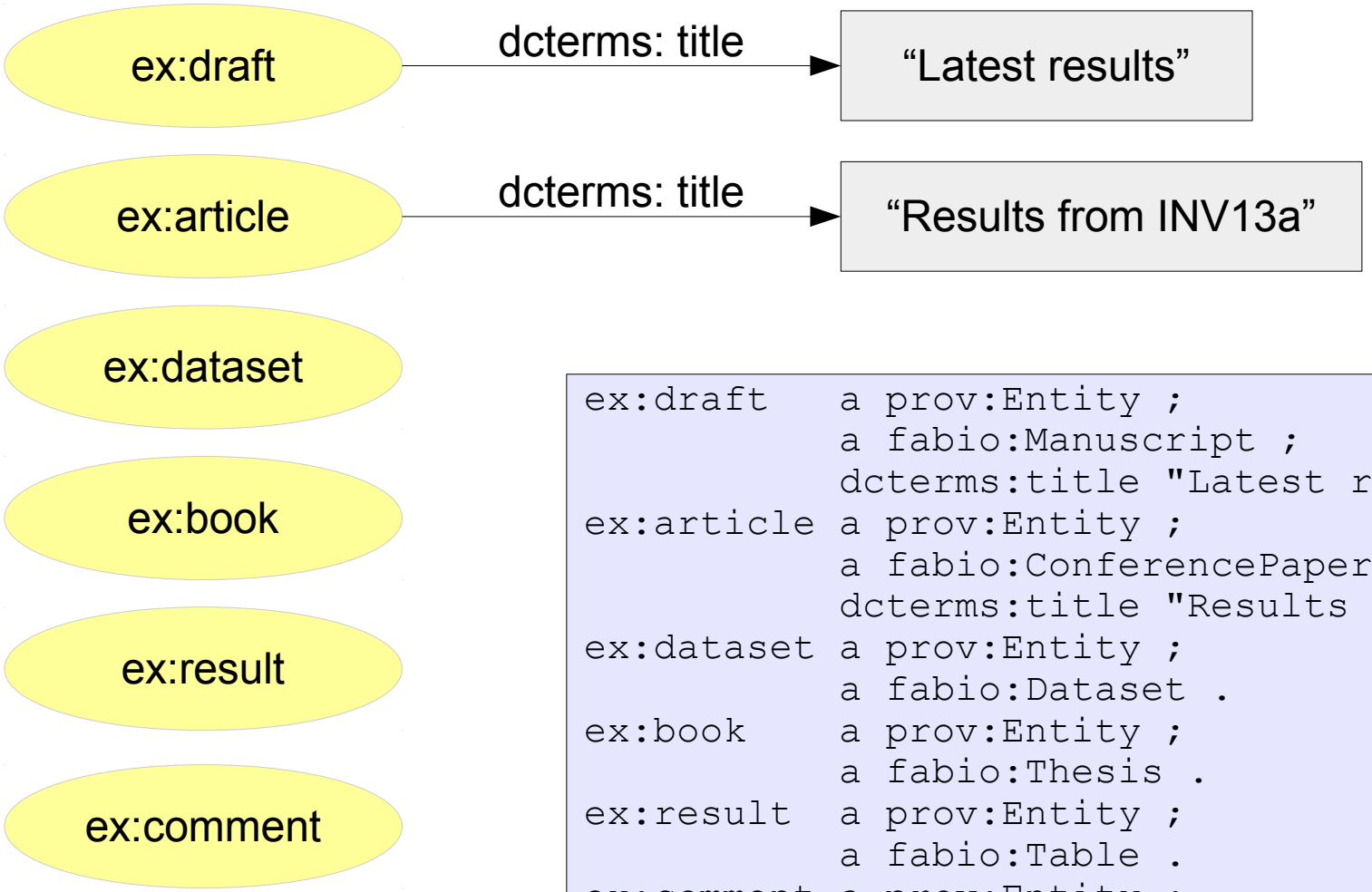
A professor made some comments on the draft

The student cites prior work from a book

The paper includes a table that was generated by a program

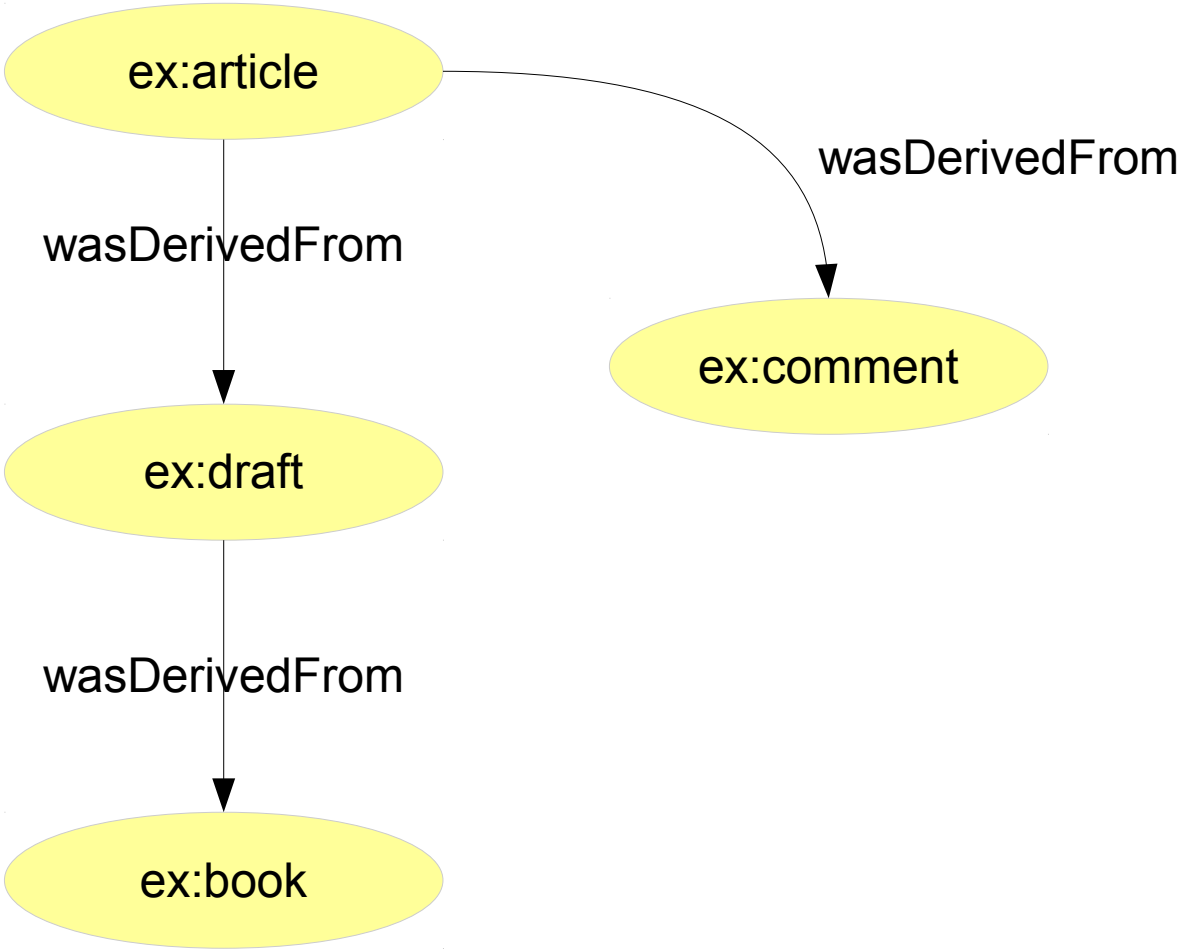
The program used a dataset to generate the table

