Some of the newest SW technologies at W3C: RDFa1.1 and R2ML

Benelux Semantic Web Meetup, Amsterdam
2010-11-24
Ivan Herman, W3C
RDFa
What is RDFa?

- For RDF people, it *sounds* very simple:
  - RDFa is a serialization of RDF embedded in XHTML, HTML, or XML in general
So why bother? Why should we care? Why is that of any importance?
RDFa may become the single biggest source of RDF triples on the Web after direct database access!
Data for a Web of Data

- Apart from relational databases, most of the data on the Web are in... (X)HTML content
- New content is generated every day
- How would one get structured data from that information?
Authors of the “traditional Web”...

- Do not generate RDF/XML files separately
  - RDF/XML is complex
  - it requires a separate storage, generation, etc mechanism
    - that is also valid for, e.g., Turtle
    - even when authoring with, say, Emacs, creating an extra file is a load
Solution

- Add extra structured content to the (X)HTML pages
- Let processors extract those and turn into RDF
Existing approaches

- **Microformats**
  - reuses HTML attributes like @class, @title
  - separate vocabularies (address, CV, …)
  - difficult to mix microformats (no concept of namespaces)
  - possible to transform via, e.g., XSLT + GRDDL, but all transformations are vocabulary dependent
Existing approaches

- **Microdata**
  - adds new attributes to HTML5 to express metadata
  - can use URI-s, it also fixes some vocabulary mappings (e.g., to Dublin Core elements)
  - works for simpler usages, not well suited for complex vocabularies
  - has no notion of datatypes, namespaces
  - but generic processing becomes possible to generate RDF
Existing approaches

- **RDFa**
  - adds new (X)HTML/XML attributes
  - has namespaces and URIs at its core; i.e., mixing vocabulary is just as easy as in RDF
  - complete flexibility for using Literals or URI Resources
  - *is a complete serialization of RDF*
  - generic processing becomes possible to generate RDF
RDFa is a complete bridge between the Web of Documents and the Web of Data
Therefore...

- It is very important for RDF experts to
  - know RDFa
  - parse it alongside Turtle, RDF/XML or other
  - when appropriate, generate RDFa pages
What does RDFa look like?
Main principles of RDFa

- RDFa means “RDF in attributes”. I.e.,:
  - all RDF contents are defined through XML attributes (no elements)
  - the XML/HTML tree structure is used
  - many of the attributes are defined by RDFa
    - some attributes (@href, @rel) are also reused
  - if possible, the text content is also reused (for literals) as well as @href values
What does this mean in practice?

- The same (X)HTML file:
  - is used, unchanged, by browsers
    - they ignore attributes they do not know
  - can be used by specialized processors (or APIs) to extract RDF triples
Before getting into details...

- The current Recommendation is RDFa 1.0
- There is an RDFa1.1 in the making, almost ready
- *I will talk about RDFa1.1* and warn when the feature is not available in RDFa1.0
XML or (X)HTML?

- Formally:
  - RDFa WG defines Core and XHTML
  - HTML WG defines HTML5
- we will use XHTML examples
A typical usage pattern

- A browser usually asks for an HTML content:

![Image of a browser with URI in the address bar](image)

Unique identifier for RDFS Entailment.

“http://www.w3.org/ns/entailment/RDFS” is the URI. The specification for the RDFS entailment is part of the RDF Semantics W3C Recommendation.

For more information about RDF, please refer to the the RDF Concepts and Abstract Syntax Recommendation.

Ivan Herman, ivan@w3.org, W3C, Semantic Web Activity Lead, 2009-05-03
A typical usage pattern

- Via content negotiations this goes to:

![Image of web browser with URI](image)

**W3C Semantic Web**

Unique identifier for **RDFS Entailment**.

"http://www.w3.org/ns/entailment/RDFS" is the URI. The specification for the RDFS entailment is part of the **RDF Semantics** W3C Recommendation.

