

# **A Unified Approach for Representing Metametadata**

**Kai Eckert**  
**University of Mannheim**

**Dublin Core 2009**  
**Seoul, South Korea**  
**October 14<sup>th</sup> 2009**

## People

### **Kai Eckert**

Mannheim University  
kai@informatik.uni-mannheim.de

### **Magnus Pfeffer**

Mannheim University Library  
pfeffer@bib.uni-mannheim.de

### **Heiner Stuckenschmidt**

Mannheim University  
heiner@informatik.uni-mannheim.de



# What is this presentation about?

- Proposal for „**statements about statements**“ or „**metametadata**“ or „**provenance on statement level**“.
- *That was done before!*
- Implementation of metametadata by means of **RDF Reification**.
- *That was done before! (That's what Reification was made for.)*

### Anything new?

- Practical examples based on two scenarios and several use-cases.
- Both scenarios show (again) the need for metametadata.
- Easy implementation based on RDF and SPARQL.
- Standardization?

## RDF Reification

- RDF supports statements about statements by means of Reification, literally „objectification“ (actually a “subjectification” ...).
- “The book is written by Goethe” is said by Kai.  
Subject Predicate Object
- How is it done in RDF:

```
ex:someID rdf:type      rdf:Statement .  
ex:someID rdf:subject  "The book".  
ex:someID rdf:predicate ex:isWrittenBy .  
ex:someID rdf:object   "Goethe" .  
ex:someID ex:isSaidBy  "Kai" .
```

## Simplified Presentation

- Based on Notation 3 (RDF/N3)

```
Subject Predicate Object
1 ex:p123 rdf:type ex:person
2 ex:p123 ex:hasName "Kai Eckert"
3 ex:p123 ex:worksFor ex:unima
```

Example 1: A simple RDF example

- Identification of statements by the line number:

```
4 #1 dc:creator 'Kai Eckert'
```

The subject of a statement is a reference to another statement.  
With this notation, we imply a reification.

## Need for Metametadata

- Metadata are also data, so we need additional data about them. **→ Metametadata**
- Metadata about a whole metadata record, not for single statements:
  - Who created this metadata record?
  - When was this record created?
  - ...
- This exists: **→ Metadata Provenance**

### Statements about (single) statements

- Often proposed, but only vague instructions how to implement it.
- Needed, if metadata records are created by the combination of **single statements** from **different sources**.
- Needed for the storage of arbitrary **additional information for single statements**, that can not be represented in the metadata format easily.

## Scenario 1: Crosswalks

- Crosswalks define rules, how metadata from one schema are represented in a different schema.

| MARC field                                       |   | Dublin Core element |
|--|---|---------------------|
| 260\$c (Date of publication, distribution, etc.) | → | Date.Created        |
| 522 (Geographic Coverage Note)                   | → | Coverage.Spatial    |
| 300\$a (Physical Description)                    | → | Format.Extent       |

- Problems:
  - Loss of information
  - Erroneous Crosswalks

### Possibilities for Metametadata

- Storage of additional information, which would be lost in the target format.
- Identification of Crosswalks with version and the specific rule for every generated statement.

*Which statements are generated by a specific rule?*

*Which rule is responsible for a specific (erroneous) statement?*

*Which data in the originating format was used to generate a specific statement?*

## Example 1: Crosswalk Data

|    | <i>Subject</i>  | <i>Predicate</i> | <i>Object</i>           |
|----|-----------------|------------------|-------------------------|
| 1  | ex:docbase/doc1 | dc:title         | "Example title"         |
| 2  | #1              | ex:rule          | 16                      |
| 3  | #1              | ex:crosswalk     | 3                       |
| 4  | #1              | ex:origin        | MARC:245                |
| 5  | ex:docbase/doc2 | dc:title         | "About finding a title" |
| 6  | #5              | ex:rule          | 16                      |
| 7  | #5              | ex:crosswalk     | 3                       |
| 8  | #5              | ex:origin        | MARC:245                |
| 9  | ex:docbase/doc3 | dc:title         | "Lorem ipsum dolor"     |
| 10 | #9              | ex:rule          | 18                      |
| 11 | #9              | ex:crosswalk     | 3                       |
| 12 | #9              | ex:origin        | MARC:245                |
| 13 | #9              | ex:origin        | MARC:246                |
| 14 | ex:docbase/doc4 | dc:title         | "Consetetur Sadipscing" |
| 15 | #14             | ex:rule          | 19                      |
| 16 | #14             | ex:crosswalk     | 6                       |
| 17 | #14             | ex:origin        | xml:/record/description |

Example 4: Resulting RDF statements with additional Metametadata

## Crosswalk Updates

- Which statements are generated by a given rule and need to be regenerated after an update?

```
SELECT ?document ?field ?value WHERE {  
  ?t rdf:subject      ?document .  
  ?t rdf:predicate    ?field .  
  ?t rdf:object       ?value .  
  ?t ex:rule          16 .  
  ?t ex:crosswalk     3 .  
}
```

| document        | field                           | value                |
|-----------------|---------------------------------|----------------------|
| ex:docbase/doc1 | http://www.example.org/dc#title | "Example title"      |
| ex:docbase/doc2 | http://www.example.org/dc#title | "About ding a title" |