Example: Entities

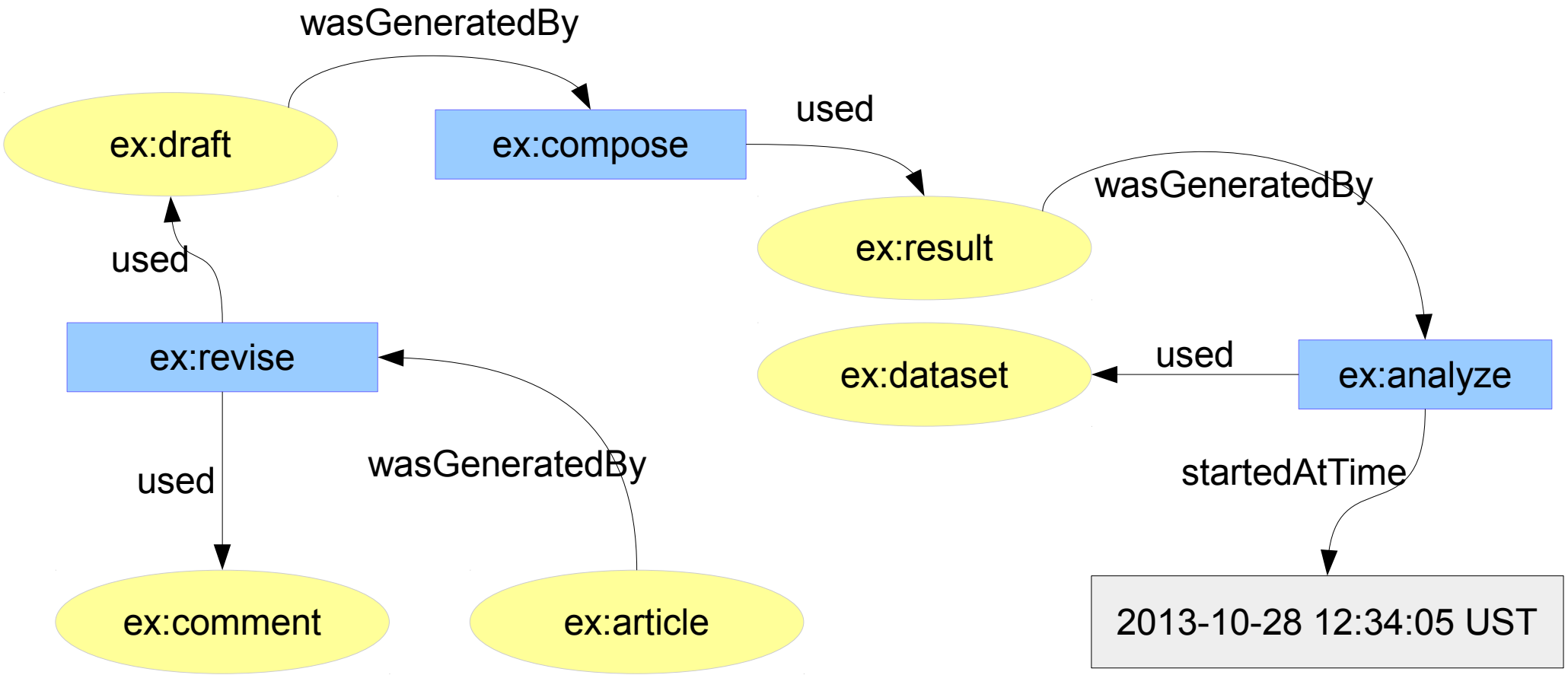



```
ex:draft   a prov:Entity ;
           a fabio:Manuscript ;
           dcterms:title "Latest results" .
ex:article a prov:Entity ;
           a fabio:ConferencePaper ;
           dcterms:title "Results from INV13a" .
ex:dataset a prov:Entity ;
           a fabio:Dataset .
ex:book    a prov:Entity ;
           a fabio:Thesis .
ex:result  a prov:Entity ;
           a fabio:Table .
ex:comment a prov:Entity ;
           a fabio:Review .
```