For more information about RDF, please refer to the **the RDF Concepts and Abstract Syntax Recommendation**.

Ivan Herman, ivan@w3.org, W3C, Semantic Web Activity Lead, 2009-05-03
A typical usage pattern

- Via content negotiations this goes to:

```
http://www.w3.org/ns/entailment/data/RDFS.html
```

Unique identifier for RDFS Entailment.

"http://www.w3.org/ns/entailment/RDFS" is the URI. The specification for the RDFS entailment is part of the RDF Semantics W3C Recommendation.

For more information about RDF, please refer to the RDF Concepts and Abstract Syntax Recommendation.

Ivan Herman, ivan@w3.org, W3C, Semantic Web Activity Lead, 2009-05-03
A typical usage pattern

- But a client could ask for, say, Turtle:

```turtle
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ent:RDFS a ent:Entailment ;
dc:creator <http://www.ivan-herman.net/foaf#me> ;
dc:date "2009-05-03" ;
dc:description "Unique identifier for RDFS Entailment" ;
rdfs:comment "The specification for the RDFS entailment is part of the RDF Semantics W3C Recommendation." ;
rdfs:isDefinedBy <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/#rdfs_entailment> ;
rdfs:seeAlso <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> .

<http://www.w3.org/ns/entailment/data/RDFS.html> dc:title "Information Resource RDFS Entailment" ;
xhv:stylesheet <http://www.w3.org/StyleSheets/TR/base> .
```
Where does the Turtle content come from?

- The triples are embedded in the HTML file
  - a client may know how to extract RDF triples directly from that file; or
  - an online “distiller” service is used; or
  - the server is set up to generate the Turtle file automatically
For example, use an online service...

```reasonml
@prefix dc: <http://purl.org/dc/terms/> .
@prefix ent: <http://www.w3.org/ns/entailment/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xhv: <http://www.w3.org/1999/xhtml/vocab#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ent:RDFS a ent:Entailment ;
   dc:creator <http://www.ivan-herman.net/foaf#me> ;
   dc:date "2009-05-03" ;
   dc:description "Unique identifier for RDFS Entailment" ;
   rdfs:comment "The specification for the RDFS entailment is part of the RDF Semantics W3C Recommendation." ;
   rdfs:isDefinedBy <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/#rdfs_entailment> ;
   rdfs:seeAlso <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> .

<http://www.w3.org/ns/entailment/data/RDFS.html> dc:title "Information Resource RDFS Entailment" ;
   xhv:stylesheet <http://www.w3.org/StyleSheets/TR/base> .
```
... or set up the server...

RewriteEngine On
RewriteBase /ns/entailment/data/

RewriteRule RDFS.ttl
   /2007/08/pyRdfa/extract?format=turtle\&amp;
      uri=http://www.w3.org/ns/entailment/data/RDFS.html [L]
The important point: the *content* is created only once
Enough talk; how does it work?
Unique identifier for **RDFS Entailment**.

“http://www.w3.org/ns/entailment/RDFS” is the URI. The [specification for the RDFS entailment](http://www.w3.org/ns/entailment/RDFS) is part of the [RDF Semantics](http://www.w3.org/TR/rdf-semantics/) W3C Recommendation.

For more information about RDF, please refer to the [RDF Concepts and Abstract Syntax Recommendation](http://www.w3.org/TR/rdf-concepts/).

*Ivan Herman, ivan@w3.org, W3C, Semantic Web Activity Lead, 2009-05-03*
The source and generated RDF…

<p about="http://www.w3.org/ns/entailment/RDFS"
    property="http://purl.org/dc/terms/description">
    Unique identifier for <em>RDFS Entailment</em>.
</p>
The source and generated RDF...

<p about="http://www.w3.org/ns/entailment/RDFS" property="http://purl.org/dc/terms/description">
  Unique identifier for <em>RDFS Entailment</em>.
</p>

<http://www.w3.org/ns/entailment/RDFS>

...
The source and generated RDF...

