LOD2 Stack Tutorial

A tutorial by
Sebastian Tramp, University of Leipzig
WebID: http://sebastian.tramp.name

with slides from
- Bert van Nuffelen, Tehforce
- Robert Isele, FU Berlin
- Hugh Williams, Openlink Software
- Katja Eck, Wolters Kluwer Germany
- Philipp Frischmuth, University of Leipzig
- Sören Auer, University of Leipzig
LOD2 PROJECT & STACK INTRODUCTION
Intelligent Information Management

Collaborative Project 2010-2014
in Information and Communication Technologies

Project No. 257943
Start Date 01/09/2010
A strong partnership
Creating Knowledge out of Interlinked Data

From the Web of Documents to the Semantic Data Web

**Data Web** (since 2006)
- URI de-referencability
- Web Data integration
- RDF serializations

**Social Web** (since 2003)
- Folksonomies/Tagging
- Reputation, sharing
- Groups, relationships

**Web** (since 1992)
- HTTP
- HTML/CSS/JavaScript

**Semantic Web** (Vision 1998, starting ???)
- Reasoning
- Logic, Rules
- Trust
Creating Knowledge out of Interlinked Data

The Emerging Web of Data: Achievements and Challenges

- Web - a global, distributed platform for data, information and knowledge integration
- exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF

Achievements
1. Extension of the Web with a data commons (currently amounting to 25 Bn. facts)
2. Vibrant, global RTD community
3. Industrial uptake begins (e.g. BBC, Thomson Reuters, Eli Lilly)
4. Emerging governmental adoption in sight
5. Establishing Linked Data as a deployment path for the Semantic Web.

Challenges
1. Coherence: Relatively few, expensively maintained links
2. Quality: partly low-quality data and inconsistencies
3. Performance: Still substantial penalties compared to relational
4. Data Consumption: large-scale processing, schema mapping and data fusion still in its infancy
5. Usability: Missing direct end-user tools and network effect

These issues are closely related and should ultimately lead to an ecosystem of interlinked...
Creating Knowledge out of Interlinked Data

LOD2 in a Nutshell (1)

Research Focus
- very large RDF data management
- knowledge enrichment & interlinking
- fusion & information quality
- adaptive, semantic user interfaces

Use Cases
- Media & Publishing
- Enterprise Data Webs
- Open Gov Data
- Public Sector Contracts

Main Result
- integrated LOD2 Stack for Linked Data lifecycle management

Partners
ULEI, CWI, NUIG, FUB/UMA, SWCG, OGL, Tenforce, Exalead, WKD, SKFN, I2G, IMP, KAIST

http://lod2.eu
**LOD2 in a Nutshell (2)**

**LOD2**
EC-funded collaborative project that aims to utilize the Web as an integration platform for data and information.

**Linked Data**
Linked Data provides the necessary basic technologies and standards to realize the goal of LOD2.

**Linked Open Data**
Publicly accessible data which is to be integrated into the web and linked among one another and with non-public contents such as enterprise intranets.

**Project Highlights**
Open Government Linked Data Initiative
Common European platform [publicdata.eu](http://publicdata.eu)

Leading **Web 3.0 technologies** are combined in the project into the coherent LOD2 Stack (e.g. DBpedia, Virtuoso, Sindice, Silk).
Objectives of LOD2

- LOD2 project objectives:
  - Increase visibility of Linked Data activities
  - Improve the software technology which support it
  - Support deployment Linked Data components
  - Improve information sharing between Linked Data components so that publishing Linked Data is eased.
  - Improve access to the content: the online Linked Open Data.

