Tutorial Agenda

- Introduction to Linked Data (45 m – 60 m) Andreas
- Consuming Norwegian Linked Data (30 m) Titi
- Large Scale Linked Data Management (30 m) Andreas
- Big Data Intro and Analytics (60 m – 90 m) Marko
- Questions & Answers Session (30 m) all
Introduction to Linked Data (Andreas)

- Motivation
- Linked Data
  - Principles (Web Architecture and RDF, Resource Description Framework)
  - SPARQL RDF Query Language
- Ontology Languages
  - RDF Vocabulary Description Language (RDFS)
  - Web Ontology Language (OWL)
- Application Architectures
- Summary
MOTIVATION
Motivation

- With increased use of computers more and more data is being stored
  - Organisations rely on data for business decisions
  - Data drives policy decisions in government
  - Individuals rely on data from the Web for information and communication

- Data volumes explode
  - More and more data available on the Web is represented in Semantic Web standards
  - Linking Open Data (LOD) initiative

- Semantic Web technologies facilitate the integration of data from multiple sources
- Combining data from multiple sources enables insights
Linked Data on the Web
Linked Data on the Web

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Linked Data on the Web

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Linked Data on the Web

2008-09

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Linked Data on the Web

As of July 2009

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Linked Data on the Web

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Types of Data in the Linking Open Data Cloud

http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (Sept 2010)
Scenario Overview

- Semantic Technologies facilitate access to data

1. Query
2. Answer

- Q: data about Berlin?
- Q: famous people that died in Berlin?
- Q: data about Hegel?
- Q: Hegel’s publications?
- Q: data about Marlene Dietrich?
- Q: Dietrich’s songs?
DBpedia

- Linked Data version of Wikipedia
- Scripts that extract data (text, links, infoboxes) from Wikipedia
- Published as Linked Data
- Interlinking hub in the Linked Data web

- Berlin
  - http://dbpedia.org/resource/Berlin
- Hegel
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- Marlene Dietrich
  - http://dbpedia.org/resource/Marlene_Dietrich
BBC Music

- Data about BBC (radio) programmes, artists, songs…
- Combination of BBC-internal data (playlists), MusicBrainz (artists, albums), Wikipedia (artists)
- Underpinning the BBC Music website
- Data published according to Linked Data principles

Marlene Dietrich

- http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf#artist
Virtual International Authority File (VIAF)

- Joint project of national libraries and related organisations
  - 21 institutions, among them the Library of Congress, Deutsche Nationalbibliothek, Bibliothèque nationale de France

- Provide access to “authority files”

- Matching and interlinking collections from participating institutions

- Hegel
  - http://viaf.org/viaf/89774942/

- Marlene Dietrich
  - http://viaf.org/viaf/97773925/
LINKED DATA PRINCIPLES
Semantic Web technologies, standardised by the W3C, are mature:
- RDF recommendation in 1999, update in 2004
- RDFa (RDF in HTML) note in 2008
- RDFS recommendation in 2004
- SPARQL recommendation in 2008
- OWL recommendation in 2004, update in 2009

Linked Data is a subset of the Semantic Web stack, including web architecture:
- IRI (IETF RFC 3987, 2005)
- HTTP (IETF RFC 2616, 1999)
Linked Data Principles

1. Use URIs as names for things
2. Use HTTP URIs so that people 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 they can discover more things.

http://www.w3.org/DesignIssues/LinkedData
1. Use URIs as Names for Things

- Use a unique identifier to denote things
- URIs are defined in RFC 2396

- Hegel, Georg Wilhelm Friedrich
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
  - http://viaf.org/viaf/89774942/
  - ...