Unique identifier for <em>RDFS Entailment</em>.

...
The source and generated RDF...

```xml
<p about="http://www.w3.org/ns/entailment/RDFS"
    property="http://purl.org/dc/terms/description">
  Unique identifier for <em>RDFS Entailment</em>.
</p>

<http://www.w3.org/ns/entailment/RDFS>
  <http://purl.org/dc/terms/description>
    "Unique identifier for RDFS Entailment."
  </http://purl.org/dc/terms/description>
</http://www.w3.org/ns/entailment/RDFS>
```
Unique identifier for *RDFS Entailment*.

“http://www.w3.org/ns/entailment/RDFS” is the URI. The [specification for the RDFS entailment](http://www.w3.org/ns/entailment/RDFS) is part of the [RDF Semantics](http://www.w3.org/TR/rdf-sparql-10/) W3C Recommendation.

For more information about RDF, please refer to the [RDF Concepts and Abstract Syntax Recommendation](http://www.w3.org/TR/rdf-concepts/).

*Ivan Herman, ivan@w3.org, W3C, Semantic Web Activity Lead, 2009-05-03*
The source and generated RDF...

<a about="http://www.w3.org/ns/entailment/RDFS"
    rel="http://www.w3.org/2000/01/rdf-schema#seeAlso"
    href="http://www.w3.org/TR/2004/REC-rdf-mt-20040210/">
    RDF Semantics.
</a>
The source and generated RDF...

<a about="http://www.w3.org/ns/entailment/RDFS"
   rel="http://www.w3.org/2000/01/rdf-schema#seeAlso"
   href="http://www.w3.org/TR/2004/REC-rdf-mt-20040210/">
   RDF Semantics.
</a>

<http://www.w3.org/ns/entailment/RDFS>

....
The source and generated RDF…

```
<a about="http://www.w3.org/ns/entailment/RDFS"
rel="http://www.w3.org/2000/01/rdf-schema#seeAlso"
</a>
```

```
<http://www.w3.org/ns/entailment/RDFS>
  <http://www.w3.org/2000/01/rdf-schema#seeAlso>
    ...
  </http://www.w3.org/2000/01/rdf-schema#seeAlso>
```

The source and generated RDF...

```xml
<a about="http://www.w3.org/ns/entailment/RDFS"
    rel="http://www.w3.org/2000/01/rdf-schema#seeAlso"
    href="http://www.w3.org/TR/2004/REC-rdf-mt-20040210/">
    RDF Semantics.
</a>
```

```
http://www.w3.org/ns/entailment/RDFS
    http://www.w3.org/2000/01/rdf-schema#seeAlso
```
Is that it?
The combination of @about with @rel/@property and possibly @href covers most of what we need.

But this is too complex for authors.
Just compare

```
<http://www.w3.org/ns/entailment/RDFS>
  <http://purl.org/dc/terms/description>
    "Unique identifier for RDFS Entailment." .
<http://www.w3.org/ns/entailment/RDFS>
  <http://www.w3.org/2000/01/rdf-schema#seeAlso>
```

▷ with

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc: <http://purl.org/dc/terms/> .

<http://www.w3.org/ns/entailment/RDFS>
  rdfs:seeAlso
    <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> ;
  dc:description "Unique identifier for RDFS Entailment." .
```
The “Turtle” aspects of RDFa

- Use compact URI-s when possible
- Make use of XML structure for
  - shared subjects
  - shared predicates
  - create blank nodes
  - ...
Compact URIs ("CURIE"s)

- Just like in Turtle:
  - define a prefix via @prefix
  - use prefix:reference to abbreviate a URI
CURIE definition and usage

```html
<html>
  ...
  <p about="http://www.w3.org/ns/entailment/RDFS"
      property="http://purl.org/dc/terms/description">
    Unique identifier for <em>RDFS Entailment</em>.
  </p>
  ...
</html>