- Core enabler and end-user accessible result: the LOD2 stack
LOD2, a FP7 EU funded project
AKSW @ University of Leipzig

Agile Knowledge Engineering and Semantic Web
Founded in 2006
AKSW aims:

- Contributing to the advancement of science in Semantic Web, Knowledge Engineering, Software Engineering
- Cost efficient, high-impact R&D, which proves usefulness at an early stage
- Bridge the gap between research results and applications
- 40+ researchers
LOD2 STACK

EASY ACCESS TO LINKED DATA SOFTWARE
LOD2 stack repository is a Debian package repository
http://stack.lod2.eu

We have chosen a new reference OS: **Ubuntu12.04 LTS**
This version is supported for the next 5 years.

Changes in repository management system for
enabling quality control (development -> test -> stable)
enabling architecture dependent distribution support (e.g. Virtuoso RDF store)

Public access to documentation
http://wiki.lod2.eu

Releases
Intermediate release 1.1 (more components)
Release 2.0 (in preparation - October 2012)
LOD2 stack contribution process

- Component
- Debian Package
- LOD2 Stack testing
- Valid?
- LOD2 Stack Stable

Component Owner

LOD2 stack maintainer

- Does the Debian package work?
- Does it break flows | UI?
## LOD2 Stack Progress Statistics

**Release 1:** 25 components/Packages

**Release 2:** 37 components/Packages
- 10 **new** components/Packages
- 28 have been subject to LOD2 stack **integration** efforts

**Release 3:** released 05/2014
- A lot of upgraded packages
- New base distribution (14.04 LTS)
- Some new tools
- CKAN datahub integration (more than 100 dataset packages)
- Transition of responsibility to the GeoKnow project
- New home: http://stack.linkeddata.org
Linked Data publishing capabilities currently offered

- Covers most of the LOD publishing cycle
- Combination of
  - locally installed software,
  - online available software, and
  - online available data sources as well as data packages
  - about page in the LOD demonstrator (http://demo.lod2.eu/lod2demo)

- Disclaimer. No harmonized user interface.
LOD2 Stack – OntoWiki
OntoWiki
Creating Knowledge out of Interlinked Data

OntoWiki Paradigm: Resources over Articles

[Image of OntoWiki interface and data visualization]
OntoWiki Paradigm: Forms over Markup
1. Generic data wiki for RDF models
   - No data model mismatch (structured vs. unstructured)

2. Application framework:
   - Knowledge-intensive applications,
   - Agile processes,
   - Distributed user groups
OntoWiki Core Features

• Knowledge Bases (aka. graphs, Linked Data optional)
• Generic list and resource views
• Versioning
• Commenting on arbitrary resources
• User management + access control
• Inline editing
• Navigation hierarchies (e.g. Class hierarchies)
OntoWiki Core Features (cont.)

- Search
- Linked Data Server
- Linked Data Client
- Import / Export of RDF/XML, Turtle, RDF/JSON
- View / Import arbitrary resources
- Filtering
- SPARQL editor
OntoWiki Interfaces

- SPARQL Endpoint
- Linked Data Endpoint
- REST API
- Command Line Interface
OntoWiki Architecture

- RDFauthor
- Extensions
- OntoWiki Application Framework
- Erfurt API
- Zend Framework
OntoWiki Extensibility

• Components, Plugins, Modules, Wrapper
• Views/Templates
• Themes
• Localizations
Demonstration 1
Basics
## OntoWiki Requirements

- **Apache**
  - nginx requires some work
  - IIS reported as working (never tested)
- **Virtuoso (recommended)** or MySQL (small KBs)
- **PHP 5.2+**
- **Current version of**
  - Google Chrome, Safari, Firefox or MSIE
- **Zend >= 1.5.0**
OntoWiki Installation

- Download or clone
  https://github.com/AKSW/OntoWiki/
- Extract to web documents folder
- Copy config.ini-dist to config.ini
- Adjust DB params
- Point your browser to the OW folder
  e.g. http://localhost/ontowiki
OntoWiki Support

- **Users:**
  - http://ontowiki.net/Projects/OntoWiki
  - ontowiki-user@googlegroups.com
  - https://github.com/AKSW/OntoWiki/wiki

- **Developer:**
  - https://github.com/AKSW/OntoWiki
  - ontowiki-dev@lists.informatik.uni-leipzig.de

- **LOD 2 Stack:** support-stack@lod2.eu
Q & A

• What Vocabularies does Ontowiki support?
  Any schema or vocabulary
  What is “support for a Vocabulary”? 
  OntoWiki is schema independent?
  Can be configured and extended for specific vocabularies?

