Providing Linked Data

Presented by:
Barry Norton
Maribel Acosta
Motivation: Music!
EUCLID - Querying Linked Data
Linked Data Principles

1. Use URIs as **names** for things.
2. Use HTTP URIs so that users 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 users can **discover** more things.
Linked Data Lifecycle

Source: Sören Auer. “The Semantic Data Web” (slides)

Source: José M. Alvarez. “My Linked Data Lifecycle”

Source: Michael Hausenblas. “Linked Data lifecycle”

Linked Data Lifecycle

Interlinking / Fusion

Classification / Enrichment

Feedback/ Update

Validate

Publish

Consume

Produce

Manual revision / Authoring

Storage / Querying

Quality Analysis

EUCLID - Providing Linked Data
Core Tasks for Providing Linked Data

Based on the proposed LD lifecycles and the LD principles, we can identify 3 main tasks for providing LD:

① Creating: includes data extraction, creation of HTTP URIs, and vocabulary selection. (LD principles 1 & 2)

② Interlinking: involves the creation of (RDF) links to external data sets. (LD principle 4)

③ Publishing: consists of creating the metadata and making the data set accessible. (LD principle 3)
Agenda

1. Creating Linked Data
2. Interlinking Linked Data
3. Publishing Linked Data
4. Linked Data publishing checklist
CREATING LINKED DATA
Extracting the Data

• The data of interest may be stored in a wide range or formats:

  - Spreadsheets or tabular data
  - Databases
  - Text

• Several tools support the process of mining data from different repositories, for example:

  - Refine
  - W3C R2RML
  - Zemanta
  - GATE
  - DBpedia Spotlight

EUCLID - Providing Linked Data
Using the RDF Data Model

- The RDF data model is used to represent the extracted information

- The **nodes** represent the concepts/entities within the data. A node corresponds to a URI, a blank node or a literal (only in predicates)

- The relationships between the concepts/entities are modeled as **arcs**
Naming Things: URIs

• All the *things* or distinct entities within the data must be named

• According to the Linked Data principles, the standard mechanism to name entities is the **URI**

• Designing **Cool URIs**:
  - Leave out information about the data regarding to: author, technologies, status, access mechanisms, ...
  - **Simplicity**: short, mnemonic URIs
  - **Stability**: maintain the URIs as long as possible
  - **Manageability**: issue the URIs in a way that you can manage

Source: http://www.w3.org/TR/cooluris/
Selecting Vocabularies

• Vocabularies model the concepts and the relationship between them in a knowledge domain

• Terms from well-known vocabularies should be reused wherever possible

• New terms should be define only if you can not find required terms in existing vocabularies

• A large number of vocabularies in RDF are openly available, e.g., Linked Open Vocabularies (LOV)
Selecting Vocabularies (2)

Linked Open Vocabularies

322 vocabularies classified by domain

Source: http://lov.okfn.org/dataset/lov/

EUCLID - Providing Linked Data
Selecting Vocabularies (3)

Linked Open Vocabularies: Analyzing MusicOntology

Source: http://lov.okfn.org/dataset/lov/details/vocabulary_mo.html
Other lists of well-known vocabularies are maintained by:

- **W3C SWEO Linking Open Data community project**
  
  [http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies](http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies)

- **Library Linked Data Incubator Group: Vocabularies in the library domain**
  
  [http://www.w3.org/2005/Incubator/Ild/XGR-Ild-vocabdataset-20111025](http://www.w3.org/2005/Incubator/Ild/XGR-Ild-vocabdataset-20111025)
INTERLINKING LINKED DATA
Interlinking Data Sets

• It’s one of the Linked Data principles!

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

• Involves the creation of RDF links between two different RDF data sets:
  – Links at instance level (rdfs:seeAlso, owl:sameAs)
  – Links at schema level (RDFS subclass/subproperty, OWL equivalent class/property, SKOS mapping properties)

• Appropriate links are detected via link discovery
Interlinking Data Sets (2)

Challenges for link discovery

• Linked Data sets are heterogeneous in terms of vocabularies, formats and data representation

• Large range of knowledge domains

• Scalability: LD is composed of a large number of data sets and RDF triples, hence it is not possible to compare every possible entity pair

Source: Robert Isele. “LOD2 Webinar Series: Silk”
Interlinking Data Sets (3)

Challenges for link discovery

• It corresponds to the entity resolution problem: deciding whether two entities correspond to same object in the real world

• Name ambiguities: typos, misspellings, different languages, homonyms

• Structural ambiguities: same concepts/entities with different structures. Requires the application of ontology and schema matching techniques
Interlinking Data Sets (4)

RDF data sets can be interlinked:

**Manually**
- Involves the manual exploration of LD data sets and their RDF resources to identify linking targets
- May not be feasible when the number of entities within the data set is very large

**Automatically**
- Using tools that perform link discovery based on linkage rules, for example: **Silk**, Limes and xCurator
owl:sameAs & rdfs:seeAlso

• owl:sameAs
  • Creates links between individuals
  • States that two URIs refer to the same individuals

• rdfs:seeAlso
  • States that a resource may provide additional information about the subject resource

• Links in MusicBrainz:
  – owl:seeAlso is used for music artists
  – rdfs:seeAlso is used for albums
SKOS

• Simple Knowledge Organization System
  – http://www.w3.org/TR/skos-reference/

• Data model for knowledge organization systems (thesauri, classification scheme, taxonomies)

• SKOS data is expressed as RDF triples

• Allows the creation of RDF links between different data sets with the usage of mapping properties
SKOS: Mapping Properties

These properties are used to link SKOS concepts (particularly instances) in different schemes:

- **skos:closeMatch**: links two concepts that are sufficiently similar (sometimes can be used interchangeably)
- **skos:exactMatch**: indicates that the two concepts can be used interchangeably.
  - Axiom: It is a transitive property
- **skos:relatedMatch**: states an associative mapping link between two concepts
Example of SKOS exact match

@prefix skos: <http://www.w3.org/2004/02/skos/core#>
@prefix mo: <http://purl.org/ontology/mo/>
@prefix dbpedia-ont: <http://dbpedia.org/ontology/>
@prefix schema: <http://schema.org/>

mo:MusicArtist skos:exactMatch dbpedia-ont:MusicalArtist.


Example of SKOS close match

@prefix skos: <http://www.w3.org/2004/02/skos/core#>
@prefix mo: <http://purl.org/ontology/mo/>
@prefix dbpedia-ont: <http://dbpedia.org/ontology/>
@prefix schema: <http://schema.org/>

mo:SignalGroup skos:closeMatch schema:MusicAlbum.

mo:SignalGroup skos:closeMatch dbpedia-ont:Album.
Integrity conditions

• Guarantee consistency and avoid contradictions in the relationships between SKOS concepts

Partial Mapping Relation diagram with integrity conditions

**Symmetric**
- skos:close Match
- skos:related Match

**Symmetric & Transitive**
- skos:exact Match

**Disjoint with**
PUBLISHING LINKED DATA
Publishing Linked Data

Once the RDF data set has been created and interlinked, the publishing process involves the following tasks:

1. **Metadata** creation for describing the data set
2. Making the data set **accessible**
3. Exposing the data set in Linked Data **repositories**
4. **Validating** the data set
Describing RDF Data Sets

• Consists of providing (machine-readable) **metadata** of RDF data sets which can be processed by engines

• This information allows for:
  
  – Efficient and effective search of data sets
  
  – Selection of appropriate data sets (for consumption or interlinking)
  
  – Get general statistics of the data sets
Describing RDF Data Sets (2)

• The common language for describing RDF data sets is VoID (Vocabulary of Interlinked Data sets)

• Defines an RDF data set with the predicate `void:Dataset`

• Covers 4 types of metadata:
  • General metadata
  • Structural metadata
  • Descriptions of linksets
  • Access metadata
VoID: General Metadata

• General metadata is used by users to identify appropriate data sets.