- can be replaced by:

```html
<html prefix="dc: http://purl.org/dc/terms/">
  ...
  <p about="http://www.w3.org/ns/entailment/RDFS"
      property="dc:description">
    Unique identifier for <em>RDFS Entailment</em>.
  </p>
  ...
</html>
```
Some details on @prefix

- Can be anywhere in the XML tree and is valid for the whole sub-tree
  - i.e., the html element is not the only place to have it
- The same @prefix attribute can hold several definitions:
  - prefix="dc: http://purl.org… rdfs: http://…"
Some details on @prefix

- An alternative (deprecated) syntax is
  - xmlns:dc="http://purl.org/dc/terms/"

- CURIEs and “real” URIs can be mixed
  - if an attribute value can be interpreted as a CURIE, fine
  - alternatively, it is considered as a URI

- CURIEs can be used on RDFa attributes only!
  - e.g., not for @href
RDFa 1.0 Warnings on CURIEs

In RDFa 1.0

- only the xslt:XXX syntax is usable
- CURIEs on @about can only be used with the syntax: about="[pref:ref]"
- Only CURIEs can be used on, e.g., @property or @rel (no fallback on URIs)
Sharing subjects

- The basic principle: @about is inherited by children nodes
  - i.e., no reason to repeat it
Shared subject example

```html
<html prefix="dc: http://purl.org/dc/terms/
    rdfs: http://www.w3.org/2000/01/rdf-schema#">
    ...
    <body about="http://www.w3.org/ns/entailment/RDFS">
        ...
        <p property="dc:description">
            Unique identifier for <em>RDFS Entailment</em>.</p>
        <p>...<a rel="rdfs:seeAlso"
            href="http://www.w3.org/TR/2004/REC-rdf-mt-20040210">
            RDFS Semantics</a>...</p>
```
... yielding

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc: <http://purl.org/dc/terms/> .

<http://www.w3.org/ns/entailment/RDFS>
   rdfs:seeAlso
       <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> ;
   dc:description "Unique identifier for RDFS Entailment.".
Subjects, and objects, and subjects again...
The rules until now

- What we said is:
  - @about sets the subject
  - @href sets the object

- But that is not always good enough
  - we do not always want active links (i.e., the “a” element)
We may not always want links...

- The RDFa @resource attribute is equivalent to @href
  - it sets the object, just like @href
  - but it is ignored by a browser, i.e., not a link!
  - e.g.,:

```
<span about="http://www.ivan-herman.net/foaf#me">
  <span rel="rdfs:seeAlso"
    resource="http://www.ivan-herman.net/foaf">
    Activity Lead</span>
</span>
```
“Chaining”

- Here is what we would like to have in RDFa

```xml
<http://www.w3.org/ns/entailment/RDFS>
  dc:creator <http://www.ivan-herman.net/foaf#me> .
</http://www.w3.org/ns/entailment/RDFS>

<http://www.ivan-herman.net/foaf#me>
  foaf:mailbox <mailto:ivan@w3.org> ;
  foaf:workplaceHomepage <http://www.w3.org> .
```
“Chaining”

- A straightforward way:

```html
<body about="http://www.w3.org/ns/entailment/RDFS">
 ...
 <address>
  <span rel="dc:creator"
      resource="http://www.ivan-herman.net/foaf#me"/>
  <span about="http://www.ivan-herman.net/foaf#me">
    <a rel="foaf:mailbox"
       href="mailto:ivan@w3.org">ivan@w3.org</a>,
    <a rel="foaf:workplaceHomepage"
       href="http://www.w3.org">W3C</a>
  </span>
 </address>
```
“Chaining”

- A straightforward way:

```html
<body about="http://www.w3.org/ns/entailment/RDFS">
  ...
  <address>
    <span rel="dc:creator" resource="http://www.ivan-herman.net/foaf#me"/>
    <span about="http://www.ivan-herman.net/foaf#me">
      <a rel="foaf:mailbox" href="mailto:ivan@w3.org">ivan@w3.org</a>,
      <a rel="foaf:workplaceHomepage" href="http://www.w3.org">W3C</a>
    </span>
  </address>
</body>
```
“Chaining”: objects become subjects...