Example: Relation between Entities

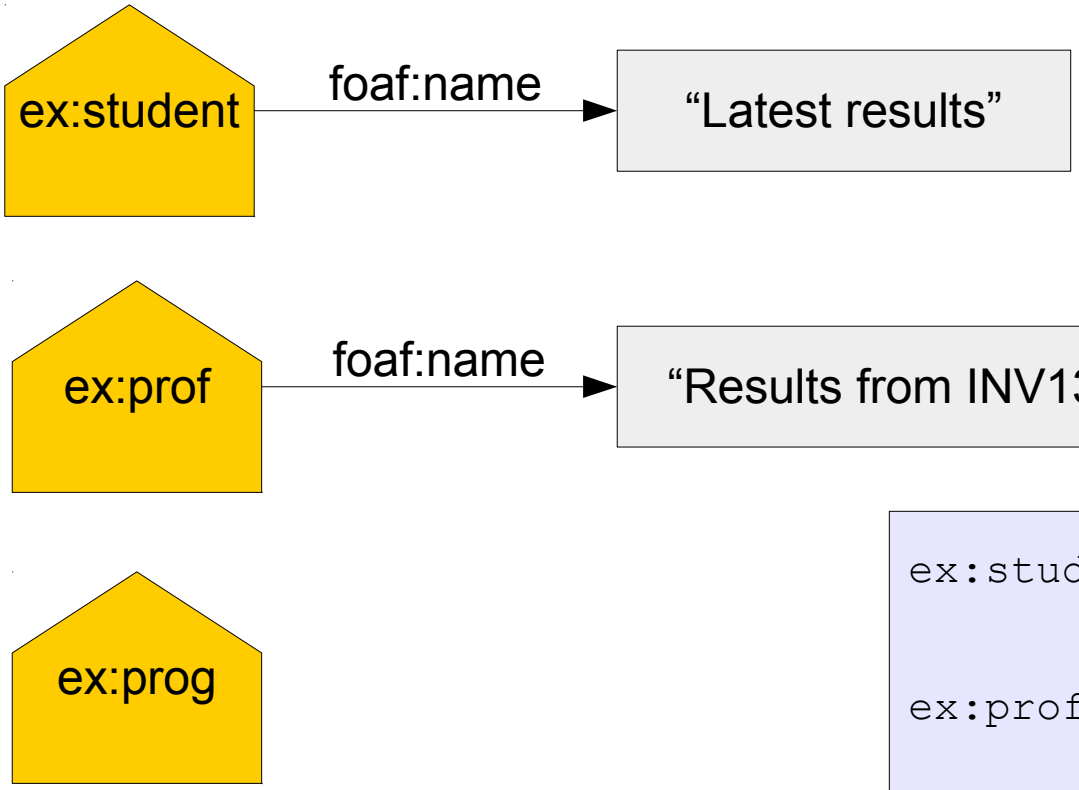


Example: Modelling the activities



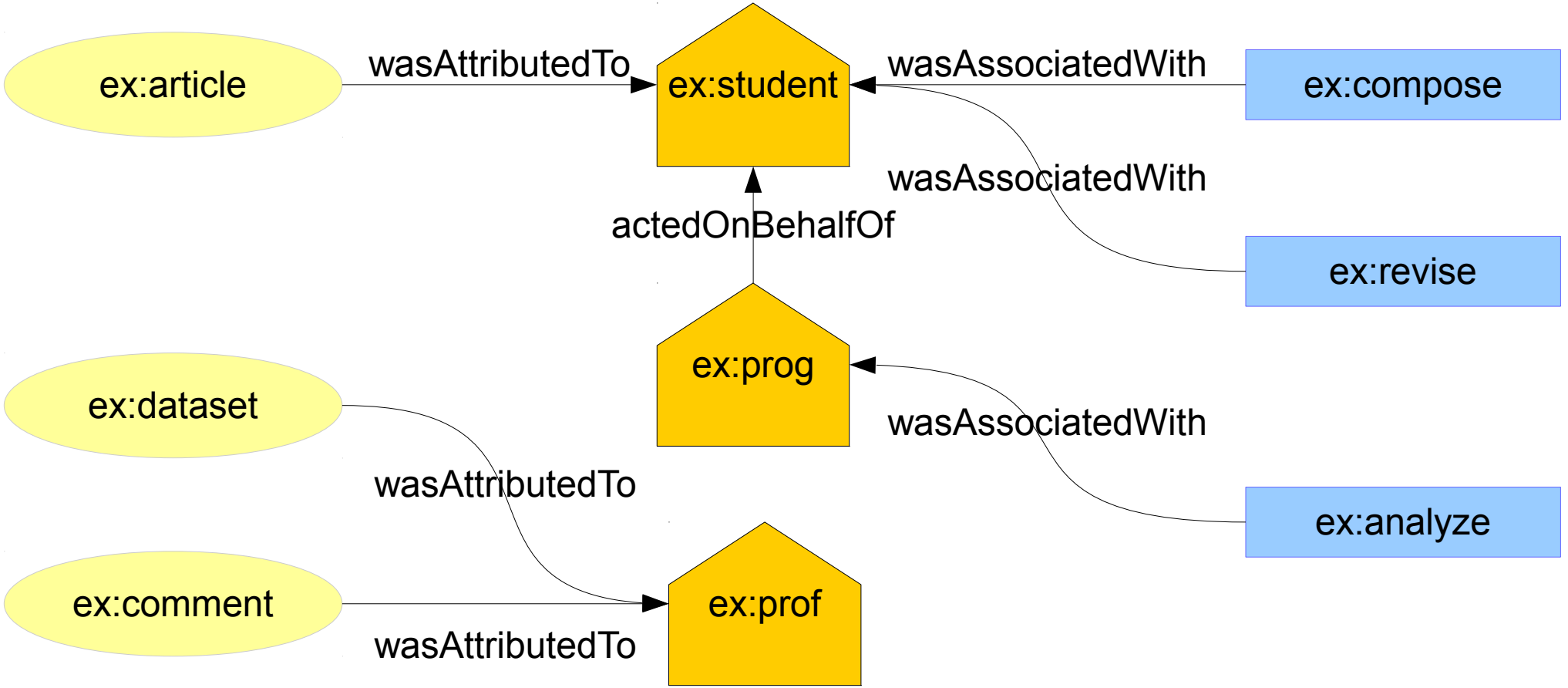

```
ex:compose a prov:Activity .
ex:revise  a prov:Activity .
ex:analyze a prov:Activity .
```

Example: Agents




```
ex:student a prov:Agent ;
           a foaf:Person ;
           foaf:name "Will Meyer" .
ex:prof    a prov:Agent ;
           a foaf:Person ;
           foaf:name "Joe Smith" .
ex:prog    a prov:Entity ;
           a fabio:Script .
```

Example: Attribution



Recap

PROV distinguishes

- Entities
- Activities
- Agents

Relations are

- Derivation of Entities from Entities
- Attribution of Entities to Agents
- Generation/Modification/Use of Entities by Activities
- Association of Agents to Activities

Extending the basic elements of PROV

Agents and Entities

The type of Agent can be specified through sub-properties

- prov:Person
- prov:Organization
- prov:SoftwareAgent

Same for type of Entity

- prov:Collection
- prov:Bundle
- prov:Plan

Types of Entities

prov:Collection

- Provides a structure to a group of Entities
  - prov:hadMember is used to describe membership
- Can be used to express the provenance of the collection itself

Types of Entities

prov:Bundle

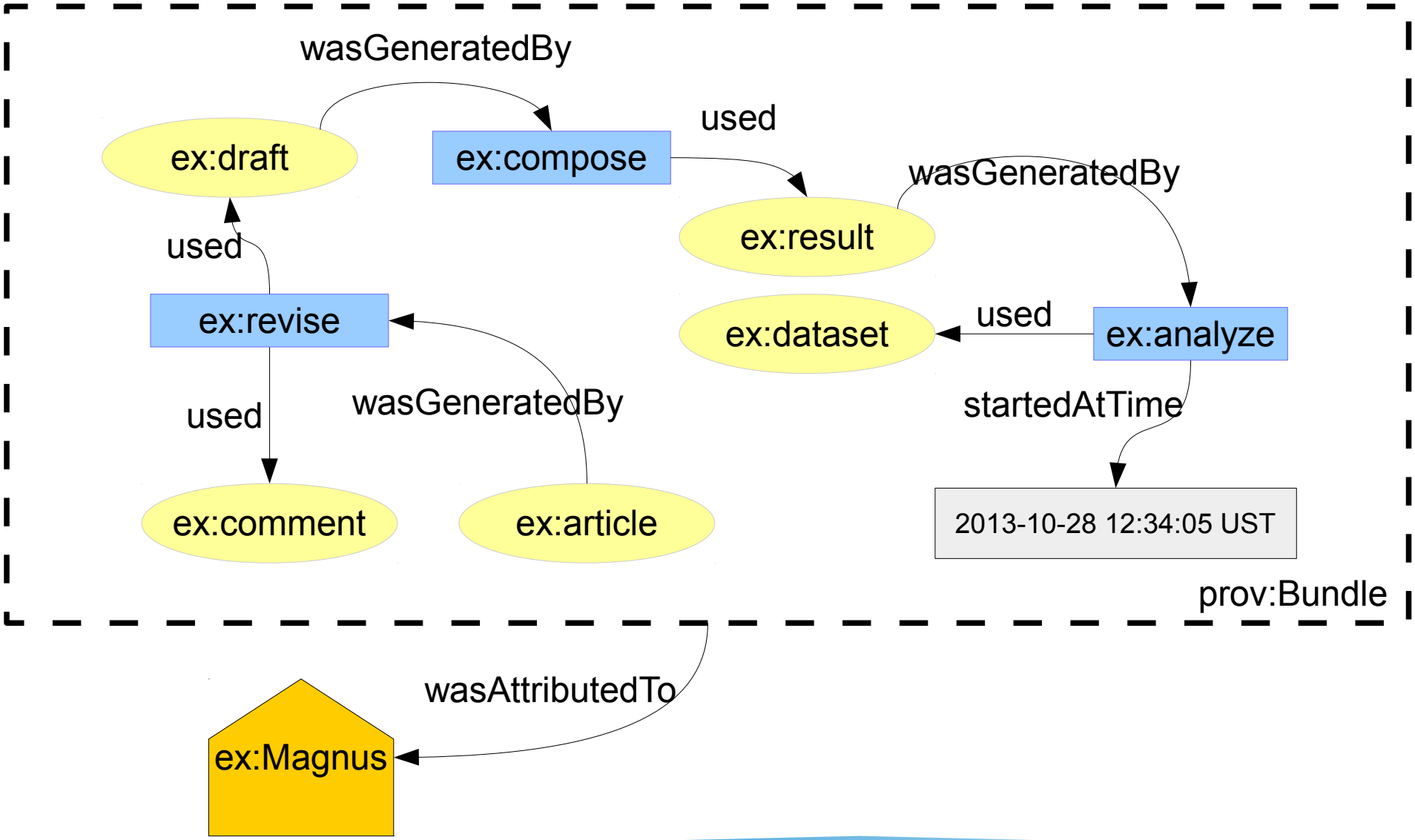
A named set of provenance descriptions, that itself can have provenance information associated with