• What is the typical use case for using Ontowiki?
  Collaborative authoring of instance data (data wiki)
  Visualization of statistical data (cubeviz extension)
  Deployment of website (site extension)

• Can Ontowiki work like a Wiki with forms that interact with graphs?
  Yes. This is exactly what OntoWiki is :)
  (the form part is named RDFauthor)
Creating Knowledge out of Interlinked Data

How to do the following in Ontowiki:

- **Create an identifier**
  - Just type it in the resource bar
  - Just type it in the browser (needs configuration)
  - Click on “Create Instance” in the context menu of a class

- **Use / connect a vocabulary**
  - Create a Knowledge Base with the content (upload)
  - Special Support: Select a Vocabulary Module (LOV directory)
  - Special Support: Linked Data Stack dataset packages
  - owl:import it to your instance data

- **Connect to an rdfs type => what happens with the linked vocabularies**
  - ??? Not sure what this means

- **How to create new Types**
  - “Create Instance” of owl:Class
  - Use the source luke (turtle editor)
How to do the following in Ontowiki:

• How to create new vocabularies
  Create an empty Knowledge Base
  Configure namespaces (add OWL)
  Import OWL or RDFS

• Connect to other wiki's (I assume wiki -> OntoWiki)
  via Linked Data gathering
  via Semantic Pingback

• Connect to other datasources (endpoints)
  feature/remoteSparqlEndpoint branch
  (not merged, experimental)
How to do the following in Ontowiki:

• How to query the wiki data
  Create specific views with the filter extension
  Use the query shell and save queries for later

• How to get additional data from other wiki’s
  Linked Data wrapper
  Implement a custom wrapper

• How to create lists/overviews
  see How to query data?

• How to visualise the query results
  as OntoWiki list view?
Further questions:

What were the reasons to include different authoring tools in the LOD2 stack (e.g. PoolParty and OntoWiki)?
- The stack is open to everyone and every tool
- Poolparty and OW provide different feature sets

What OntoWiki user, developer and administrator documentation is available as a further reference for us?
- The wiki at github.com is currently the best source
- https://github.com/AKSW/OntoWiki/wiki

Is OntoWiki entirely forms-based? Or is it also possible to create ‘free-form’ Wiki pages like in a standard Wiki?
- The article extension provides basic plain text wiki support
- https://github.com/AKSW/article.ontowiki
Further questions:

How would you use SKOS elements to structure an OntoWiki implementation? What are known best practices?
- Import it via owl:imports
- Use and adapt the SKOS navigation hierarchy scheme
- Link to your SKOS concepts with SKOS relations

What if the resource ID’s in OntoWiki must comply to a local URI strategy, how to implement that in OntoWiki?
- Use the resourcecreationuri (automatically used)
- Adapt it if it does not fit your needs
- defaultNamingScheme = "BASE/type/label"

What extensions are available for OntoWiki? And which extensions are used most often by the OntoWiki users?
- 20+ external extensions such as files, site, cubeviz, csvimport, map,
Further questions:

How would you handle a multi-language requirement in OntoWiki for e.g. for a Dutch Wiki and an English Wiki?
- Create a dutch localization
- Extras -> Select Language

Which Virtuoso versions and open/closed source variants does OntoWiki support?
- Currently working under 6.1.4/5/6 and 7.1.0
- Commercial version 7.1.0 in use
LOD2 Stack – Silk Link Discovery Framework
Motivation

The Web of Data is a single global data space because data sources are connected by links
Over 30 billion triples published as Linked Open Data (09/19/2011)
But:
- Less than 500 million links
- Most publishers only link to one other dataset