- An alternative:

```html
<body about="http://www.w3.org/ns/entailment/RDFS">
  ...
  <address>
    <span rel="dc:creator" resource="http://www.ivan-herman.net/foaf#me">
      <a rel="foaf:mailbox" href="mailto:ivan@w3.org">ivan@w3.org</a>,
      <a rel="foaf:workplaceHomepage" href="http://www.w3.org">W3C</a>
    </span>
  </address>
</body>
```
Chaining means

- @resource (or @href) becomes a subject for the subtree
- This feature is a bit like in RDF/XML
Some extra features
Some extra features we do not have time for…

- Blank nodes can be created using “_:XX”
- Datatypes for literals can be set
- Shorthand for RDF types
- An API is being defined for Web Applications
Some extra features we do not have time for...

- Prefix declarations can be collected in a separate file and referred to via a @profile attribute
  - the “profile file”
  - RDFa1.0 warning: this is an RDFa1.1 feature!
“Term” declarations

- A profile file can also define a *term*:
  - assign a URI to a simple string
- The term can be used directly by authors, without prefixes
Term Example

- Say, file “http://ex.org/prof” defines a mapping:
  - "desc" → "http://purl.org/dc/terms/description"
Term Example

```html
<html prefix="dc: http://purl.org/dc/terms/"
      rdfs: http://www.w3.org/2000/01/rdf-schema#">
 ...
<body profile="http://ex.org/prof.html" about="...">
 ...
<p property="desc">
  Unique identifier for <em>RDFS Entailment</em>.</p>
```
Terms are important…

- Usage of CURIEs and URIs is intuitive for RDF people…
- It is *not* for average HTML authors!
- Profile files can be published by major publishers:
  - Dublin Core, FOAF, …
  - FaceBook, Google, …
- … and users can simply refer to the profiles
Publishing RDFa

- RDFa gives an easy way of publishing RDF data on the Web
- Often, the same RDF data is available in different formats, including RDFa
  - it is up to the client to choose which one to use
  - Web Applications would rely on RDFa, though…
LOC example

Semantic Web

From Library of Congress Subject Headings

<table>
<thead>
<tr>
<th>Details</th>
<th>Visualization</th>
<th>Suggest Terminology</th>
</tr>
</thead>
</table>

**Semantic Web**

**URI**
<http://id.loc.gov/authorities/sh2002000569#concept>

**Type**
Topical Term

**Broader Terms**
- Semantic Integration (Computer systems)
- Semantic networks (Information theory)
- World Wide Web

**Related Terms**
- Microformats

**Sources**
- Engr. index online, May 6, 2002 (identifier: Semantic Web)
- ASTI on FirstSearch, May 6, 2002: in titles (semantic Web)

**LC Classification**
TK5105.88815
LOC example

Term</p>
   </li>

<li>
<h3>Broader Terms</h3>
<ul class="std">
   <li>
      <a href="http://id.loc.gov/authorities/sh2004000479#concept" rel="skos:broader">
         <span property="skos:prefLabel" xml:lang="en">Semantic Integration (Computer Theory)</span>
      </a>
   </li>
   <li>
      <a href="http://id.loc.gov/authorities/sh920004914#concept" rel="skos:broader">
         <span property="skos:prefLabel" xml:lang="en">Semantic Networks (Information Systems)</span>
      </a>
   </li>
   <li>
      <a href="http://id.loc.gov/authorities/sh95000541#concept" rel="skos:broader">
         <span property="skos:prefLabel" xml:lang="en">World Wide Web</span>
      </a>
   </li>
</ul>
</li>
Consuming RDFa

- Various search engines begin to consume RDFa
  - Google, Yahoo, ...
    - they may specify which vocabularies they “understand”
    - this is still an evolving area
- Facebook’s “social graph” is based on RDFa
Google’s rich snippet