No further subclasses provided – better left to other standards

prov:Plan

A set of actions done by (an) agent(s) to achieve a goal

Example: Bundle



Describing Entities

Entities can be described further by

prov:value

a literal value that represents an Entity

prov:Location

A geographic place

A non-geographic place such as a filesystem directory, URL, row in a table, …

Derivation

The type of derivation can be specified through sub-properties

prov:hadPrimarySource

Specific for first-hand reports, orignal works, etc.

prov:wasQuotedFrom

Specific for the extraction of a small part of the Entity

prov:wasRevisionOf

Relation between Entities

Relation between Entities can be further described

prov:specializationOf

Used to link a more specific Entity to a more general one

prov:alternateOf

Used to link Entities that present aspects of the same thing, but not necessarily the same aspects or at the same time

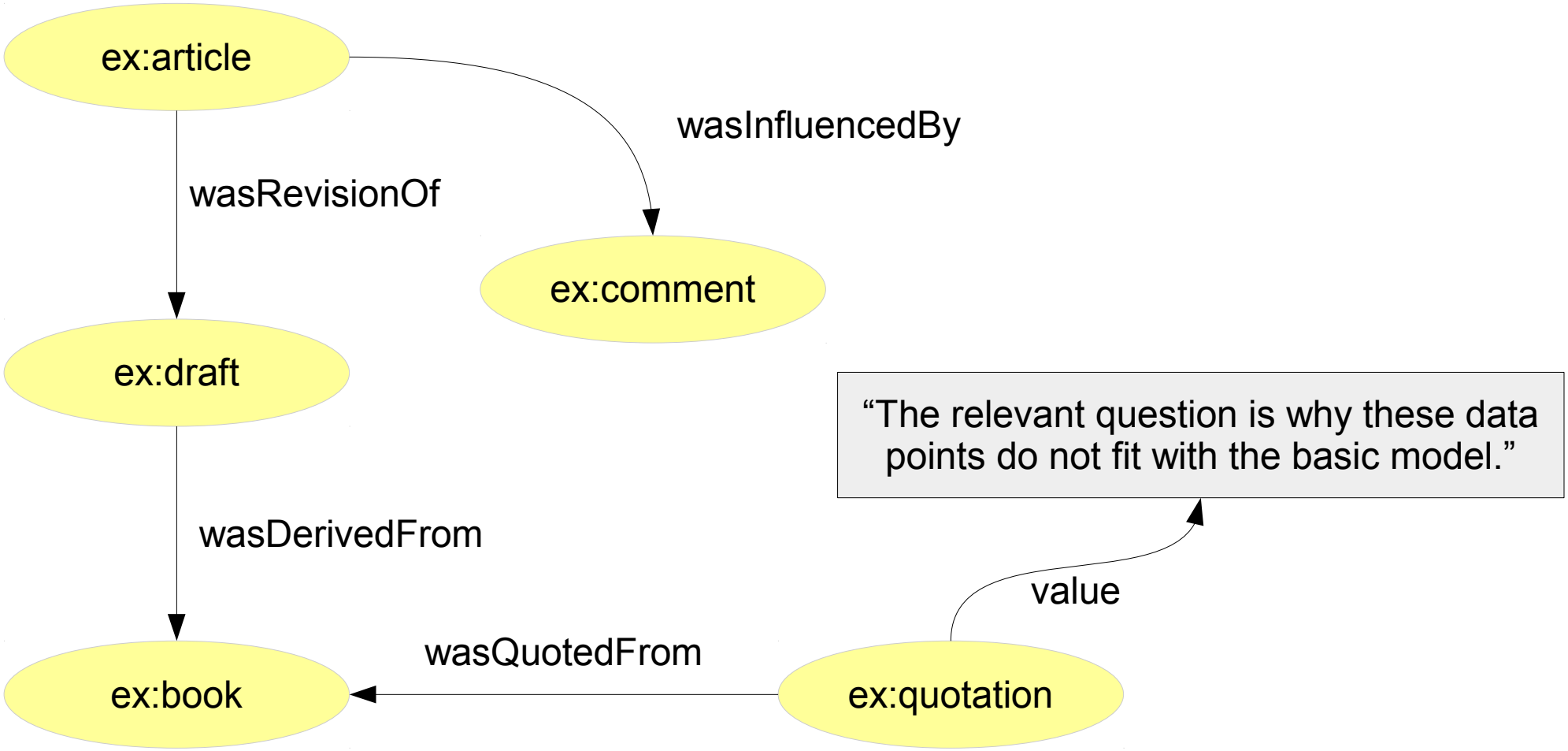
Broader Terms

A superproperty is introduced that relates any influenced Entity, Activity, or Agent to any other influencing Entity, Activity, or Agent that had an effect on its characteristics.

prov:wasInfluencedBy

But: The more specific properties should be used where possible

Example


ex:article
wasInfluencedBy
wasRevisionOf
ex:comment
ex:draft
“The relevant question is why these data points do not fit with the basic model.”
wasDerivedFrom
value
wasQuotedFrom
ex:book
ex:quotation

Lifetime of an Entity

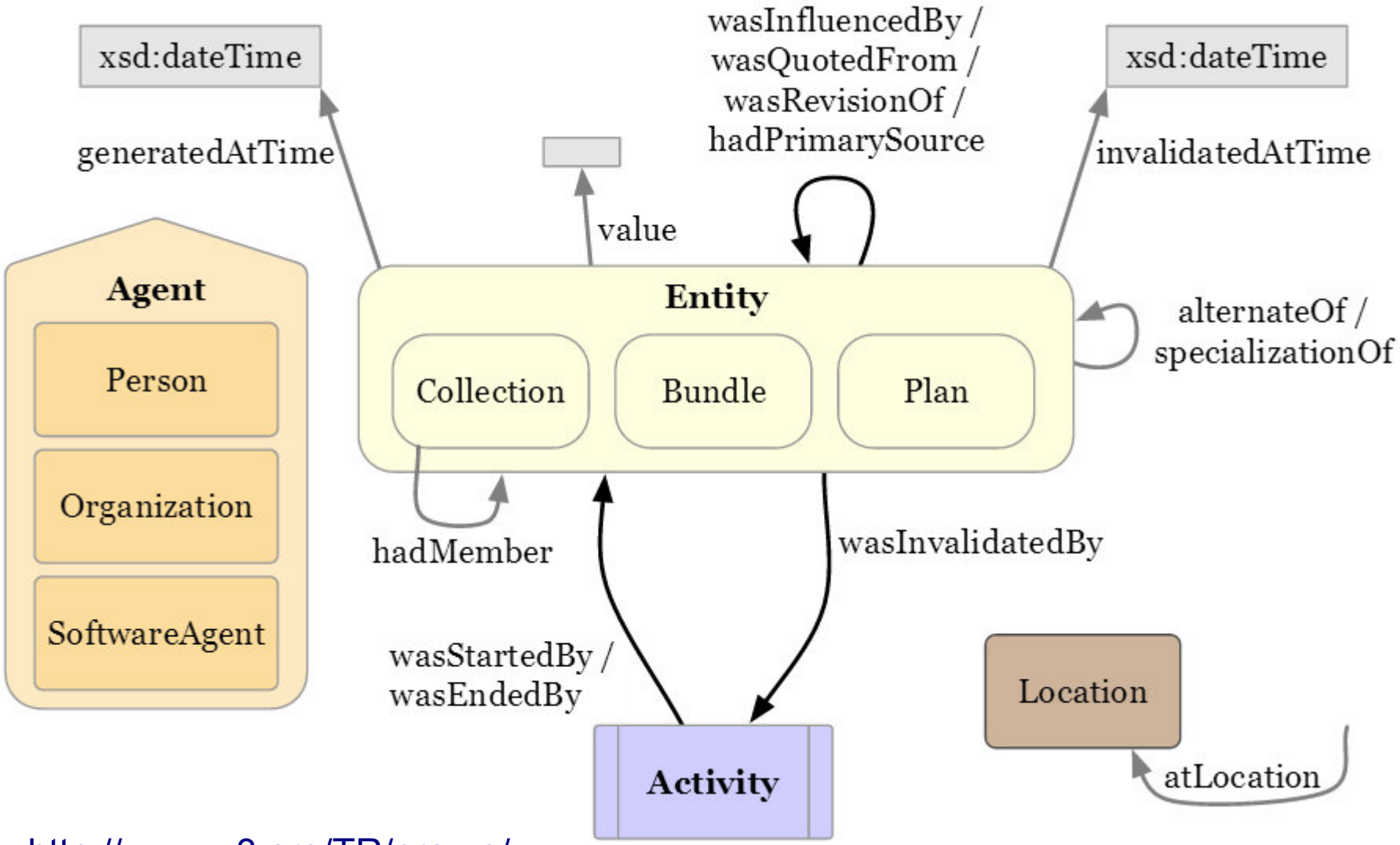
One can provide a starting and ending time of an Entity's existence

