# OpenDocument meta data

Florian Reuter

# What is RDF?

- Very roughly: Triples of
  - Subject
  - Predicate
  - Object
- How are RDF and XML linked?
  - RDF/XML coding
- Please note the difference between RDF (subject, predicate, object) and it's RDF/XML encoding.

# RDF and OpenDocument meta data?

- Question:
  How do the OpenDocument meta data map into the RDF world; i.e. what are the (subject, predicate, object) triples derived from the OpenDocument meta data?

# RDF and OpenDocument meta data?

- What is the subject of an OpenDocument meta datum?
  - The current document
- What is the predicate of an OpenDocument meta datum?
  - The meta XML element.
- What is the object of an OpenDocument meta datum?
  - The meta XML elements value.

# Mapping Dublin core elements

- Office meta datum:

```
<office:meta>
  <dc:date>2005-09-05T16:06:46</dc:date>
<office:meta>
```

- RDF triple:

(_:currentDocument, dc:date, “2005-09-05T16:06:46”)

- RDF/XML equivalent:

```
<rdf:Description rdf:nodeID="currentDocument">
  <dc:date>2005-09-05T16:06:46</dc:date>
</rdf:Description>
```

# Mapping DC elements (2nd Sample)

- Office meta datum:

```
<office:meta>
  <dc:language>en-us</dc:language>
<office:meta>
```

- RDF triple:

(_:currentDocument, dc:language, “en-us”)

- RDF/XML equivalent:

```
<rdf:Description rdf:nodeID="currentDocument">
  <dc:language>en-us</dc:language>
</rdf:Description>
```

# Mapping Dublin core elements

- Similar all the following DC elements defined in office:meta can be mapped into RDF:
  - <dc:creator>
  - <dc:date>
  - <dc:description>
  - <dc:language>
  - <dc:subject>
  - <dc:title>

# Mapping office:meta elements

- Office meta datum:

```
<office:meta>
  <meta:creation-date>2005-09-05</meta:creation-date>
<office:meta>
```

- RDF triple:

(_:currentDocument, meta:creation-date, 2005-09-05)

- RDF/XML equivalent:

```
<rdf:Description rdf:nodeID="currentDocument">
  <meta:creation-date>2005-09-05</meta:creation-date>
</rdf:Description>
```

# Mapping meta:* elements

- The following meta:* elements can be mapped in the previously shown way:
  - <meta:creation-date>
  - <meta:editing-cycles>
  - <meta:editing-duration>
  - <meta:generator>
  - <meta:initial-creator>
  - <meta:keyword>
  - <meta:print-date>
  - <meta:printed-by>