- Embedded metadata (in microformat or RDFa) is used to improve search result page
  - at the moment only a few vocabularies are recognized, but that will evolve over the years
Effects of, e.g., Google

- A number of popular sites publish RDFa as part of their normal pages:
  - Tesco, BestBuy, Slideshare, The London Gazette, Newsweek, O’Reilly Catalog, the White House…
  - Creative Commons snippets are in RDFa
BestBuy Example for RDFa Usage
Best Buy Example for RDFa Usage
Effects on BestBuy

- Reported in a BestBuy blog:
  - GoodRelations+RDFa improved Google rank tremendously
  - 30% increase in traffic on BestBuy store pages
  - Yahoo observes a 15% increase in click-through rate
- Not bad…😊
Bell and Howell DV550UW 12MP Digital Video Camera with Underwater Housing

Rating 3.8 ★★★★☆ 10 reviews
Read Reviews | Write a review

Today: $68.99

Get 5% back with Club O
View Details

Brief Description
Item#: 12403958
- Take great videos with the Bell + Howell digital video camera
- Digital video camera has ultra compact body and features a flip-up USB
- Camcorder includes waterproof underwater housing to take video underwater (up to 33 ft)

Sold out!
This product is not in stock.
We will notify you when this product is in stock. Please enter your email address below, then...
Overstock.com example

Free Shipping on yo

with-Underwater-Housing/4450313/product.html#myoffer}.


<http://www.overstock.com/Electronics/Bell-and-Howell-DV550UW-12MP-Digital-Video-Camera-with-Underwater-Housing/4450313/product.html#TypeAndQuantityNode> a gr:TypeAndQuantityNode;

g:hasUnitOfMeasurement "C62"^^xsd:string;


g:hasBusinessFunction gr:Sell;

g:hasPriceSpecification  
{ gr:hasCurrency "USD"^^xsd:string;
  gr:hasCurrencyValue "68.99"^^xsd:float
R2RML
Bridge to relational databases: R2RML

- Data on the Web are mostly stored in databases
- “Bridges” are being defined:
  - a layer between RDF and the relational data
    - RDB tables are “mapped” to RDF graphs, possibly on the fly
    - different mapping approaches are being used
  - a number RDB systems offer this facility already (eg, Oracle, OpenLink, ...)
- R2RML is W3C’s evolving standard in this area
What R2RML does

- It defines how a table is mapped on RDF
  - is defined in term of an RDB schema
    - each row is mapped on a common subject
    - column headers yield predicates
    - each cell is mapped on an object
  - different tables within the same database are also linked in the graph
An R2RML processor has access to:

- an RDB schema
- an R2RML instance
- a database governed by the schema

... and produces an RDF graph
A bookshop example

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>
A bookshop example

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

The Glass Palace

2000

http://isbn/000651409X

Ghosh, Amitav

http://www.amitavghosh.com
A bookshop example

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

The Glass Palace

2000

http://...isbn/000651409X

Ghosh, Amitav

http://www.amitavghosh.com
A bookshop example
Step 1: transform “Person Table”

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

ID | Name             | Homepage                   |
---|------------------|-----------------------------|
id_xyz | Ghosh, Amitav | http://www.amitavghosh.com |

:P_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [ rr:termtype "BlankNode" ;
  rr:column "ID" ;
  ];
  rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:name ];
  rr:objectMap [ rr:column "Name" ]
  ];
  rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:homepage ];
  rr:objectMap [ rr:column "Homepage" ;
  rr:termtype "IRI" ]
  ];
Step 2: transform “Book Table”

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

:B_Table rdf:type rr:TriplesMap;
   rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:title ];
     rr:objectMap [ rr:column "Title" ] ];
   rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:year ];
     rr:objectMap [ rr:column "Year" ] ];
   rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:author ];
     rr:objectMap [ rr:column "Name" ];
   ];
   rr:predicateObjectMap [ rr:predicateMap [ rr:predicate a:homepage ];
     rr:objectMap [ rr:column "Homepage" ] ];
Step 3: “bind” the two tables