- prov:generatedAtTime
- prov:invalidatedAtTime

The involved Activities can be linked by

- prov:wasGeneratedBy / prov:generated
- prov:wasInvalidatedBy / prov:invalidated

Overview



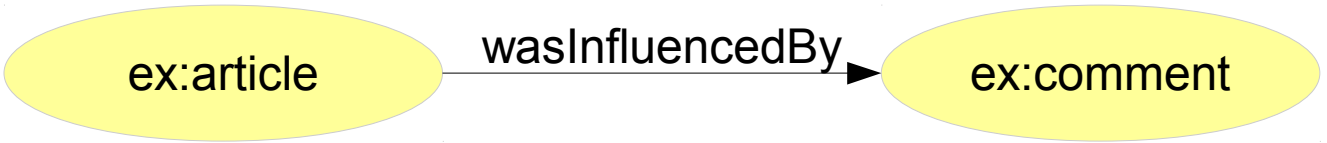
Source: http://www.w3.org/TR/prov-o/

Qualifying relations in PROV

Qualifying relations

Problem: Binary relations cannot be further elaborated

But one would like to describe aspects of the relation



e.g. the why, when, how, where of the influence between comment and article

The PROV solution

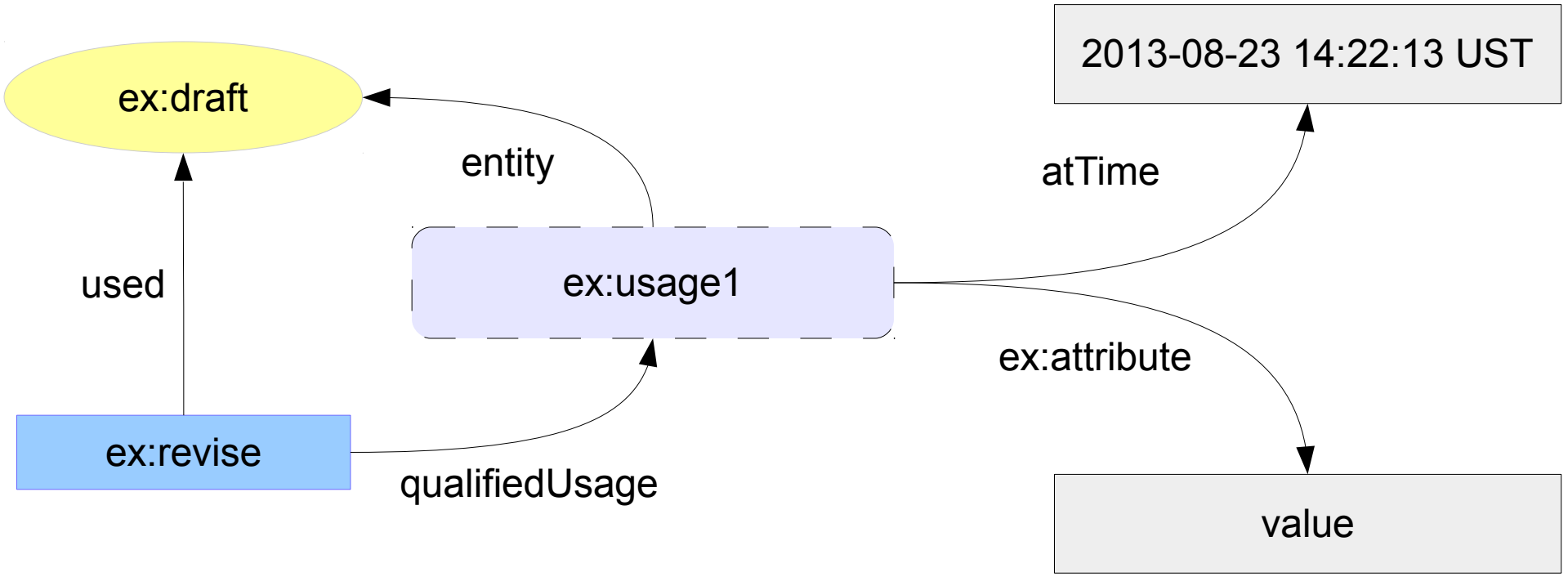
"All problems in computer science can be solved by another level of indirection"

(Attributed to David Wheeler, who apparently added: "But that usually will create another problem.")

Instead of using a binary relation, an intermediate class that represents the influence between two resources is used

This class can then be described by further attributes

Qualified Usage




```
ex:usage1 a prov:Usage ;
          prov:entity ex:draft ;
          prov:atTime "2013-08-23 14:22:13 UST"
          ex:attribute value .
```