• Specifies information about description of the data set, contact person/organization, the license of the data set, data subject and some technical features.

• VoID (re)uses predicates from the Dublin Core Metadata\(^1\) and FOAF\(^2\) vocabularies.

---


VoID: General Metadata

General Information

Source: [http://www.w3.org/TR/void/#metadata](http://www.w3.org/TR/void/#metadata)

Contains information about the creation of the data set

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcterms:title</td>
<td>Literal</td>
<td>Name of the data set.</td>
</tr>
<tr>
<td>dcterms:description</td>
<td>Literal</td>
<td>Description of the data set.</td>
</tr>
<tr>
<td>dcterms:source</td>
<td>RDF resource</td>
<td>Source from which the data set was derived.</td>
</tr>
<tr>
<td>dcterms:creator</td>
<td>RDF resource</td>
<td>Primarily responsible of creating the data set.</td>
</tr>
<tr>
<td>dcterms:date</td>
<td>xsd:date</td>
<td>Time associated with an event in the life-cycle of the resource.</td>
</tr>
<tr>
<td>dcterms:created</td>
<td>xsd:date</td>
<td>Date of creation of the data set.</td>
</tr>
<tr>
<td>dcterms:issued</td>
<td>xsd:date</td>
<td>Date of publication of the data set.</td>
</tr>
<tr>
<td>dcterms:modified</td>
<td>xsd:date</td>
<td>Date on which the data set was changed.</td>
</tr>
<tr>
<td>foaf:homepage</td>
<td>Literal</td>
<td>Name of the data set.</td>
</tr>
<tr>
<td>dcterms:publisher</td>
<td>RDF resource</td>
<td>Entity responsible for making the data set available.</td>
</tr>
<tr>
<td>dcterms:contributor</td>
<td>RDF resource</td>
<td>Entity responsible for making contributions to the data set.</td>
</tr>
</tbody>
</table>
VoID: General Metadata (3)

Other Information

- **License of the data set:** specifies the usage conditions of the data. The license can be pointed with the property `dcterms:license`

- **Category of the data set:** to specify the topics or domains covered by the data set, the property `dcterms:subject` can be used

- **Technical features:** the property `void:feature` can be used to express technical properties of the data (e.g. RDF serialization formats)
VoID: Structural Metadata

• Provides high-level information about the internal structure of the data set

• This metadata is useful when exploring or querying the data set

• Includes information about resources, vocabularies used in the data set, statistics and examples of resources in the data set
Information about resources

- **Example resources**: allow users to get an impression of the kind of resources included in the data set. Examples can be shown with the property `void:exampleResource`

  ```PREFIX void: http://void Spec/2010/
  :MusicBrainz a void:Dataset; 
  void:exampleResource <http://musicbrainz.org/artist/b10b3bfc-cf9e-42e0-be17-e2c3e1d260d> .
  ```

- **Pattern for resource URIs**: the `void:uriSpace` property can be used to state that all the entity URIs in a data set start with a given string

  ```PREFIX void: http://void Spec/2010/
  :MusicBrainz a void:Dataset; 
  ```
Vocabularies used in the data set

- The `void:vocabulary` property identifies the vocabulary or ontology that is used in a data set

- Typically, only the most relevant vocabularies are listed

```
:MusicBrainz a void:Dataset;
```

- This property can only be used for entire vocabularies. It **cannot** be used to express that a subset of the vocabulary occurs in the data set.
# VoID: Structural Metadata

## Statistics about a data set

Express numeric statistics about a data set:

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void:triples</td>
<td>Number</td>
<td>Total number of triples contained in the data set.</td>
</tr>
<tr>
<td>void:entities</td>
<td>Number</td>
<td>Total number of entities that are described in the data set. An entity must have a URI, and match the void:uriRegexPattern</td>
</tr>
<tr>
<td>void:classes</td>
<td>Number</td>
<td>Total number of distinct classes in the data set.</td>
</tr>
<tr>
<td>void:properties</td>
<td>Number</td>
<td>Total number of distinct properties in the data set.</td>
</tr>
<tr>
<td>void:distinctSubjects</td>
<td>Number</td>
<td>Total number of distinct subjects in the data set.</td>
</tr>
<tr>
<td>void:distinctObjects</td>
<td>Number</td>
<td>Total number of distinct objects in the data set.</td>
</tr>
<tr>
<td>void:documents</td>
<td>Number</td>
<td>Total number of documents, in case that the data set is published as a set of individual documents.</td>
</tr>
</tbody>
</table>

Source: http://www.w3.org/TR/void/#metadata
Partitioned data sets

- The `void:subset` property provides description of *parts* of a data set

  ```
  :MusicBrainz a void:Dataset;
  void:subset :MusicBrainzArtists .
  ```

- Data sets can be partitioned based on **classes** or **properties**:
  - `void:classPartition` contains only instances of a particular class
  - `void:propertyPartition` contains only triples with a particular predicate

  ```
  :MusicBrainz a void:Dataset;
  void:classPartition [ void:class mo:Release . ] ;
  ```
VoID: Describing Linksets

- **Linkset**: collection of RDF links between two RDF data sets

@PREFIX void:<http://rdfs.org/ns/void#>
@PREFIX owl:<http://www.w3.org/2002/07/owl#>

:DS1 a void:Dataset .
:DS2 a void:Dataset .
:LS1 a void:Linkset;
  void:linkPredicate owl:sameAs;
  void:target :DS1, :DS2 .
Example

@PREFIX void:<http://rdfs.org/ns/void#>  
@PREFIX skos:<http://www.w3.org/2002/07/owl#>

:MusicBrainz a void:Dataset .  
:DBpedia a void:Dataset .

:MusicBrainz void:classPartition :MBArtists .  
:MBArtists void:class mo:MusicArtist .

:MBArtists a void:Linkset;  
    void:linkPredicate 
        skos:exactMatch;  
    void:target :MusicBrainz, :DBpedia .
VoID: Access Metadata

The access metadata describes the methods of accessing the actual RDF data set

<table>
<thead>
<tr>
<th>Method</th>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI look up endpoint</td>
<td>void:uriLookupEndpoint</td>
<td>Specifies the URI of a service for accessing the data set (different from the SPARQL protocol)</td>
</tr>
<tr>
<td>Root resource</td>
<td>void:rootResource</td>
<td>URI of the top concepts (only for data sets structured as trees)</td>
</tr>
<tr>
<td>SPARQL endpoint</td>
<td>void:sparqlEndpoint</td>
<td>Provides access to the data set via the SPARQL protocol.*</td>
</tr>
<tr>
<td>RDF data dumps</td>
<td>void:dataDump</td>
<td>Specifies the location of the dump file. If the data set is split into multiple files, then several values of this property are provided.</td>
</tr>
</tbody>
</table>

* This assumes that the default graph of the SPARQL endpoint contains the data set. VoID cannot express that a data set is contained a specific named graph. This can be specified with SPARQL 1.1. Service Description
Providing Access to the Data Set

The data set can be accessed via different mechanisms:

- Dereferencing HTTP URIs
- RDFa
- SPARQL endpoint
- RDF dump
Dereferencing HTTP URIs

- Allows for easily **exploring** certain resources contained in the data set

- **What to return for a URI?**
  - **Immediate description:** triples where the URI is the subject.
  - **Backlinks:** triples where the URI is the object.
  - **Related descriptions:** information of interest in typical usage scenarios.
  - **Metadata:** information as author and licensing information.
  - **Syntax:** RDF descriptions as RDF/XML and human-readable formats.