- Hegel, Georg Wilhelm Friedrich: Gesammelte Werke / Vorlesungen über die Logik
  - urn:isbn:978-3-7873-1964-0
Names for Things

“Now! That should clear up a few things around here!”
2. Use HTTP URIs

- Enables “lookup” of URIs
- Via Hypertext Transfer Protocol (HTTP)
- Piggy-backs on hierarchical Domain Name System to guarantee uniqueness of identifiers
- Uses established HTTP infrastructure
- Connects logical level (thing) with physical level (source)
- Important: distinction between “thing URI” and “source URI” („other resource“ vs. „information resource“)
Information Resources vs. Other Resources

Marlene Dietrich, the person

File containing data about Marlene Dietrich

Name?
Creator?
Birth date?
Last change date?
License?
Copyright?
...

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Correspondence between thing-URI and source-URI („hash URIs“)

User Agent

HTTP GET

Web Server

RDF

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf#artist

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
$ curl -H "Accept: application/rdf+xml" -v http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf#artist

GET /music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf HTTP/1.1
User-Agent: curl/7.25.0
Host: bbc.co.uk
Accept: application/rdf+xml

HTTP/1.1 200 OK
Date: Tue, 08 May 2012 07:12:19 GMT
Server: Apache/2.2.3 (Red Hat)
Content-Type: application/rdf+xml
Content-Length: 1956

{ [data not shown] }
Correspondence between thing-URI and source-URI („slash URIs“)

User Agent

HTTP GET 303 HTTP GET RDF

Web Server

http://dbpedia.org/resource/Marlene_Dietrich

http://dbpedia.org/data/Marlene_Dietrich

http://dbpedia.org/page/Marlene_Dietrich

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
3. Provide Useful Information

- When somebody looks up a URI, return data using the standards (RDF*, SPARQL)
- Resource Description Framework, a format for encoding graph-structured data (with URIs to identify nodes/vertices and links/edges)
Resource Description Framework

- Directed, labeled graph
- \text{triple}(\text{subject}, \text{predicate}, \text{object})
  - subject: URI (or blank node)
  - predicate: URI
  - object: URI (or blank node) or RDF literal (string, integer, date…)

- RDF/XML is the most widely deployed serialisation
- Other serialisations possible (N-Triples, Turtle, Notation3…)

- Quadruples (or quads) used as internal representation when integrating data
  - \text{quad}(\text{subject}, \text{predicate}, \text{object}, \text{context})
    - context: URI (used to store origin of triple)
RDF Example

dbpedia:Georg_Wilhelm_Friedrich_Hegel rdf:type foaf:Person .
dbpedia:Georg_Wilhelm_Friedrich_Hegel rdf:type yago:PoliticalPhilosophers .
dbpedia:Georg_Wilhelm_Friedrich_Hegel rdfs:comment "Georg Wilhelm Friedrich Friedrich Hegel var en tysk filosof."@no .

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Merging Data with RDF

http://example.org/smith#albert

http://example.org/smith#child

http://example.org/smith#brian

+ 

http://example.org/muller#dora

http://xmlns.com/foaf/0.1 knows

http://example.org/smith#albert

= 

http://example.org/smith#dora

http://xmlns.com/foaf/0.1 knows

http://example.org/smith#albert

http://example.org/fam#child

http://example.org/smith#brian

http://example.org/fam#child

http://example.org/smith#carol

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
4. Link to Other URIs

- Enable people (and machines) to jump from server to server
- External links vs. internal links (for any predicate)
- Special owl:sameAs links to denote equivalence of identifiers (useful for data merging)
Equivalences via owl:sameAs

http://viaf.org/viaf/89774942/
- http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- http://www.idref.fr/026917467/id
- http://libris.kb.se/resource/auth/190350
- http://d-nb.info/gnd/118547739

http://www.bbc.co.uk/music/artists/191c6a-b83f-49ca-883c-02b20c7a9dd5#artist
- http://dbpedia.org/resource/Marlene_Dietrich

http://viaf.org/viaf/97773925/
- http://dbpedia.org/resource/Marlene_Dietrich
- http://d-nb.info/gnd/118525565
- http://libris.kb.se/resource/auth/238817
- http://www.idref.fr/027561844/id

http://dbpedia.org/resource/Berlin
- http://mpii.de/yago/resource/Berlin
- http://data.nytimes.com/N50987186835223032381 - Berlin (Germany)
- http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Berlin
- http://data.nytimes.com/16057429728088573361 - Gaspe Peninsula (Quebec) (?)
SPARQL RDF PROTOCOL AND QUERY LANGUAGE
SPARQL