## Crosswalk Debugging

- Which rule is responsible for a given statement and what was the original data?

```
SELECT ?crosswalk ?rule ?origin WHERE {  
  ?t rdf:subject      <ex:docbase/doc1> .  
  ?t rdf:predicate    dc:title .  
  ?t rdf:object       "Example title" .  
  ?t ex:rule          ?rule .  
  ?t ex:crosswalk     ?crosswalk .  
  ?t ex:origin        ?origin .  
}
```

| crosswalk | rule | origin     |
|-----------|------|------------|
| 3         | 16   | "MARC:245" |

### Scenario 2: Different Sources for Metadata

- Manual indexing is costly.
- Many documents are not indexed at all or not searchable:
  - Journal Articles
  - Externally owned documents
  - Working papers
  - Webpages
- **New sources for metadata?**

# New ways for document indexing

- Automatic processes
- Tagging
- (Automatic) mapping of metadata from external sources
- **Problem: Lack of quality**

*How do you integrate these data from different sources without compromising the retrieval quality?*

### Possibilities for Metametadata

- Storage of the source of single statements.
- Storage of further source-specific information:
  - Weighting for automatically generated subject headings.
  - Number of users who tagged a document with a given tag.
  - The original subject heading in case of an automatic translation or mapping.

*Can we use these additional information to improve document retrieval?*

## Example 2: Subject indexing

|    | <i>Subject</i>        | <i>Predicate</i> | <i>Object</i>         |
|----|-----------------------|------------------|-----------------------|
| 1  | ex:docbase/doc1       | dc:subject       | ex:thes/sub20         |
| 2  | #1                    | ex:source        | ex:sources/autoindex1 |
| 3  | #1                    | ex:rank          | 0.55                  |
| 4  | ex:docbase/doc1       | dc:subject       | ex:thes/sub30         |
| 5  | #4                    | ex:source        | ex:sources/autoindex1 |
| 6  | #4                    | ex:rank          | 0.8                   |
| 7  | ex:docbase/doc1       | dc:subject       | ex:thes/sub30         |
| 8  | #7                    | ex:source        | ex:sources/pfeffer    |
| 9  | #7                    | ex:rank          | 1.0                   |
| 10 | ex:docbase/doc1       | dc:subject       | ex:thes/sub40         |
| 11 | #10                   | ex:source        | ex:sources/pfeffer    |
| 12 | #10                   | ex:rank          | 1.0                   |
| 13 | ex:sources/autoindex1 | ex:type          | ex:types/auto         |
| 14 | ex:sources/pfeffer    | ex:type          | ex:types/manual       |

Example 7: Subject assignments by different sources

## Backward compatibility

- While there are four assignments for subject headings, the statement  
“`ex:docbase/doc1 dc:subject ex:thes/sub30`”  
is still one statement, regardless of the number of times you put it into your RDF store.
- Important for applications, that access the RDF Data, but do not handle the RDF reification.
- **Your metadata remains valid, in particular there are no doublets.**

## Separating the sources

- Which statements are made by a specific source (here: Pfeffer)?

```
SELECT ?document ?value WHERE {  
  ?t rdf:subject      ?document .  
  ?t rdf:predicate    dc:subject .  
  ?t rdf:object       ?value .  
  ?t ex:source        <ex:sources/pfeffer> .  
}
```

| <b>document</b> | <b>subject</b> |
|-----------------|----------------|
| ex:docbase/doc1 | ex:thes/sub30  |
| ex:docbase/doc1 | ex:thes/sub40  |

## Extended queries

- Use all manually created subject headings.
- Use all subject headings with a rank > 0.7.

```
SELECT DISTINCT ?document ?subject WHERE {  
  ?t rdf:subject      ?document .  
  ?t rdf:predicate   dc:subject .  
  ?t rdf:object      ?subject .  
  ?t ex:source ?source .  
  ?source ex:type      ?type .  
  ?t ex:rank      ?rank .  
  FILTER (  
    ?type = <ex:types/manual> || ?rank > 0.7  
  )  
}
```

| document        | subject       |
|-----------------|---------------|
| ex:docbase/doc1 | ex:thes/sub30 |
| ex:docbase/doc1 | ex:thes/sub40 |

# Conclusion

- Many applications of metametadata in the library fields can be realized with RDF Reification.
- No need for the reinvention of the wheel, based on existing standards and recommendations.
- With SPARQL and SeRQL you can access and make use of the additional information.
- Lot of mature products - developed by the semantic web community or commercial ones.

### Standardization?

- Would be great for generally accepted use-cases.
- What are “generally accepted use-cases”?
- Metametadata we used so far:
  - A rank for subject headings
  - Qualifications for sources (automatic, manual, expert, layman, ...)
  - Link to further information about a specific indexing process
  - Link to original data source
  - Debugging information