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

ID | Name | Homepage
---|------|----------------|
id_xyz | Ghosh, Amitav | http://www.amitavghosh.com

:B_Table a rr:TriplesMap ;
  ... rr:refPredicateObjectMap [ 
    rr:refPredicateMap [ 
      rr:predicates a:author 
    ]; 
    rr:refObjectMap [ 
      rr:parentTriplesMap :P_Table ; 
      rr:joinCondition
        "{child}.Author = {parent}.ID"
    ] 
  ] .
Further R2RML features

- There are some additional features
  - assign a datatype to a literal object
  - more complicated object assignments (e.g., for a specific column the object is a cell of another column)

```
:B_Table rdf:type rr:TriplesMap ;
  ...  
  rr:predicateObjectMap [  
    ...  
    rr:objectMap [   
      rr:column "Year" ;   
      rr:datatype xsd:year
    ]
  ] ;
```
Further R2RML features: logical table

- Back to our example:

```
:B_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [  # B.Table is a TriplesMap
    rr:value [  # containing a subjectMap
      rr:prefix "http://...isbn/" ;
      rr:column "ISBN" ;  
    ]
  ];
```

<table>
<thead>
<tr>
<th>ISBN</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>...</td>
</tr>
</tbody>
</table>
Further R2RML features: logical table

- An alternative could have been to use SQL
  - generate a “logical table”
  - all other definitions are on that logical table
- Would be an overkill for our example, but can be very powerful for complicated cases!

```
:B_Table rdf:type rr:TriplesMap ;
rr:SQLQuery "Select ("http://...isbn/" || ISBN) AS id, Author, Title, Publisher, Year from b_table " ;

:B_Table rdf:type rr:TriplesMap ;
... rr:subjectMap [ ff:column "id" ];
```

<table>
<thead>
<tr>
<th>ISBN</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>...</td>
</tr>
</tbody>
</table>
R2RML Direct Mapping

- A R2RML processor might be complex
  - includes an SQL engine, for example
  - fine when bound to big RDB systems
- An alternative: generate an crude RDF Graph
  - no transformations, links among tables only, etc.
  - it is the equivalent of a “Null” R2RML, i.e., no need to specify one
- That graph can be processed further with RDFS, rules, etc
- This is also properly defined, referred to as “Direct Mapping”
### Direct mapping of the bookshop tables

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>
Direct mapping of the bookshop tables
Direct mapping of the bookshop tables

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

The Glass Palace

2000

0006511409X

<Book/ISBN=0006511409X#_>

<Book#Title>

<Book#Year>

<Book#Author>

<Author/ID=id_xyz#_>

<Author#Name>

<Author#Homepage>

Ghosh, Amitav

http://www.amitavghosh.com
Property names should be mapped
URI-s should be minted
Literals should be replaced by URI-s

Direct graph must be transformed
Transformation with SPARQL 1.1

CONSTRUCT {
?id a:title ?title ;
a:year ?year ;
a:author _:x .
_:x a:name ?name ;
a:homepage ?hp .
}
WHERE {
?book
<Book#ISBN> ?isbn ;
<Book#Title> ?title ;
<Book#Year> ?year ;
<Book#Author> ?author .
?author
<Author#Name> ?name ;
<Author#Homepage> ?homepage .
BIND (IRI(fn:concat("http://...",?isbn)) AS ?id)
BIND (IRI(?homepage) AS ?hp)
}

Ghosh, Amitav
http://www.amitavghosh.com

The Glass Palace
2000

http://isbn/000651409X

0006511409X

The Glass Palace

http://www.amitavghosh.com

Ghosh, Amitav
http://www.amitavghosh.com
Thank you for your attention

These slides are also available on the Web:

http://www.w3.org/2010/Talks/1124-Amsterdam-IH/