- SPARQL Protocol and RDF Query Language
- Query language for RDF graphs
- “SQL for RDF”

SPARQL specification consists of
- Query language
- Result formats (representation of results in RDF and XML)
- Query protocol (mechanisms to pose queries and retrieve results)
Simple Query Example

PREFIX dct: <http://purl.org/dc/terms/>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
SELECT *  
WHERE {  
  ?s rdfs:label ?name.  
}  

- Main part is query pattern (WHERE clause)  
  - Using Turtle syntax for RDF  
  - Query patterns may contain variables (?s, ?name)  
- Shortcuts for URIs (PREFIX)  
- Query results via selection of variables (SELECT)
## Query Results

- **Table with one row per result**

<table>
<thead>
<tr>
<th>?s</th>
<th>?name</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dbpedia.org/resource/Erik_Nevland">http://dbpedia.org/resource/Erik_Nevland</a></td>
<td>&quot;Erik Nevland&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Jan_Simonsen">http://dbpedia.org/resource/Jan_Simonsen</a></td>
<td>&quot;Jan Simonsen&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Laila_Goody">http://dbpedia.org/resource/Laila_Goody</a></td>
<td>&quot;Laila Goody&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Henriette_Henriksen">http://dbpedia.org/resource/Henriette_Henriksen</a></td>
<td>&quot;Henriette Henriksen&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Guri_Hjeltnes">http://dbpedia.org/resource/Guri_Hjeltnes</a></td>
<td>&quot;Guri Hjeltnes&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Johan_E._Holand">http://dbpedia.org/resource/Johan_E._Holand</a></td>
<td>&quot;Johan E. Holand&quot;@no</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Kristian_Valen">http://dbpedia.org/resource/Kristian_Valen</a></td>
<td>&quot;Kristian Valen&quot;@no</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Further Functionality

- Optional triple patterns (e.g., return name and optionally birthdate if available)

- Unions (e.g., return material scientists and also physicists)

- Filter (e.g., only return scientists born before 1970)

- Result formats (e.g., return RDF triples instead of results table)

- Modificators (e.g., sort results, only return unique results)
Benefits of Linked Data

- Explicit, simple data representation
  - Common data representation (Resource Description Framework, RDF) hides underlying technologies and systems
- Distributed System
  - Decentralised distributed ownership and control facilitates adoption and scalability
- Cross-referencing
  - Allows for linking and referencing of existing data, via reuse of URIs
- Loose coupling with common language layer
  - Large scale systems require loose coupling, via HTTP as common access protocol
- Ease of publishing and consumption
  - Simple and easy-to-use systems and technologies to facilitate uptake
- Incremental data integration
  - Start with merged RDF graphs and provide mappings as you go
Challenges (I)

- Ramp-up cost for data conversion
  - May be alleviated by semi-automatic mappings and adequate tool support for manual conversion

- Integrated data may be messy at first
  - But can be refined as need arises

- Distributed creation and loose coordination may result in inconsistencies
  - Can be detected, diagnosed, and fixed with appropriate tools
The Pedantic Web Group

- Get the community to contact publishers about errors/issues as they arise
- Get involved: http://pedantic-web.org/
- 137 members!