Source: *How to Publish Linked Data on The Web* - Chris Bizer, Richard Cyganiak, Tom Heath.

- Applications (e.g. LD browsers) render the retrieved information so it can be perceived by a user.
Dereferencing HTTP URIs (2)

Example: Dereferencing

dbpedia.org/page/The_Beatles

About: The Beatles
An Entity of Type: organisation, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

The Beatles were an English rock band formed in Liverpool in 1960 and one of the most commercially successful and critically acclaimed acts in the history of popular music. The group’s best-known lineup consisted of John Lennon (rhythm guitar, vocals), Paul McCartney (bass guitar, vocals), George Harrison (lead guitar, vocals) and Ringo Starr (drums, vocals).

- The Beatles formed a group of música procedente de Liverpool, Inglaterra, format per John Lennon, Paul McCartney, George Harrison i Ringo Starr. Són un dels èxits i vencedors de discos de la història de la música popular, i, a més, han aconseguit el reconeiximent dels crítics. Les seves aportacions innovadores a la música cultural van tenir una gran transcendència en els anys setanta, que es prolongà d’una manera o altra en els anys posteriors. The Beatles són un dels dels millors que existien avui en dia i acollides per la crítica de la història de la música popular. The Beatles formen el grup amb més vencedors del segle XX. Al Reg singles i álbums que van arribar al número u. El seu èxit es va repetir també en molts altres països: EMI assegura que el 1965 el grup havia venut més milions de singles a principis dels anys 60, van ser els iniciadors del fenomen beat. Partint de la base rítmica del rock, les seves composicions formen una veritable obra amb influències del rhythm and blues, i del blues progressiu, el seu estil va seguir una evolució constant. Des de mitjans dels anys 60, amb l’ambició de l’èxit decantant cap a un cicleclopàtic que inclouia una forta influència oriental i que volia donar com a resultat aquestes de les obres més significatives de la palmedora progressiu. Les seves cançons van reflectir els problemes d’un cert sector de la joventut de l’època, que pretencia restar al marg de l’amonissat societat dels aspectes més reformatgers de la societat occidental, ahora que es demanava nous valors tant estètics i artístics com espirituals i socials relacionals, per oferir l’antimilitarisme. La seva dissolució oficial va tenir lloc l’any 1970. Els seus integrants van seguir les respectives carreeres musicals en solitari, tot i que van siells. L’any 1985, amb motiu de l’edició del primer volum de la trilogia Beatles Anthology, va tenir lloc una reunió virtual del grup, amb imatges d’unes cintes que erenregistrades i que van ser la base de dues noves cançons: Free as a Bird i Real Love.


- The Beatles were a British rock band in the 1960s. With more than 600 million albums sold alone in the United States, they were a major influence on the music of the Beatles, who formed the backbone of the Beatles, and later were the group's namesake. In the mid-1960s and 1968, they became one of the most successful bands in the history of popular music. The group's best-known lineup consisted of John Lennon (rhythm guitar, vocals), Paul McCartney (bass guitar, vocals), George Harrison (lead guitar, vocals) and Ringo Starr (drums, vocals). The Beatles formed a group of música procedente de Liverpool, Inglaterra, format per John Lennon, Paul McCartney, George Harrison i Ringo Starr. Són un dels èxits i vencedors de discos de la història de la música popular, i, a més, han aconseguit el reconeiximent dels crítics. Les seves aportacions innovadores a la música cultural van tenir una gran transcendència en els anys setanta, que es prolongà d’una manera o altra en els anys posteriors. The Beatles són un dels dels millors que existien avui en dia i acollides per la crítica de la història de la música popular. The Beatles formen el grup amb més vencedors del segle XX. Al Reg singles i álbums que van arribar al número u. El seu èxit es va repetir també en molts altres països: EMI assegura que el 1965 el grup havia venut més milions de singles a principis dels anys 60, van ser els iniciadors del fenomen beat. Partint de la base rítmica del rock, les seves composicions formen una veritable obra amb influències del rhythm and blues, i del blues progressiu, el seu estil va seguir una evolució constant. Des de mitjans dels anys 60, amb l’ambició de l’èxit decantant cap a un cicleclopàtic que inclouia una forta influència oriental i que volia donar com a resultat aquestes de les obres més significatives de la palmedora progressiu. Les seves cançons van reflectir els problemes d’un cert sector de la joventut de l’època, que pretencia restar al marg de l’amonissat societat dels aspectes més reformatgers de la societat occidental, ahora que es demanava nous valors tant estètics i artístics com espirituals i socials relacionals, per oferir l’antimilitarisme. La seva dissolució oficial va tenir lloc l’any 1970. Els seus integrants van seguir les respectives carreeres musicals en solitari, tot i que van siells. L’any 1985, amb motiu de l’edició del primer volum de la trilogia Beatles Anthology, va tenir lloc una reunió virtual del grup, amb imatges d’unes cintes que erenregistrades i que van ser la base de dues noves cançons: Free as a Bird i Real Love.


- The Beatles were a British rock band in the 1960s. With more than 600 million albums sold alone in the United States, they were a major influence on the music of the Beatles, who formed the backbone of the Beatles, and later were the group's namesake. In the mid-1960s and 1968, they became one of the most successful bands in the history of popular music. The group's best-known lineup consisted of John Lennon (rhythm guitar, vocals), Paul McCartney (bass guitar, vocals), George Harrison (lead guitar, vocals) and Ringo Starr (drums, vocals).
RDFa

• RDFa = “RDF in attributes”

• Extension to HTML5 for embedding RDF within HTML pages:
  – The HTML is processed by the browser, the (human) consumer don’t see the RDF data
  – The RDF triples within the page are consumed by APIs to extract the (semi-)structured data

• It is considered as the bridge between the Web of Data and the Web of Documents

• It is a complete serialization of RDF
# RDFa: Attributes

<table>
<thead>
<tr>
<th>Attribute role</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>prefix</td>
<td>List of prefix-name IRIs pairs</td>
</tr>
<tr>
<td></td>
<td>vocab</td>
<td>IRI that specifies the vocabulary where the concept is defined</td>
</tr>
<tr>
<td>Subject</td>
<td>about</td>
<td>Specifies the subject of the relationship</td>
</tr>
<tr>
<td>Predicate</td>
<td>property</td>
<td>Express the relationship between the subject and the value</td>
</tr>
<tr>
<td></td>
<td>rel</td>
<td>Defines a relation between the subject and a URL</td>
</tr>
<tr>
<td></td>
<td>rev</td>
<td>Express reverse relationships between two resources</td>
</tr>
<tr>
<td>Resource</td>
<td>href</td>
<td>Specifies an object URI for the rel and rev attributes</td>
</tr>
<tr>
<td></td>
<td>resource</td>
<td>Same as href (used when href is not present)</td>
</tr>
<tr>
<td></td>
<td>src</td>
<td>Specifies the subject of a relationship</td>
</tr>
<tr>
<td>Literal</td>
<td>datatype</td>
<td>Express the datatype of the object of the property attribute</td>
</tr>
<tr>
<td></td>
<td>content</td>
<td>Supply machine-readable content for a literal</td>
</tr>
<tr>
<td></td>
<td>xml:lang, lang</td>
<td>Specifies the language of the literal</td>
</tr>
<tr>
<td>Macro</td>
<td>typeof</td>
<td>Indicate the RDF type(s) to associate with a subject</td>
</tr>
<tr>
<td></td>
<td>inlist</td>
<td>An object is added to the list of a predicate.</td>
</tr>
</tbody>
</table>