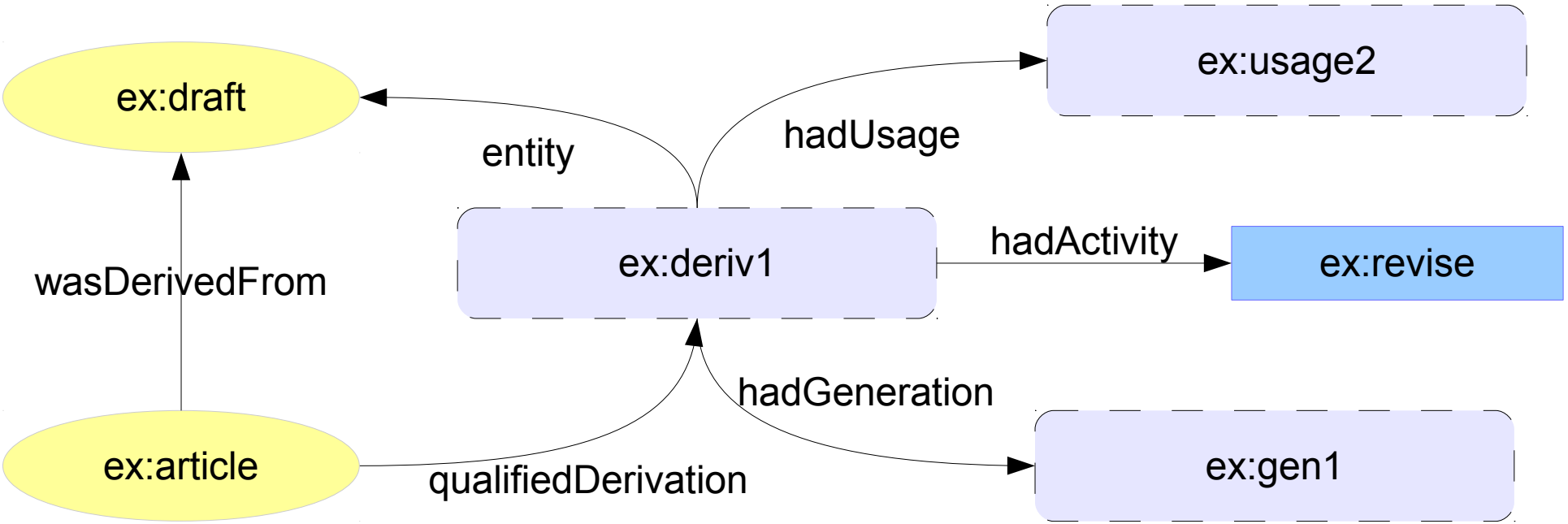
Qualified expressions

Influenced Class	Unqualified Influence	Influencing Class	Qualification Property	Qualified Influence	Influencer Property
Entity	wasGeneratedBy	Activity	qualifiedGeneration	Generation	activity
Entity	wasDerivedFrom	Entity	qualifiedDerivation	Derivation	entity
Entity	wasAttributedTo	Agent	qualifiedAttribution	Attribution	agent
Activity	used	Entity	qualifiedUsage	Usage	entity
Activity	wasInformedBy	Activity	qualifiedCommunication	Communication	activity
Activity	wasAssociatedWith	Agent	qualifiedAssociation	Association	agent
Agent	actedOnBehalfOf	Agent	qualifiedDelegation	Delegation	agent

Qualified expressions

Influenced Class	Unqualified Influence	Influencing Class	Qualification Property	Qualified Influence	Influencer Property
Entity Activity Agent	wasInfluencedBy	Entity Activity Agent	qualifiedInfluence	Influence	influencer
Entity	hadPrimarySource	Entity	qualifiedPrimarySource	PrimarySource	entity
Entity	wasQuotedFrom	Entity	qualifiedQuotation	Quotation	entity
Entity	wasRevisionOf	Entity	qualifiedRevision	Revision	entity
Entity	wasInvalidatedBy	Activity	qualifiedInvalidation	Invalidation	activity
Activity	wasStartedBy	Entity	qualifiedStart	Start	entity
Activity	wasEndedBy	Entity	qualifiedEnd	End	entity

Qualified Derivation




```
ex:deriv1 a prov:Derivation ;
          prov:entity ex:draft ;
          prov:hadActivity ex:revise ;
          prov:hadGeneration ex:gen1 ;
          Prov:hadUsage ex:usage2 .
```

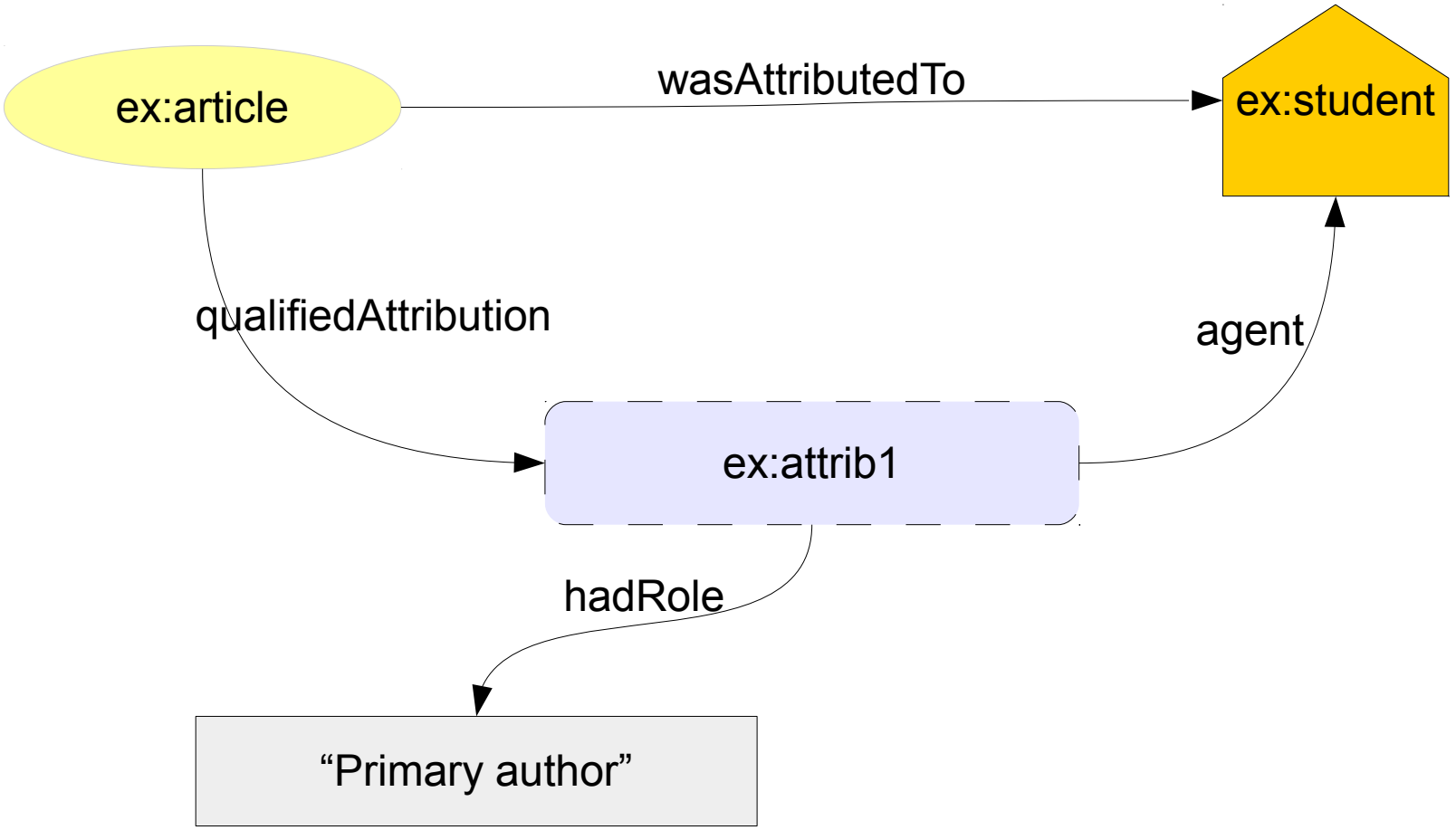
Roles

A role is the function of an entity or agent with respect to an activity, in the context of a usage, generation, invalidation, association, start, and end.

Class is prov:Role

Attribute is prov:hadRole

Qualified Attribution



Summary

Basic and extended PROV relations are unqualified

To qualify a relation

- An intermediate node is introduced
- There is a corresponding class for all relations
- The intermediate node can be described further

Special attributes exist to connect roles and activities

Mapping DC provenance information to PROV

Dublin Core

Remember: Many DC terms contain provenance information

- Who affected a resource
  - Creator, contributor, publisher, etc..
- How the resource was affected
  - Access rights, license, hasFormat, etc.
- When the resource was affected
  - Created, issued, dateSubmitted, etc.