LOD data sets by the number of other data sources that are target of outgoing RDF links.
Challenges for Link Discovery

The Web of Data is heterogeneous
• Many different vocabularies are in use
• Different data formats
• Many different ways to represent the same information

Distribution of the most widely used vocabularies
Challenges for Link Discovery

Large range of domains
- 277 data sources in the LOD cloud from a variety of domains
- Linkage Rules are different in each domain
- Writing a Linkage Rule is for each of these domains is usually not trivial

Distribution of triples by domain
Challenges for Link Discovery

- **Scalability**
  - The current LOD cloud contains 277 datasets (August 2011)
  - Over 31 billion triples in total
  - Infeasible to compare every possible entity pair

<table>
<thead>
<tr>
<th>Domain</th>
<th>Number of datasets</th>
<th>Triples</th>
<th>%</th>
<th>(Out-)Links</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>27</td>
<td>1,855,413,060</td>
<td>5.88%</td>
<td>50,469,665</td>
<td>10.10%</td>
</tr>
<tr>
<td>Geographic</td>
<td>26</td>
<td>6,111,263,253</td>
<td>19.36%</td>
<td>35,751,295</td>
<td>7.16%</td>
</tr>
<tr>
<td>Government</td>
<td>45</td>
<td>13,302,470,684</td>
<td>42.14%</td>
<td>19,353,426</td>
<td>3.87%</td>
</tr>
<tr>
<td>Publications</td>
<td>86</td>
<td>2,958,969,764</td>
<td>9.37%</td>
<td>135,925,930</td>
<td>27.21%</td>
</tr>
<tr>
<td>Cross-domain</td>
<td>36</td>
<td>4,157,191,654</td>
<td>13.17%</td>
<td>62,805,095</td>
<td>12.57%</td>
</tr>
<tr>
<td>Life sciences</td>
<td>42</td>
<td>3,042,142,230</td>
<td>9.64%</td>
<td>191,825,949</td>
<td>38.40%</td>
</tr>
<tr>
<td>User-generated content</td>
<td>14</td>
<td>115,072,057</td>
<td>0.36%</td>
<td>3,431,983</td>
<td>0.69%</td>
</tr>
<tr>
<td><strong>LOD datasets per domain</strong></td>
<td></td>
<td><strong>31,568,522,702</strong></td>
<td></td>
<td><strong>499,564,104</strong></td>
<td></td>
</tr>
</tbody>
</table>
Tools enable data publishers to set links

Most tools generate links based on user-defined linkage rules

A linkage rule specifies the conditions data items must fulfill in order to be interlinked

Popular Link Discover Tools:
• Silk Link Discovery Framework
• LIMES
Silk Link Discovery Framework

Tool for discovering links between data items within different Linked Data sources.

The Silk Link Specification Language (Silk-LSL) allows to express complex linkage rules

Can be used to generate owl:sameAs links as well as other relationships

Scalability and high performance through efficient data handling
Silk Versions

Silk Single Machine
- Generate links on a single machine
- Local or remote data sets

Silk MapReduce
- Generate RDF links using a cluster of multiple machines
- Based on Hadoop (usable with Amazon Elastic MapReduce)

Silk Server
- Provides an HTTP API for matching instances from an incoming stream of RDF data
- Can be used as an identity resolution component within applications that consume Linked Data from the Web
(Simplified) Linking Workflow

Select Datasets
- Select two data sources
- Select the entity types to be interlinked

Write Linkage Rule
- Specifies how two entities are compared
- Can be written manually or learned

Generate Links
- Locally or on a Hadoop Cluster
- Write Links to file or a triple store
Creating Knowledge out of Interlinked Data

Linkage Rule Components

A linkage rule is represented as a tree consisting of 4 types of operators:

- **RDF paths**
  - Similar to SPARQL 1.1 Property Paths
  - Examples:
    - `?movie/dbpedia:director/rdfs:label`
    - `?person/label[@lang='en']`