# Mapping meta:* elements

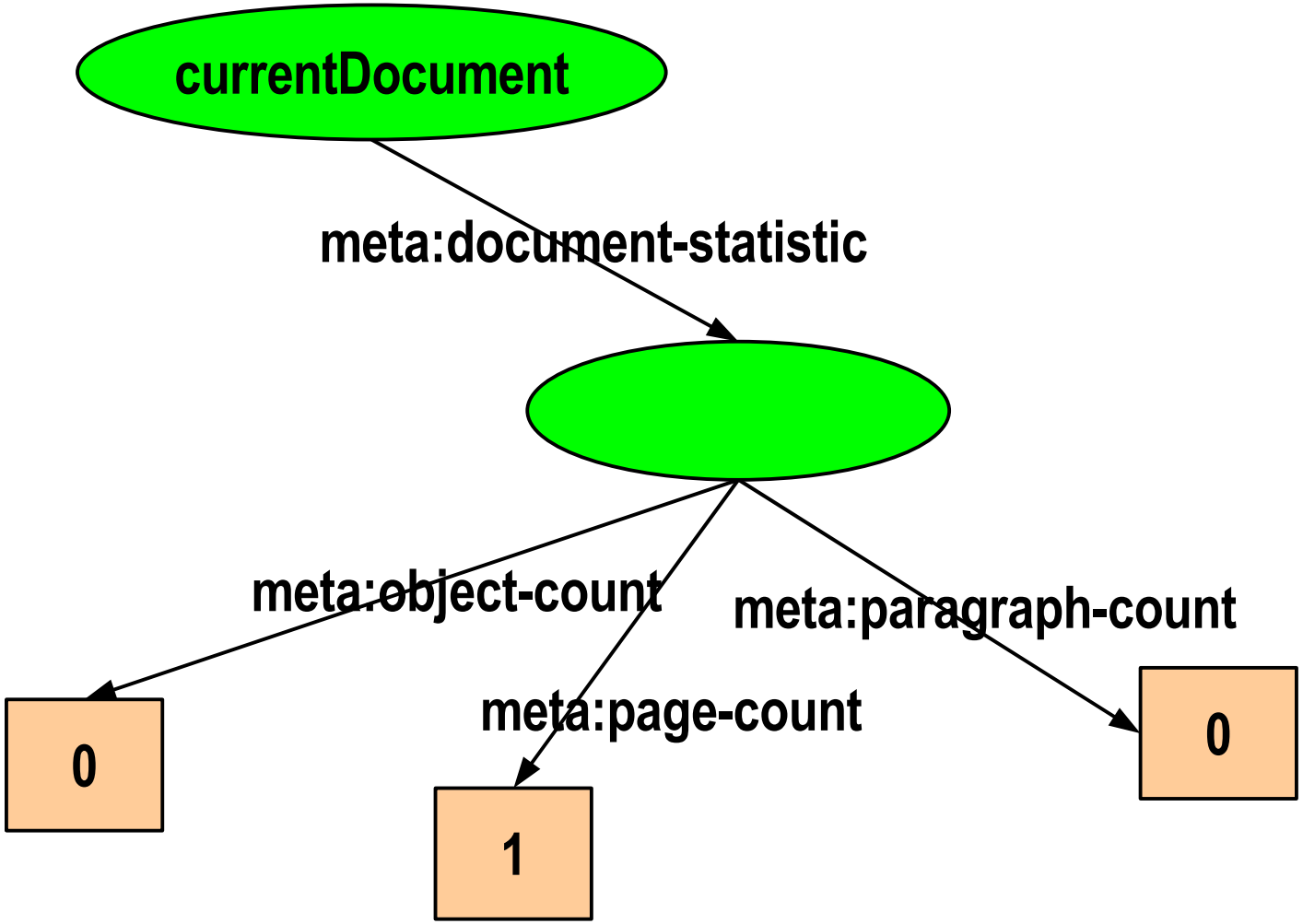
- The following meta:* elements need special mapping:
  - <meta:auto-reload>
  - <meta:document-statistic>
  - <meta:hyperlink-behaviour>
  - <meta:template>
  - <meta:user-defined>

# Mapping meta:document-statistic:

- The following meta:* elements need special mapping:
- Sample

```
<office:meta>
   <meta:document-statistic
      meta:object-count="0"
      meta:page-count="1"
      meta:paragraph-count="0" />
<office:meta>
```

# Mapping meta:document-statistic:



# Mapping meta:document-statistic:

- RDF/XML encoding:

```
<rdf:Description rdf:nodeID="currentDocument">
  <meta:document-statistic rdf:nodeID="currentStatistic"/>
</rdf:Description>
<rdf:Description rdf:nodeID="currentStatistic">
  <meta:object-count>0</meta:object-count>
  <meta:page-count>1</meta:page-count>
  <meta:paragraph-count>0<meta:paragraph-count>
</rdf:Description>
```