EUCLID - Providing Linked Data

46
RDFa: Example

Extracting RDF from HTML

HTML (+RDFa):

```html
<div class="artistheader"
    about="http://musicbrainz.org/artist/b10bbbf-cf9e-42e0-be17-e2c3e1d2600d# _">
    ...
</div>
```

RDF:

```xml
<http://musicbrainz.org/artist/b10bbbf-cf9e-42e0-be17-e2c3e1d2600d# _>
```
RDFa: Example

Extracting RDF from HTML

HTML (+RDFa):

```html
<div class="artistheader" about="http://musicbrainz.org/artist/b10b9bfc-cf9e-42e0-be17-e2c3e1d2600d#_" typeof="http://purl.org/ontology/mo/MusicGroup">
  ...
</div>
```

RDF:

```xml
<http://musicbrainz.org/artist/b10b9bfc-cf9e-42e0-be17-e2c3e1d2600d#_>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
```
RDFa: Example

Extracting RDF from HTML

HTML (+RDFa):

```html
<div class="artistheader"
    about="http://musicbrainz.org/artist/b10bbfc-cf9e-42e0-be17-e2c3e1d2600d#_
    typeof="http://purl.org/ontology/mo/MusicGroup">
    ...
</div>
```

RDF:

```xml
<http://musicbrainz.org/artist/b10bbfc-cf9e-42e0-be17-e2c3e1d2600d#_>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
```
Extracting RDF from MusicBrainz.org

http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d

The Beatles

Overview | Releases | Recordings | Works | Relationships | Aliases | Tags | Details | Edit

Annotation

The official name of “The White Album” is “The Beatles”; please do not add incorrectly titled releases to the database!

Please do not re-add individual mono remaster releases to their individual release groups. The mono remasters were never released separately, and are all already in the database as 13 mediums of the release the name “The Beatles in Mono”.

Annotation last modified on 2012-10-09 07:44 UTC.

Wikipedia

The Beatles were an English rock band formed in Liverpool in 1960. They became the most commercially successful and critically acclaimed act in the rock music era. The group's best-known lineup consisted of John Lennon, Paul McCartney, George Harrison, and Ringo Starr. Rooted in skiffle and 1950s rock and roll, the Beatles later utilized several genres, ranging from pop belters to psychedelic rock, often incorporating classical and other elements in innovative ways. In the early 1960s, their enormous popularity first emerged as "Beatlemania", but as their songwriting grew in sophistication, they came to be perceived by many fans and cultural observers as an embodiment of the ideals shared by the era’s sociocultural revolutions.

Continue reading at Wikipedia...

Discography

<table>
<thead>
<tr>
<th>Year</th>
<th>Title</th>
<th>Artist</th>
<th>Rating</th>
<th>Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>1963</td>
<td>Please Please Me</td>
<td>The Beatles</td>
<td>★★★★★</td>
<td>10</td>
</tr>
<tr>
<td>1963</td>
<td>With The Beatles</td>
<td>The Beatles</td>
<td>★★★★★</td>
<td>10</td>
</tr>
<tr>
<td>1964</td>
<td>Introducing... The Beatles</td>
<td>The Beatles</td>
<td>★★★★★</td>
<td>3</td>
</tr>
<tr>
<td>1964</td>
<td>Meet The Beatles!</td>
<td>The Beatles</td>
<td>★★★★★</td>
<td>1</td>
</tr>
</tbody>
</table>
**RDFa: Example (2)**

**Extracting RDF from MusicBrainz.org**

---

### RDFa Distiller and Parser

This distiller corresponds to the **RDFa 1.0** specification. In 2012, W3C has published an updated version of that specification, called **RDFa Core 1.1**. A new distiller, processing RDFa 1.1 content, has been implemented which supersedes this one. Note that the new distiller can also process RDFa 1.0 content (there are some minor incompatibilities) if the XHTML+RDFa file uses the right (RDFa 1.0) DTD and/or the @version attribute. Users are advised to migrate to RDFa 1.1 in general, including the RDFa 1.1 distiller.

If you intend to use this service regularly on large scale, consider downloading the package and use it locally. Storing a (conceptually) “cached” version of the generated RDF, instead of referring to the live service, might also be an alternative to consider in trying to avoid overloading this server...

---

**Source:** [http://www.w3.org/2007/08/pyRdfa/](http://www.w3.org/2007/08/pyRdfa/)
RDFa: Example (2)

Extracting RDF from MusicBrainz.org

http://www.w3.org/2007/08/pyRdfa/extract?uri=http%3A%2F%2Fmusicbrainz.org%2Fartist%2Fb10bbbfccf9e-42e0-be17-e2c3e1d2600d&format=nt

Watch the EUCLID screencast: http://vimeo.com/euclidproject
RDF Dump

• An RDF dump refers to a file which contains (part of) a data set specified in an RDF format (RDF/XML, N-Triples, N-Quads)

• The data set can be split into several RDF dumps

• A list of available data sets available as RDF dumps can be found at:
  – http://www.w3.org/wiki/DataSetRDFDumps
The SPARQL endpoint refers to the URI of the listener of the SPARQL protocol service, which handles requests for SPARQL protocol operations.

The user submits SPARQL queries to the SPARQL endpoint in order to retrieve only a desired subset of the RDF data set.

List of available SPARQL endpoints:
- [http://www.w3.org/wiki/SparqlEndpoints](http://www.w3.org/wiki/SparqlEndpoints)
Using Linked Data Catalogs

• Data catalogs, markets or repositories are platforms dedicated to provide access to a wide range of data sets from different domains

• Allow data consumers to easily find and use the data

• Usually the catalogs offer relevant metadata about the creation of the data set
How to publish an RDF data set into a catalog?

Create your own data catalog

- Recommended for big organizations/institutions aiming at providing a large number of data sets
- Use a data management system, for example: ckan

Upload your data set into an existing catalog

- Allows data consumers to easily find new data sets
- Common LD catalogs are:
  - the Data Hub
  - The Linking Open Data Cloud
Validating Data Sets

There are different ways to validate the published RDF data set:

**Accessibility**
- **Vapour** - Performs two types of tests: without content negotiation and requesting RDF/XML content
  [http://validator.linkeddata.org/vapour](http://validator.linkeddata.org/vapour)
- **URI Debugger** - Retrieves the HTTP responses of accessing a URI
  [http://linkeddata.informatik.hu-berlin.de/uridbg/](http://linkeddata.informatik.hu-berlin.de/uridbg/)
- **RDF Triple-Checker** – Dereferences namespaces associated with the resources used in the document
  [http://graphite.ecs.soton.ac.uk/checker/](http://graphite.ecs.soton.ac.uk/checker/)

**Parsing & Syntax**
- **W3C RDF/XML Validation Service** – Evaluates the syntax of RDF/XML documents and displays the RDF triples in it
  [http://validator.linkeddata.org/vapour](http://validator.linkeddata.org/vapour)
- **W3C Markup Validation Service** – Checks syntactic correctness for web documents with RDFa markup
  [http://validator.w3.org/](http://validator.w3.org/)

**General validators**
- **RDF:ALERTS** – Validates syntax, undefined resources, datatype and other types of errors
  [http://swse.deri.org/RDFAlerts/](http://swse.deri.org/RDFAlerts/)
Validating Data Sets (2)

Example: Validating URIs with Vapour

Source: http://idi.fundacionctic.org/vapour
Validating Data Sets

Example: Validating URIs with Vapour

Vapour Report

All tests passed!