Property ranges

Terms with dct:Agent as range

- creator
- contributor
- publisher
- rightsHolder

Property ranges

Terms with time as range

- available
- created
- date
- dateAccepted
- dateCopyrighted
- dateSubmitted
- issued
- modified
- valid

Property ranges

Terms with another resource as range

accessRights
hasFormat
hasVersion
isFormatOf
isVersionOf
license
isReferencedBy
isReplacedBy
references
replaces
rights
source

Direct mappings

Equivalences between PROV attributes and DC terms

Described in using

- rdfs:subClassOf
- rdfs:subPropertyOf
- owl:equivalentClass.

Direct mappings: DC Terms

DC Term	Mapping	PROV Property
created	subPropertyOf	generatedAtTime
dateAccepted	subPropertyOf	generatedAtTime
dateCopyRighted	subPropertyOf	generatedAtTime
dateSubmitted	subPropertyOf	generatedAtTime
issued	subPropertyOf	generatedAtTime
modified	subPropertyOf	generatedAtTime
creator	subPropertyOf	wasAttributedTo
contributor	subPropertyOf	wasAttributedTo
publisher	subPropertyOf	wasAttributedTo
rightsHolder	subPropertyOf	wasAttributedTo
source	subPropertyOf	wasDerivedFrom
hasFormat	subPropertyOf	alternateOf
isFormatOf	subPropertyOf	alternateOf, wasDerivedFrom

time

Agent

Direct mappings: Generalizations

Properties generalizing PROV terms

PROV property	Mapping	DC Term
hadPrimarySource	subPropertyOf	source
wasRevisionOf	subPropertyOf	isVersionOf

Classes generalizing PROV terms

PROV property	Mapping	DC Term
Location	subClassOfOf	LocationPeriodOrJurisdiction

Direct mappings: classes

DC Term	Relation	PROV Term
dct:Agent	owl:equivalentClass	prov:Agent
dct:BibliographicResource	rdfs:subClassOf	prov:Entity
dct:LicenseDocument	rdfs:subClassOf	prov:Entity
dct:LinguisticSystem	rdfs:subClassOf	prov:Plan
dct:Location	owl:equivalentClass	prov:Location
dct:MethodOfAccrual	rdfs:subClassOf	prov:Plan
dct:MethodOfInstruction	rdfs:subClassOf	prov:Plan
dct:RightsStatement	rdfs:subClassOf	prov:Entity
dct:PhysicalResource	rdfs:subClassOf	prov:Entity
dct:Policy	rdfs:subClassOf	prov:Plan
dct:ProvenanceStatement	rdfs:subClassOf	prov:Bundle

Complex mappings

Defined to generate *qualified* PROV statements from DC statements

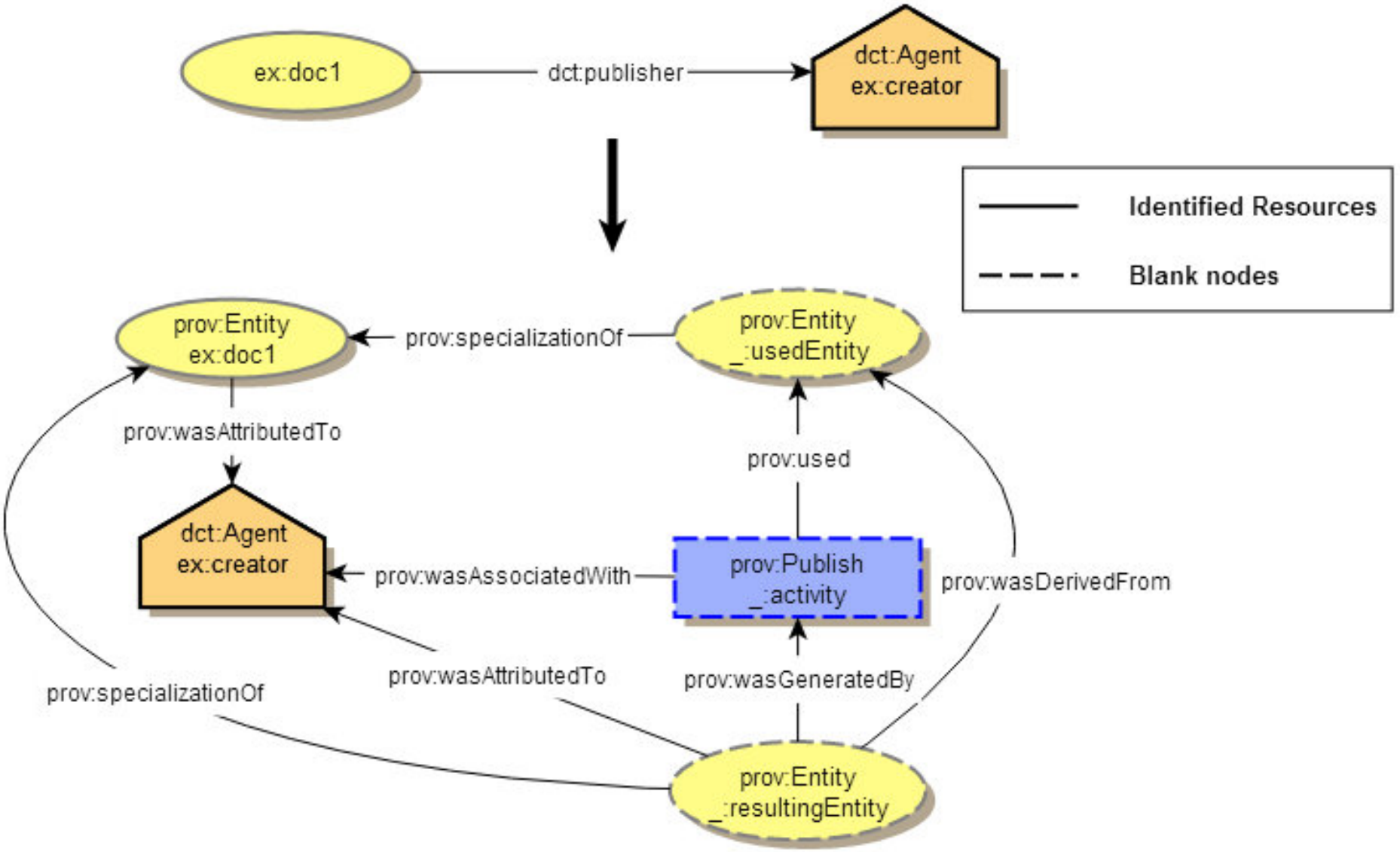
- Retain more information from the DC statements
- Can be adapted to include domain-specific elements
- Provided in the form of SPARQL construct queries

But: Need subclasses extending the base PROV classes to express the type of activity or role

PROV refinements: subclasses

Extended Term	Relation to PROV	PROV extended Term
Publish	subClassOf	Activity
Contribute	subClassOf	Activity
Create	subClassOf	Activity
RightsAssignment	subClassOf	Activity
Modify	subClassOf	Activity
Accept	subClassOf	Activity
Copyright	subClassOf	Activity
Submit	subClassOf	Activity
Replace	subClassOf	Activity
Publisher	subClassOf	Role
Contributor	subClassOf	Role
Creator	subClassOf	Role
RightsHolder	subClassOf	Role