- **Transformations**
  - Transforms the result set of an RDF paths
  - Variety of built-in transformations
  - Examples:
    - LowerCase
    - RegexReplace
    - Stem

- **Similarity Metrics**
  - Similarity of two inputs based on a user-defined metric.
  - Examples:
    - Various string similarity metrics
    - Geographic similarity
    - Date similarity

- **Aggregations**
  - Aggregates multiple similarity metrics
  - Examples:
    - Min, Max, Average
    - Quadratic Mean
    - Geometric Mean
Example: Interlinking cities

: Aggregation
  function: minimum

: Comparison
distanceMeasure = Levenshtein
  threshold = 1

: Transformation
  func. = lowerCase

: Property
  property = label

: Comparison
distanceMeasure = wgs84
  threshold = 1

: Transformation
  func. = lowerCase

: Property
  property = coords

: Property
  property = coords
Silk Workbench

Silk Workbench is a web application which guides the user through the process of interlinking different data sources.

Enables the user to manage sets of data sources and linking tasks.

Offers a graphical editor which enables the user to easily create and edit linkage rules.

Offers tools to evaluate the current linkage rule.

Includes support for learning linkage rules.
Creating Knowledge out of Interlinked Data

Workspace

The Workspace holds a set of projects consisting of:

Data Sources
• Holds all information that is needed by Silk to retrieve entities from it.
• Usually a file dump or a SPARQL endpoint

Linking Tasks
• Interlinks a type of entity between two data sources e.g. Interlinking movies in Dbpedia and LinkedMDB
Linkage Rule Editor

Allows to view and edit linkage rules
Linkage Rules are shown as a tree
Editing using drag & drop.
Learning Linkage Rules

Linkage Rules can be learned interactively
Can be used to generate new linkage rules or to improve existing rules
Learned Linkage Rule can be viewed and edited by the user
Availability

Silk can be downloaded from the official homepage at:

http://www4.wiwiss.fu-berlin.de/bizer/silk/

Support is provided through the official mailing list:

http://groups.google.com/group/silk-discussion

The latest source code is available from the project's Git repository and can be browsed online at:

http://www.assembla.com/code/silk/git/nodes/

Silk is licensed under the terms of the Apache Software
Demonstration

Interlinking movies between two data sources

DBpedia is a dataset extracted from Wikipedia, LinkedMDB is a large dataset for movies. This demonstration shows, how different properties can be used together in order to create sameAs links between these two datasets.

For demonstration, we assume that no existing links are available.
Core Platform
(Graph DBMS and
Linked Data
Deployment) behind
DBpedia, many
bubbles in the LOD
Cloud, and the LOD
Cloud cache itself.
## Virtuoso Linked Data projects snapshot

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBpedia</strong></td>
<td>public SPARQL endpoint over the DBpedia data (and <strong>international Chapters</strong>)</td>
</tr>
<tr>
<td><strong>LOD Cloud Cache</strong></td>
<td>public server hosting <strong>LOD cloud</strong> datasets</td>
</tr>
<tr>
<td><strong>URIBurner</strong></td>
<td><strong>Linked Data</strong> generation &amp; transformation service</td>
</tr>
<tr>
<td><strong>Linked Geo Data</strong></td>
<td>OpenStreetMap Spatial data as <strong>Linked Data</strong></td>
</tr>
<tr>
<td><strong>Sindice</strong></td>
<td>SPARQL endpoint behind its Semantic Web Index</td>
</tr>
<tr>
<td><strong>Data.gov</strong></td>
<td>US Government <strong>Linked Data</strong></td>
</tr>
<tr>
<td><strong>Health.data.gov</strong></td>
<td>Clinical Quality <strong>Linked Data</strong> on <strong>health.data.gov</strong></td>
</tr>
<tr>
<td><strong>Seevl</strong></td>
<td><strong>Linked Data</strong> music discovery service</td>
</tr>
<tr>
<td><strong>Bio2RDF</strong></td>
<td>Life science data mapped to <strong>Linked Data</strong></td>
</tr>
<tr>
<td><strong>Neurocommons</strong></td>
<td>Life science data mapped to <strong>Linked Data</strong></td>
</tr>
<tr>
<td><strong>Musicbrainz</strong></td>
<td>MusicBrainz database published as <strong>Linked Data</strong></td>
</tr>
<tr>
<td><strong>Many others</strong></td>
<td>…</td>
</tr>
</tbody>
</table>