Summary:

<table>
<thead>
<tr>
<th>Test requirement</th>
<th>Passed tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dereferencing resource URI (requesting RDF/XML)</td>
<td>3/3</td>
</tr>
<tr>
<td>Dereferencing resource URI (without content negotiation)</td>
<td>2/2</td>
</tr>
</tbody>
</table>

Source: http://idi.fundacionctic.org/vapour
Validating Data Sets (4)

Example: Validating URIs with Vapour

Dereferencing resource URI (without content negotiation)

GET /resource/The_Beatles
User-Agent: vapour.sourceforge.net

303 Location:
http://dbpedia.org/page/The_Beatles

GET /page/The_Beatles
User-Agent: vapour.sourceforge.net

200 Content-type: text/html;
charset=UTF-8
Vary: Accept-Encoding

Dereferencing resource URI (requesting RDF/XML)

http://dbpedia.org/data/The_Beatles.xml

Source: http://idi.fundacionctic.org/vapour
PROVIDING LINKED DATA: CHECKLIST
Providing Linked Data: Checklist (1)

Creating Linked Data

- All the relevant entities/concepts were effectively extracted from the raw data?
- Are all the created URIs dereferenceable?
- Are you reusing terms from widely accepted vocabularies?
Providing Linked Data: Checklist (2)

Interlinking Linked Data

- Is the data set linked to other RDF data sets?
- Are the created vocabulary terms linked to other vocabularies?
Providing Linked Data: Checklist (3)

Publishing Linked Data

○ Do you provide data set metadata?
○ Do you provide information about licensing?
○ Do you provide additional access methods?
○ Is the data set available in LD catalogs?
○ Did the data set pass the validation tests?
Summary

In this chapter we studied:

• The Linked Data **lifecycle:**
  • 3 core tasks: creating, interlinking and publishing

• **Creation of Linked Data:**
  • Extracting relevant data, using URIs to name entities and selecting vocabularies and expressing the data using the RDF data model

• **Interlinking Linked Data:**
  • Challenges of link discovery, using Silk to create links between two data sets and using SKOS links

• **Publishing Linked Data:**
  • Creation of data set metadata; publishing the data set via RDF dumps, SPARQL endpoints or RDFa; using RDFa and schema.org to enrich search results, and uploading the data set to a LD catalog
The Web & **Linked Data**

- Linked Data catalogs
- Applications
CKAN

• CKAN is an open source platform for developing data set catalogs

• Implement useful tools for data publishers to support:
  • Data harvesting
  • Creation of metadata
  • Access mechanisms to the data set
  • Updating the data set
  • Monitoring the access to the data set
CKAN (2)

**Source:** http://ckan.org
Metadata
A CKAN portal provides a rich set of metadata for each dataset.

Title – allows intuitive labelling of the dataset for search, sharing and linking.

Unique identifier – dataset has a unique URL which is customizable by the publisher.

Description – additional information describing or analysing the data. This can either be static or an editable wiki which anyone can contribute to instantly or via admin moderation.

Revision history – CKAN allows you to display a revision history for datasets which are freely editable by users (as is thedatahub.org)

Data preview – preview .csv data quickly and easily in browser to see if this is the dataset you want.

Extra fields – these hold any additional information, such as location data (see geospatial feature) or types relevant to the publisher or dataset. How and where extra fields display is customizable.

Source: http://ckan.org
The Data Hub

• The Data Hub is a community-run data catalog which contains more than 5,000 data sets\(^1\)

• “(...) is an openly editable open data catalogue, in the style of Wikipedia”.\(^2\)

• It is implemented on top of the CKAN platform

• Allows the creation of **groups:**
  – The Linking Open Data Cloud group exclusively contains Linked Data sets

---

\(^1\) According to the information presented in the portal on March 2013

\(^2\) Source: http://datahub.io/about
The Data Hub

Welcome to the Data Hub!

Find data

Find datasets

the Data Hub contains 5158 datasets that you can browse, learn about and download.

Share data

Add your own datasets to share them with others and to find other people interested in your data.

Sign up »

Collaborate

Find out more about working with open data by exploring these resources:

- GetTheData.org
- DataPatterns.org
- Open Data Handbook

Who else is here?

Canada

Datasets for http://www.datadotgc.ca/
DataDotGC, which launched, in February 2010, is a Canadian, citizen-led effort to promote open data and help share data that has already been...
Canada has 521 datasets.

Economics Datasets

Group for Economics data especially that which is open data. This can be any kind of data related to economics from development to finance, and micro to macro. We run an open group policy...

Source: http://datahub.io/
The Data Hub (3)

Source: http://datahub.io/
The Linking Open Data Cloud

September 2011

Source: Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch
How to publish an RDF data set in this cloud?

1. The data set must follow the Linked Data principles
2. The data set must contain at least 1,000 RDF triples
3. The data set must contain at least 50 RDF links to a data set that is already in the diagram
4. Access to the data set must be provided

Once these criteria are met, the data publisher must add the data set to the Data Hub catalog, and contact the administrators of the Linking Open Data Cloud group.

Source: http://lod-cloud.net/
• Search engines collect information about web resources in order to produce **richer search results** by improving the display of the results

• This is only possible if the search engines are able to **understand the content** within the web pages

• The HTML pages must be annotated with machine-readable content to describe their content:

  **Mark up format** +  **Vocabulary**
RDFa for marking up data

• RDFa is used to provide (semi-)structured Linked Data embedded in web content

• Examples:
  – Some search engines use RDFa, e.g., Google, Yahoo! and Bing
  – Facebook’s Open Graph is based on RDFa
Google Rich Snippets

• Embedding semantics via RDFa (or microformats/microdata) enhances search results:
Google Rich Snippets (2)

Extracted structured data

```
rdfa-node

property:

  title: Good Old Fashioned Pancakes
  type: food
  image: http://images.media-allrecipes.com/userphotos/250x250/00/22/85/228589.jpg
  site_name: Allrecipes.com
  app_id: 66102450266
```

Item

type: http://schema.org/recipe

property:

  image: http://images.media-allrecipes.com/userphotos/250x250/00/22/85/228589.jpg
  video: Item 1
  name: Good Old Fashioned Pancakes
  aggregating:
  description: “This is a great recipe that I found in my Grandma’s recipe book. Judging from the weathered look of this recipe card, this was a family favorite.”
  author: dakota kelly
  recipeYield: 8 servings
  ingredients:
    1 1/2 cups all-purpose flour
    3 1/2 teaspoons baking powder
    1 teaspoon salt
```
Schema.org

• Collection of schemas/vocabularies to markup the HTML pages

• It is recognized by Bing, Google, Yahoo! and Yandex

• Covers a wide range of knowledge domains

• It also offers an extension mechanism in case the publisher is interested in adding new concepts to the vocabularies
The vocabularies cover the following topics:

- Creative works: CreativeWork, Book, Movie, MusicRecording, Recipe, TVSeries ...
- Embedded non-text objects: AudioObject, ImageObject, VideoObject
- Event
- Organization
- Person
- Place, LocalBusiness, Restaurant ...
- Product, Offer, AggregateOffer
- Review, AggregateRating

“The world is too rich, complex and interesting for a single schema to describe fully on its own. With schema.org we aim to find a balance, by providing a core schema that covers lots of situations, alongside extension mechanisms for extra detail.” (Dan Brickley, schema.org)

Source: http://schema.org/docs/schemas.html
Integrates (/aligns) existing vocabularies where appropriate, e.g. rNews

Source: http://schema.org/Article
Google Knowledge Graph

- The user is able to find answer to their queries without browsing pages
- Provides detailed information
Google Search results include structured data from Freebase

• Might disambiguate search terms
Freebase

- Knowledge base of structured data
- Data is stored as a graph
- Describes data from different domains

An entity graph of people, places and things, built by a community that loves open data.
Bing Snapshot

• Provides structured data related to the search term

• Includes a significant number of entities from more domains

• Connects data from LinkedIn

• Is is powered by the graph engine Trinity.RDF
Leopard

The leopard, Panthera pardus, is a member of the Felidae family and the smallest of the four "big cats" in the genus Panthera, the other three being the tiger, lion, and jaguar. The leopard was onc... + en.wikipedia.org

Scientific Name: Panthera pardus
Biological Classification: Species
Belongs to: Panthera
Notables: Leopard of Panar · Leopard of Rudraprayag · Larisa · Sipira · Angie

Subspecies

African Leopard · Indian Leopard · Amur Leopard · Arabian leopard

People also search for
Open Graph Protocol

• It was originally created by Facebook

• Allows describing web content as **graph objects**, establishing connections between people and objects

• The descriptions are embedded in the web page as **RDFa** data

• Supports description of several domains: basic metadata, **music**, video, articles, books, websites and user profiles

Source: http://ogp.me/
Who is using Open Graph protocol?

Consumers

Google
Mixi
Facebook

Publishers

IMDb
Posterous
Rotten Tomatoes
TIME
Microsoft
NHL

Source: http://ogp.me/
Open Graph Protocol (3)

- Facebook expands vocabulary of relationships beyond “friendship” and “like” ➡️ more actions!

Source: https://developers.facebook.com/docs/opengraph/
List of domains and actions

General
- Like
- Recommend
- Follow

Music
- Listen
- Create a playlist

Movies & TV
- Watch
- Rate
- Wants to watch

Games
- Achieve
- High score

Fitness
- Bike
- Run
- Walk

Book
- Rate
- Read
- Quote
- Wants to read

How can we exploit these links and relationships?

Source: https://developers.facebook.com/docs/opengraph/
Facebook Graph Search

- **Focuses on people and their interests**, exploiting how everything is related to each other.

- Queries are specified using **natural language**.

- Takes advantage of **context** and suggests possible queries.

- Allows for building more **complex** (expressive) queries that are not possible with normal search:
  - For example, “music liked by me and friends who live in my city”
Facebook Graph Search

Context (information from profile):

Living

Karlsruhe, Germany
Current City

Caracas, Venezuela
Hometown

Graph search suggestions:

- Music liked by me and my friends who live in Caracas, Venezuela
  - City - 407,193 like this

- Music liked by me and my friends who live in Karlsruhe, Germany
  - City

- Music liked by me and my friends who live in San Antonio De Los Altos, Miranda, Venezuela
  - City

- Music liked by me and my friends who live nearby and live in Caracas, Venezuela
  - City

- Music liked by me and my friends who live near Venezuela
  - Country

- Music liked by me and my friends who live nearby and live in Karlsruhe, Germany
  - City
Results

Facebook Graph Search

More Than 100 Pages

Refine this search

Page type: Musician

Liked by: My friends who like SemanticWeb.com

Name: Add...

Extend this search

- Movies they've watched
- Photos from these pages
- Videos from these pages

Discover something new

Give feedback

EUCLID - Providing Linked Data
Observations

- Allows for conjunctive queries (applying filter over intermediate results = “apply operator”)

- **Disjunctive** queries are not supported:
  - For example: “My friends who like SemanticWeb.com OR ReadWrite”

- **Post search** is not supported
  - It is not possible to search in post content submitted to the timeline

- User **privacy settings** affect the results
Tools for providing **Linked Data**

- Extracting data from spreadsheets: OpenRefine
- Extracting data from RDBMS: R2RML
- Extracting data from text: Zemanta, OpenCalais, GATE
- Interlinking data sets: Silk
EXTRACTING DATA FROM SPREADSHEETS WITH OPENREFINE
Integrate Chart Data

Task: Integrate latest chart information into your RDF database.

Data may be available in non-RDF formats:
- Plain text
- CSV, TSV, separator-based files
- HTML tables
- Spreadsheets (OpenDocument, Excel, ...)
- XML
- JSON
- ...
Example Data

CSV

The Beatles, 250 million
Elvis Presley, 203.3 million
Michael Jackson, 157.4 million
Madonna, 160.1 million
Led Zeppelin, 135.5 million
Queen, 90.5 million

JSON

```json
{
  "artist": {
    "class": "artist",
    "name": "The Beatles"
  },
  "rank": 1,
  "value": 250 million
}
...
```

HTML tables

<table>
<thead>
<tr>
<th>Artist</th>
<th>Country of origin</th>
<th>Period active</th>
<th>Release-year of first charted record</th>
<th>Total certified units (from available markets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elvis Presley</td>
<td>United States</td>
<td>1954–1977</td>
<td>1954</td>
<td>Total available certified units: 203.3 million</td>
</tr>
<tr>
<td>Michael Jackson</td>
<td>United States</td>
<td>1964–2009</td>
<td>1971</td>
<td>Total available certified units: 157.4 million</td>
</tr>
<tr>
<td>Madonna</td>
<td>United States</td>
<td>1979–present</td>
<td>1982</td>
<td>Total available certified units: 160.1 million</td>
</tr>
<tr>
<td>Queen</td>
<td>United Kingdom</td>
<td>1971–present</td>
<td>1973</td>
<td>Total available certified units: 90.5 million</td>
</tr>
</tbody>
</table>

Quick Facts

OpenRefine

- transforms and cleans messy input data sets.
- is an open-source successor of Google Refine.
- allows for entity reconciliation against SPARQL endpoints or RDF data.
- is extended with plugins that enhance its functionality, e.g. for RDF support.
Use of OpenRefine

1. Messy input data is imported, transformed into a table representation and cleaned.

2. Entity reconciliation is applied to allow for interlinking with existing data sets.

3. Define the structure of the RDF output.

4. The data is exported into some RDF syntax.
Data Transformation

Typical steps:
- Group and explore data items
- Deleting columns or rows based on filter condition
- Split columns into several columns based on condition
- Modify messy data items with GREL, a powerful expression language
- Replay steps from a previous Refine project
How to Generate RDF?

- Additional problem: data needs to be interlinked with existing MusicBrainz data

- This is the point where plugins come into play:
  - RDF Refine: developed by DERI
  - An extension of OpenRefine to support RDF
Core Capabilities

• Interlinking of data by entity reconciliation
  – Against SPARQL endpoints, RDF dumps
  – Discovery of relevant RDF data sets

• RDF export with the help of RDF skeletons
  – Define the vocabulary and graph structure of the RDF serialization
  – In Turtle, RDF/XML
Entity Reconciliation

Typical steps:
• Define a reconciliation service
• Select specific types to reconcile against
• Start reconciling a column against the service
Define RDF Skeletons

• An RDF skeleton defines the structure of the RDF triples that are exported
RDF Skeletons

This is a sample Turtle representation of (up-to) the first 10 rows:

```turtle
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

```
EXTRACTING DATA FROM RDBMS WITH R2RML
**Task**: Integrate data from relational DBMS with Linked Data

**Approach**: map from relational schema to semantic vocabulary with R2RML

**Publishing**: two alternatives —
- Translate SPARQL into SQL on the fly
- Batch transform data into RDF, index and provide SPARQL access in a triplestore
The W3C made, last year, two recommendations for mapping between relational databases and RDF:

- **Direct mapping** directly exposes data as RDF
  - Not allowance for vocabulary mapping
  - No allowance for interlinking (unless URIs used in relational data)
  - Not appropriate for this topic

- **R2RML**, the RDB to RDF mapping language
  - Allows vocabulary mapping (subject, predicate and object maps with class options)
  - Allows interlinking – URIs can be constructed
  - Means to provide MusicBrainz RDF/SPARQL itself

http://www.w3.org/2001/sw/rdb2rdf/
MusicBrainz Next Gen Schema

• **Artist**
  As pre-NGS, but further attributes

• **Artist Credit**
  Allows joint credit

• **Release Group**
  Cf. ‘album’ versus:

  • **Release** • **Track** • **Work**

  • **Medium** • **Track List** • **Recording**

Source: https://wiki.musicbrainz.org/Next_Generation_Schema
Music Ontology

• OWL ontology with following core concepts (classes) and relationships (properties):

Source: http://musicontology.com
R2RML Class Mapping

- Mapping tables to classes is ‘easy’:

```
lb:Artist a rr:TriplesMap ;
   rr:logicalTable [rr:tableName "artist"] ;
   rr:subjectMap
      [rr:class mo:MusicArtist ;
       rr:template
         "http://musicbrainz.org/artist/{gid}#_" ] ;
   rr:predicateObjectMap
      [rr:predicate mo:musicbrainz_guid ;
       rr:objectMap [rr:column "gid" ;
                     rr:datatype xsd:string]] .
```
Mapping columns to properties can be easy:

```
lb:artist_name a rr:TriplesMap;
rr:logicalTable [rr:sqlQuery
  "'"SELECT artist.gid, artist_name.name
  FROM artist
  INNER JOIN artist_name ON artist.name = artist_name.id"'"];
rr:subjectMap lb:sm_artist;
rr:predicateObjectMap
  [rr:predicate foaf:name;
   rr:objectMap [rr:column "name"]].
```
NGS Advanced Relations

• Major entities (Artist, Release Group, Track, etc.) plus URL are paired (l_artist_artist)

• Each pairing of instances refers to a Link

• Links have types (cf. RDF properties) and attributes

Source: http://wiki.musicbrainz.org/Advanced_Relationship
R2RML Advanced Mapping

- Mapping advanced relationships (SQL joins):

```xml
lb:artist_member a rr:TriplesMap ;
  rr:logicalTable [ rr:sqlQuery
    """SELECT a1.gid, a2.gid AS band
    FROM artist a1
    INNER JOIN l_artist_artist ON a1.id = l_artist_artist.entity0
    INNER JOIN link ON l_artist_artist.link = link.id
    INNER JOIN link_type ON link_type.id = l_artist_artist.link
    INNER JOIN artist a2 on l_artist_artist.entity1 = a2.id
    WHERE link_type.gid='5be4c609-9afa-4ea0-910b-12ffb71e3821'
    AND link.ended=FALSE"""
  ] ;
rr:subjectMap lb:sm_artist ;
rr:predicateObjectMap
  [ rr:predicate mo:member_of ;
    rr:objectMap [ rr:template "http://musicbrainz.org/artist/{band}#_" ;
    rr:termType rr:IRI ]] .
```

EUCLID - Providing Linked Data
EXTRACTING DATA FROM TEXT
OpenCalais

It’s easy to forget and hard to believe that Low have been doing this for 20 years now. Alan Sparhawk, Mimi Parker, and John Nichols started the band and Galaxie 500 but eventually coming to represent the slowcore sub-genre by dint of sheer longevity. Those two acts have already reissued their entire catalogs, but Low have kept on keeping on. Still, slowcore is an atrocity, if only because it’s defined by its self-imposed limitation: at its best, that slow tempo can be as intense as a punk song; as boring as wallpaper at its worst. Having hit both extremes at once on the planet have owned up to the restrictions of their aesthetic.

As a result, their catalog is not necessarily diverse, but they’ve managed to find a great deal of variety within this small patch of terrain, thanks primarily to the extended nature of the band. Sparhawk and Parker have proved mainstays, with numerous musicians caught in their orbit over time (the latest being instrumentalist Steve Garrington). Furthermore, Low have never shied away from working with producers who bring their own specific styles and a Steve Fisk, and Steve Albini helped them define and refine their sound; in the next decade, Dave Fridmann and Matt Beckley helmed albums across as a blank canvas, and Low savored these outside perspectives, as though they knew the band was never quite a self-contained unit.

• Not easily customised/extended
• Domain-specific coverage varies

Source: http://viewer.opencalais.com/
It's easy to forget and hard to believe that Low have been doing this for 20 years now. Alan Sparhawk, Mimi Parker, and John Nichols started the group in 1993, following in the wake of Codeine and Galaxie 500 but eventually coming to represent the slowcore sub-subgenre by dint of sheer longevity. Those two acts have already reissued their entire catalog, keeping on. Still, slowcore is an odd sound to buoy such a long career and sturdy catalog, if only because it's defined by its self-imposed limitation: at its best, that slow tempo can be as intense as a punk song; as boring as wallpaper at its worst. Having hit both extremes at one point or another, the most perfectly named band on the planet have owned up to the restrictions of their aesthetic. As a result, their catalog is not necessarily diverse, but they've managed to find a great deal of variety within this small patch of terrain, thanks primarily to the open-ended nature of the band. Sparhawk and Parker have proved mainstays, with numerous musicians caught in their orbit over time (the latest, Arrington). Furthermore, Low have never shied away from working with producers who bring their own specific styles and aesthetics to their sound. In the late 1990s, Kramer, Steve Fisk, and Steve Albini helped them define and refine their sound; in the next decade, Dave Fridmann and Matt Beckley helmed albums with varying success. All used the band's glacial dirges as a blank canvas, and Low savored these outside perspectives, as though they knew the band was never quite a self-contained unit.

- Not easily customised/extended
- Is currently only available for English

Source: http://dbpedia-spotlight.github.com/demo/
Zemanta

• Common problem with general purpose, open-domain semantic annotation tools
• Best results require bespoke customisation

Source: http://www.zemanta.com/demo/
GATE

• General Architecture for Text Engineering
• Free open-source (LGPL) framework and development environment
• Started 1996, large developer community
• Used worldwide by many organisations to build bespoke solutions; e.g. Press Association and the National Archive
• Information Extraction in many languages

http://www.gate.ac.uk/
Increases recall over DBpedia by deriving new lexicalisations for URIs from link anchor texts, disambiguation pages, and redirect pages.
Precision and Recall

• Generic services typically very low recall
• Combination is one solution

<table>
<thead>
<tr>
<th></th>
<th>PER</th>
<th>LOC</th>
<th>ORG</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Spotlight</td>
<td>0.97 / 0.40</td>
<td>0.82 / 0.46</td>
<td>0.86 / 0.31</td>
<td>0.85 / 0.39</td>
</tr>
<tr>
<td>Zemanta</td>
<td>0.96 / 0.84</td>
<td>0.89 / 0.62</td>
<td>0.82 / 0.57</td>
<td>0.90 / 0.68</td>
</tr>
<tr>
<td>LODIE</td>
<td>0.81 / 0.82</td>
<td>0.73 / 0.76</td>
<td>0.56 / 0.59</td>
<td>0.71 / 0.74</td>
</tr>
<tr>
<td>Zemanta ∩ LODIE</td>
<td>1.00 / 0.74</td>
<td>0.95 / 0.45</td>
<td>0.97 / 0.42</td>
<td>0.97 / 0.54</td>
</tr>
<tr>
<td>Zemanta U LODIE</td>
<td>0.94 / 0.93</td>
<td>0.77 / 0.76</td>
<td>0.72 / 0.71</td>
<td>0.82 / 0.81</td>
</tr>
</tbody>
</table>

• Other solution is custom extraction
Custom GATE Gazetteer

- Retrieve MusicBrainz entity/label/class with SPARQL query
GATECloud

- Custom (e.g. based around custom gazetteer) GATE pipelines can be executed on the cloud:

  GATECloud Products

<table>
<thead>
<tr>
<th>Annotation Job - Custom</th>
<th>GBP0.00 (plus GBP0.99 per hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute your own pipeline on the GATE cloud.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANNIE (Named entity annotation service)</th>
<th>GBP0.00 (plus GBP1.49 per hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload your own documents and run our pre-packaged named entity annotator (ANNIE) on GATE Cloud</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANNIE plus Measurements and Numbers (annotation service)</th>
<th>GBP0.00 (plus GBP1.75 per hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload your documents and run our pre-packaged named entity annotator with the Numbers and Measurements add-ons</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GATE Teamware 1.4 Server (Enterprise)</th>