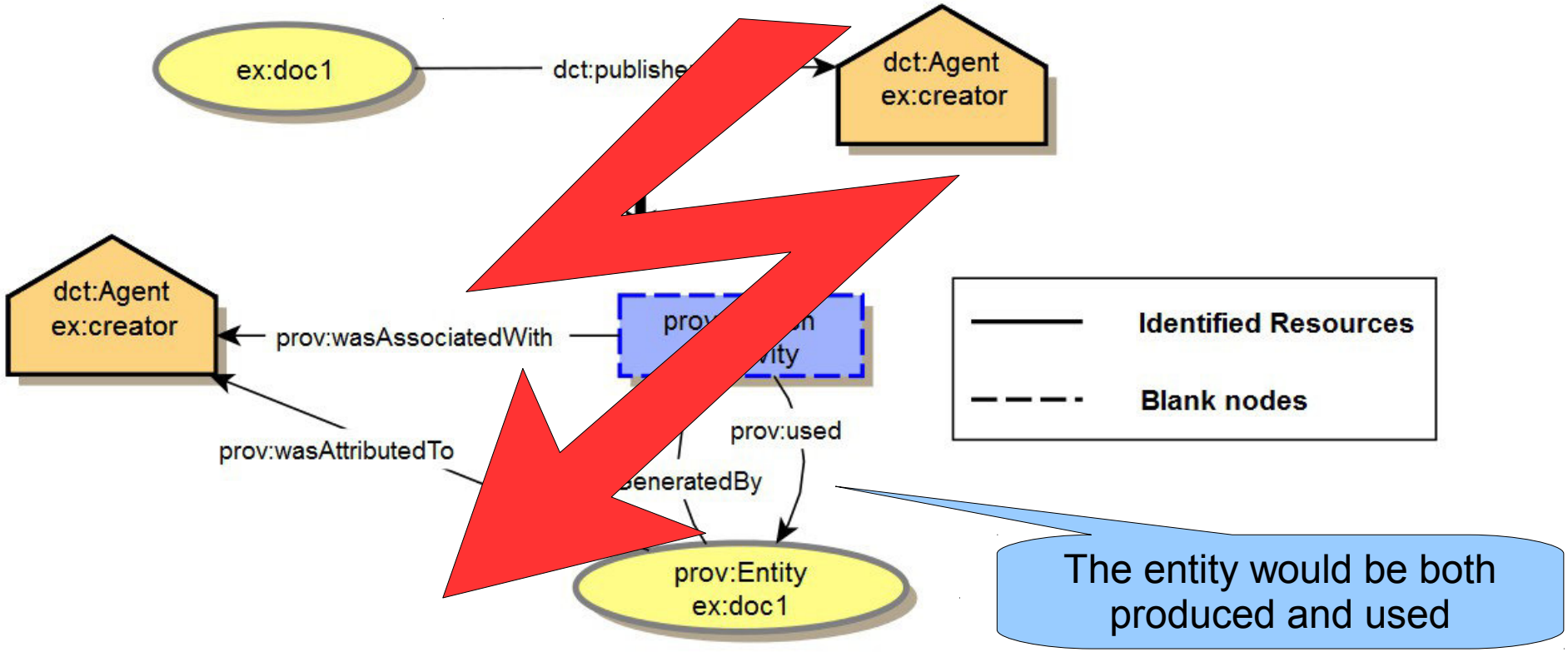
Complex mapping: Example



Source: http://www.w3.org/TR/2013/NOTE-prov-dc-20130430/

Complex mapping: Example

Is there no easier way?



Source: http://www.w3.org/TR/2013/NOTE-prov-dc-20130430/

```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
         a prov:Association;
         prov:agent ?agent;
         prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
         a prov:Association;
         prov:agent ?agent;
          prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

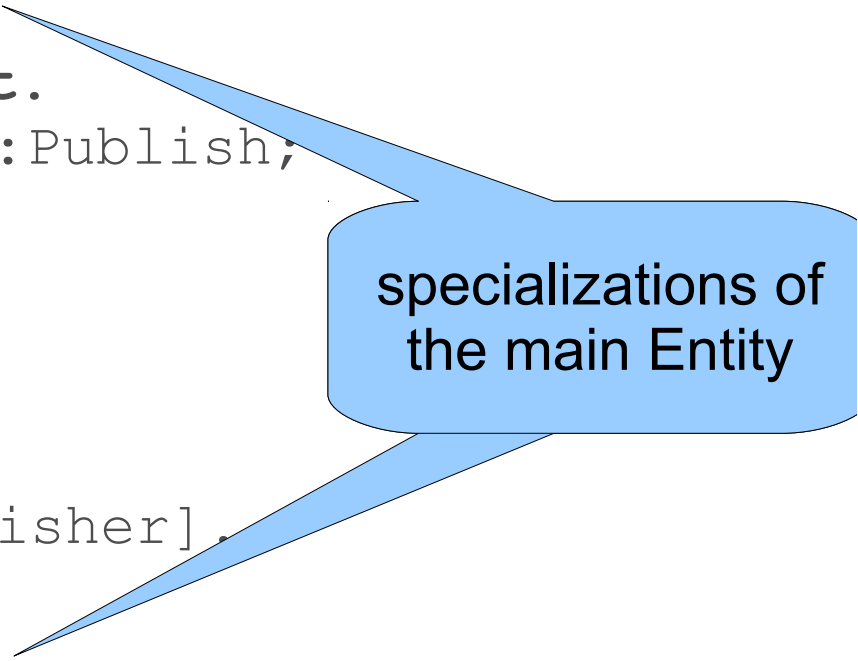
Translate the following type of statement

```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
         a prov:Association;
         prov:agent ?agent;
         prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

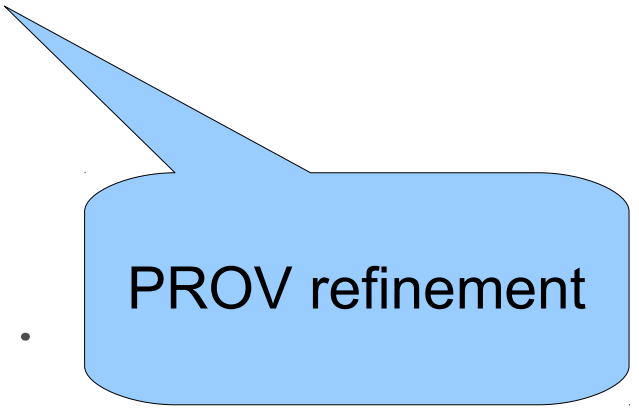
main Entity

direct mapping

```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
         a prov:Association;
         prov:agent ?agent;
         prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```



```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
         a prov:Association;
         prov:agent ?agent;
         prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```



```
CONSTRUCT {
 ?document a prov:Entity;
   prov:wasAttributedTo ?agent.
 ?agent a prov:Agent.
 _:usedEntity a prov:Entity;
   prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
   prov:used _:usedEntity;
   prov:wasAssociatedWith ?agent;
   prov:qualifiedAssociation [
        a prov:Association;
        prov:agent ?agent;
        prov:hadRole [a prov:Publisher].
   ].
 _:resultingEntity a prov:Entity;
   prov:specializationOf ?document;
   prov:wasDerivedFrom _:usedEntity;
   prov:wasGeneratedBy _:activity;
   prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

qualified association to bind activity to role

Complex mappings: Cleanup

The mappings produce many blank nodes

Ideas to reduce the blank nodes:

1. Conflate properties referring to the same state of the resource

    e.g. the terms publisher and issued

2. Sort all the activities according to their logical order and share intermediate blank nodes

    e.g. publication after creation

Summary

To convert existing provenance information in DC terms, a mapping to PROV-O is provided with the standard

It contains

- Direct mappings for terms and classes
- PROV-O Extensions for types of activities and roles
- Complex mappings to create full PROV-O provenance information

Thank you for listening.

Slides available online
http://www.slideshare.net/MagnusPfeffer/

References

PROV-O: The PROV Ontology

(W3C Recommendation)
http://www.w3.org/TR/prov-o/

PROV Model Primer

(W3C Working Group Note)
http://www.w3.org/TR/prov-primer/

This presentation is based on an earlier tutorial held at the SWIB2012 conference together with Kai Eckert.