# Mapping meta:* elements

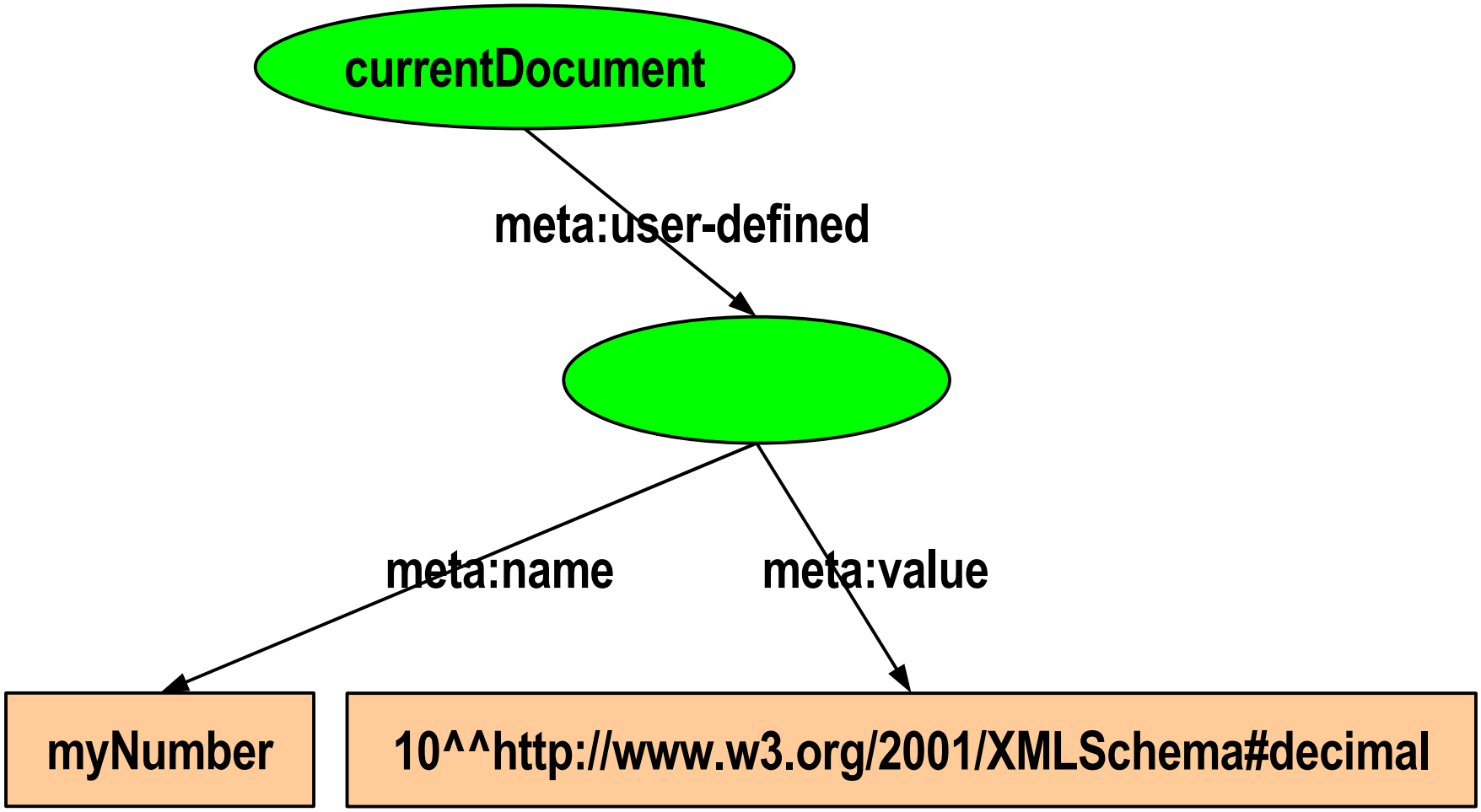
- The following meta:* elements can be handled by introducing an additional anonymous node (as illustrated previously):
  - <meta:auto-reload>
  - <meta:document-statistic>
  - <meta:hyperlink-behaviour>
  - <meta:template>

# meta:user-defined elements

- meta:user-defined elements can be mapped using RDFs type mechanism:
- Sample

```
<meta:user-defined meta:name="myNumber" meta:value-type="float">10</meta:user-defined>
```

# meta:user-defined elements



# meta:user-defined elements

- RDF/XML encoding:

```
<rdf:Description rdf:nodeID="currentDocument">
  <meta:document-statistic rdf:nodeID="pair01"/>
</rdf:Description>
<rdf:Description rdf:nodeID="pair">
  <meta:name>myNumber</meta:name>
  <meta:value
   rdf:datatype="http://www.w3.org/2001/XMLSchema#deci
   mal">10</meta:value>
</rdf:Description>
```

# Summary

- For every OpenDocument meta datum a formal RDF semantic can be given.
- Thus the OpenDocument meta data seamlessly integrates in the RDF world.
- Thus, by defining a RDF mapping for the OpenDocument meta data we support RDF.

# New meta data elements

- For new meta data elements we may adopt the RDF/XML syntax for predicate and objects.
- The subject of meta data defined as a child of <office:meta> is always “the current document”.
- Formal:

```
<define name="office-meta-data" combine="choice">
    <ref name="anything, which can be a child of rdf:Desciption"/>
</define>
```

- Good idea?

# Generic meta data

- In order to allow RDF statements about other subjects as the current document we may integrate a <office:rdf> elements, which can contain generic RDF/XML encoded triples.
- Sample

```
<office:document>
  <office:rdf>
    <rdf:Description rdf:about="...">...</rdf:Description>
  </office:rdf>
</office:document>
```

- Good idea?

# What about XMP?

- XMP support for OpenDocument means:
  - The XMP SDK maps the OpenDocument meta data into the RDF model as defined previously.
  - The XMP SDK additionally reads the <office:rdf> stream for generic meta data.