LOD2 Stack Tutorial . Page 58  http://lod2.eu
Enterprise and Individual Agility via Data Virtualization, without compromising performance, scalability, and security.
Why is Virtuoso Important to LOD2

Linked Data Deployment modulo the following challenges

- De-referencable URI complexities
- URI style (hash or slash) distractions
- Loose Coupling of Information and Data
- SPARQL endpoint commissioning
- Linked Data Views over Relational Data (incl. R2RML support)
- Faceted Browsing
- Proven Performance & Scalability.
Data Virtualization Middleware

An in-built middleware layer ("Sponger") for creating Transient & Persistent Views over Heterogeneous Data Sources.
A collection of prefabricated and customizable Data Extraction, Transformation, and Lookup cartridges (drivers) covering a vast ranges of data formats and data access protocols.
Sophisticated Content Crawler

DBMS hosted Content Crawler that’s leverages loosely coupled binding to the Sponger Middleware component for transformation of unstructured and semi-structured data into Linked Data.
**Insight Discovery & Exploration**

Native Faceted Browsing that enables multi-dimensional drill-downs via any browser.
Creating Knowledge out of Interlinked Data

Powerful SPARQL Query Service

Basic SPARQL Endpoint for Creating Query Definitions & Sharing Query Results.

Virtuoso SPARQL Query Editor

```
PREFIX oplli: <http://www.openlinksw.com/schemas/linkedin#>

SELECT DISTINCT ?e1 as ?job_id ?provided_by ?location_name ?industry ?country_code ?company_name
    ?position_summary ?experience_level ?job_type
    ?job_salary ?label as ?name ?job_function
    ( "<VOD:IRI_RANK>" ( ?e1 ) ) as ?entity_rank

WHERE {
    ?e1 a oplli:JobPosting;
    oplli:providedBy ?provided_by;
    oplli:location_name ?location_name;
    oplli:industry ?industry;
    oplli:country_code ?country_code;
    oplli:company_name ?company_name;
    oplli:position_summary ?position_summary;
    oplli:experience_level ?experience_level;
    oplli:job_type ?job_type;
    oplli:salary ?job_salary;
    rdf:label ?label;
    oplli:job_function ?job_function.
    filter ( ?job_function = "Consulting" ) .
    filter ( ?job_type = "Full-time" ) .
    filter ( ?industry in ( "Internet", "Computer Software", "Information Services", "Information Technology and Services" ) ) .
}
```
Use Query By Example (QBE) Patterns to **Construct** & **Share** Query Results.
Demonstration

Loading CKAN LOD Datasets into Virtuoso and query as Linked Data

This demonstration shows how the LOD2 Stack can be used for Loading CKAN Linked Open Data datasets which are part of the LOD2 Stack into the Virtuoso Quad Store resulting in the automatic deployment of the loaded datasets as Linked Data by Virtuoso enabling them to be discovered, traversed and navigated using Linked Data tools.
Thank you!

LOD2 stack tutorial by

Sebastian Tramp, University of Leipzig
WebID: http://sebastian.tramp.name

with slides from
- Bert van Nuffelen, Tehforce
- Robert Isele, FU Berlin
- Hugh Williams, OpenLink Software
- Katja Eck, Wolters Kluwers Germany
- Philipp Frischmuth, University of Leipzig
- Sören Auer, University of Leipzig