<th>GBP349.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web-based collaborative corpus and document annotation tool</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GATE Mimic 4.0 Server</th>
<th>GBP99.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiparadigm indexing server</td>
<td></td>
</tr>
</tbody>
</table>

Discounts:
- high volume
- research or non-profit use
- to apply create an account and send us your details
INTERLINKING DATA SETS WITH SILK
Interlinking with Silk

**Task**: Create links between the data set and external Linked Data sources.

**Approach**: Creation of specified links by querying the target data sets.

**Alternatives**:
- Manual creation of linkage rules by the user
- Automatic learning linkage rules by submitting predefined SPARQL queries
Link Discovery with Silk

• Open source tool for discovering RDF links between data items within different Linked Data sources

• It is based on the Silk Link Specification Language (Silk-LSL) for expressing linkage rules

• It accesses the target RDF data sets via SPARQL endpoints to generate RDF links

Source: Robert Isele. “LOD2 Webinar Series:Silk”
Silk Variants

• **Silk Single Machine**
  • Generates RDF links on a single machine
  • Data sets can reside either locally or in remote machines
  • Provides multithreading and caching

• **Silk MapReduce**
  • Uses a cluster composed of multiple machines
  • Based on Hadoop and designed to scale to big data sets

• **Silk Server**
  • Used within applications that consume Linked Data from the Web while keeping track of known entities
  • Provides an HTTP API for matching entities from an incoming stream

Source: http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/
Silk Workflow

Select LD data sets
- Identify suitable data sets in LD catalogs*
- Select the two data sets to link

Specify LD data sets
- Specify the access method to the data set (RDF dump, SPARQL endpoint)*
- Specify the entity types to be linked

Write linkage rule
- Specifies how to compare the resources
- Use Silk-LSL
- The rules can also be learnt

Generate RDF links
- Output links can be stored in a file or a triple store
- Can discover SKOS links

Source: Silk workflow is partially based on “LOD2 Webinar Series: Silk -(Simplified) Linking Workflow” by Rober Isele.
Linkage Rule Components

• Linkage rules define the conditions to create the links between the data sets. These rules are composed of:

1. RDF Paths
   • Describe the elements to be compared
   • Example: ?a/rdfs:label

2. Transformations
   • Apply transformations to the result set of an RDF path
   • Examples: LowerCase, Concatenate, Replace, ...

3. Comparators
   • Compute the similarity of two inputs
   • Examples: String similarity metrics, Date similarity, ...

4. Aggregations
   • Compute an aggregated value from multiple comparators
   • Examples: Min, Max, Avg, various means, Euclidian distance ...

Source: http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/
Silk Workbench

• Web application built on top of Silk, which allows the creation of projects to manage the creation of links between RDF data sets

• The data sets can be stored locally or accessed remotely by specifying the SPARQL endpoint

• The user is able to create customized linking tasks:
  – The tool offers a graphical editor to create linkage rules by combining the linkage rules components via drag & drop elements
  – Includes support for (automatic) learning linkage rules
Project configuration

1. **Project**: name and components (data sources, linking tasks and output tasks)

2. **Data sources**: specification of the data sets to be interlinked

3. **Linking task**: specification of the linkage rules and type of links to be created

4. **Output task**: mechanism to store the results from the linking process
1. **Linkage rule components**

2. **Graphical editor**: the items from (1) are dragged & dropped in this area, and connected to compose the linkage rules

3. **Generate links**: based on the defined linkage rules in (2), the data sets are accessed to discover possible links

4. **Learn**: automatic learning of linkage rules
Adding a linkage rule

The previous **linkage rule** states:

1. Retrieve the foaf:name values from MusicBrainz and the rdfs:label from DBpedia
2. Apply lower case transformation to the output of (1)
3. Compare the output from (2) using the metric “Levenshtein distance”. If this distance is greater than 0.90, then create a link.
Silk Workbench (5)

Generate Links

Overview
Executes the current linkage rule.
Based on its correctness, each link can be associated to one of the following 3 categories:

- **Correct**: Confirms the link as correct. Confirmed links are part of the positive reference link set.
- **Incorrect**: Confirms the link as incorrect. Incorrect links are part of the negative reference link set.
- **Unknown**: Link whose correctness is unknown.

<table>
<thead>
<tr>
<th>Source: MusicBrainz</th>
<th>Target: DBpedia</th>
<th>Score</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>'8699ce92-621f-4af6-67d2-107698e10606#_</td>
<td>dbpedia: 100_Monkeys</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>36c6626-6e2f-4930-8610-512e24636ed1#_</td>
<td>dbpedia: AC/DC</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>'140bed8-a19-4339-bb3b-5183b276f62#_</td>
<td>dbpedia: Anonymous_(band)</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>d52b08df-3be9-4926-24d-9d18143d8ebo#_</td>
<td>dbpedia: A_Band_Called_O</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>/2a2cbc483-6e4-442f-8e4-53ab31c738b7#_</td>
<td>dbpedia: Avantasia</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>1226a12-b654-4c59-8d55-2e113537eae9#_</td>
<td>dbpedia: Amusement_Parks_on_Fire</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>/e856b129-cafa-44da-9bd0-bf4435150552#_</td>
<td>dbpedia: Klubbheads</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>a674c06-e561-4151-80b9-52e59361448e#_</td>
<td>dbpedia: Dirty_Little_Rabbits</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>'05e4f3e-d005-4dad-8ec1-d04f521cb856#_</td>
<td>dbpedia: Def_Punk</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>:c197bed-dc9c-440d-e55b-5d2ba2e14234#_</td>
<td>dbpedia: Coldplay</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>/2b6f8b0-5680-407d-936b-5f02003688a#_</td>
<td>dbpedia: Casa_de_Lenones</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>35cc7e1-45cf-4c2a-d59c-37c7ca472070c#_</td>
<td>dbpedia: Comet_Gain</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>/3e5ef7b-ae2c-4167-9ef6-7354e03dcb5#_</td>
<td>dbpedia: Dazzling_Killmen</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
<tr>
<td>d3f9254c-08fe-4ba6-b601-8ad433d1b19#_</td>
<td>dbpedia: Diaframma</td>
<td>100.0%</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Silk Workbench (6)

Learn Rules

Learned Linkage Rule

Fitness: 99.5 (based on 0 positive and 0 negative reference links)
- Comparison: levenshteinDistance (0.6335144363574432)
  - Input: ?a/<http://xmlns.com/foaf/0.1/name>
- Transformation: tokenize
  - Input: ?b/<http://www.w3.org/2000/01/rdf-schema#label>

Uncertain links

Source: MusicBrainz Target: DBpedia
- 2407d3f9-a9ee-4e3b-b9f9-bc3271c94aad#_ dbpedia:Moke_(Amsterdam_band) Score: 62.4%
- ca30529-e9b1-4478-8455-c685974cb1d7#_ dbpedia:Kobukuro Score: 62.4%
- 0ca302fe-c064-4bf1-8247-e827a4c68a44#_ dbpedia:Minimori Score: 52.8%
For exercises, quiz and further material visit our website: http://www.euclid-project.eu

Other channels:

@euclid_project
euclidproject
euclidproject