Acknowledgements to: Aidan Hogan, Alex Passant, Me, Antoine Zimmermann, Axel Polleres, Michael Hausenblas, Richard Cyganiak, Stéphane Corlosquet
Challenges (II)

- Often very much oriented towards individuals
- Little possibilities for expressing schema knowledge
- Different data sources have different ways of representing the same facts

- Ontology languages (RDFS, OWL) solve that drawback
- RDFS and OWL are layered on top of RDF
ONTOLOGY LANGUAGES
Ontology in Philosophy

- Term exists only in singular (there are no “ontologies”)

- Ontology is concerned with the study of the nature of being, existence or reality as such

- Discussed by Aristoteles (Sokrates), Thomas von Aquin, Descartes, Kant, Hegel, Wittgenstein, Heidegger, Quine, ...
Ontology in Informatics

“An Ontology is a formal specification of a shared conceptualisation of a domain of interest”

Schema Knowledge

- RDF provides universal mechanism for the representation of facts using triples
- Possible to describe individuals and their relations
- Required: describe generic sets of individuals (classes), e.g., people, chemical compounds, organisations…
- Required: specification of logical connections between individuals, classes and properties to describe their meaning, e.g., “researchers write papers”, “materials are chemical compounds”
- In database-speak: schema knowledge
Schema Knowledge with RDFS

- RDF Vocabulary Description Language (RDFS)
- Allows for specification of schema (also: terminological) knowledge
- RDFS is a special RDF vocabulary (every RDFS document is an RDF document)
- RDFS vocabulary is generic: allows to specify the semantics of other vocabularies (and as such is a kind of “metavocabulary”)
- Thus, RDFS is an ontology language (but a lightweight ontology language)
- “A little semantics goes a long way” (Hendler, 1997)
Classes and Instances

- Property `rdf:type` defines the subject of a triple as of type of the object.
- Object of the triple is interpreted as identifier for the class, which contains the resources denoted via subject of the triple.
- Example: “The individual Hegel is of type Person”
  
  `dbpedia:Georg_Wilhelm_Friedrich_Hegel rdf:type foaf:Person`.

- Class membership is not exclusive:
  
  Example:
  
  `dbpedia:Georg_Wilhelm_Friedrich_Hegel rdf:type yago:PoliticalPhilosophers`.

- Instances and classes both use same syntax for URIs, so no syntactical distinction.
Subclasses - Motivation

- Given triple
  - dbpedia:Georg_Wilhelm_Friedrich_Hegel rdf:type yago:PoliticalPhilosophers .
  - and a query for all foaf:Person instances
    - we do not get any results

- We could add the triple
    - but would solve the problem only for one instance
Subclasses

Solution:
- Make one statement which says that every scientist is a person
- Which means every instance of class
  yago:PoliticalPhilosophers is also an instance of class
  foaf:Person

Realised via rdfs:subClass property

Example:
“The class of political philosophers is a subclass of the class of persons”
yago:PoliticalPhilosophers rdfs:subClassOf foaf:Person.
Subclasses

- `rdfs:subClassOf` is reflexive, that is, every class is a subclass of itself
  
  Example:
  `yago:PoliticalPhilosophers rdfs:subClassOf yago:PoliticalPhilosophers .`

- Possible to equate two classes via reciprocal subclass relations:
  
  Example:
  `dbpedia:Person rdfs:subClassOf foaf:Person .
  foaf:Person rdfs:subClassOf dbpedia:Person .`
Class Hierarchies

- Typically, ontologies contain not only single subclass relations, but class hierarchies
  Example:
  yago:PoliticalPhilosophers rdfs:subClassOf yago:Philosophers .
  yago:Philosophers rdfs:subClassOf dbpedia:Person .
  dbpedia:Person rdfs:subClassOf dbpedia:Mammal .

- Transitivity of rdfs:subClassOf is part of the RDFS semantics, which means e.g., the following holds:
  Example:
  dbpedia:Philosophers rdfs:subClassOf dbpedia:Mammal .
Further RDFS Primitives

- Property hierarchies via `rdfs:subPropertyOf`
- Restrictions on properties via `rdfs:domain` and `rdfs:range`
- Lists and collections
- Reification (statements about statements)
- Annotations via `rdfs:label` or `rdfs:comment`
RDFS Summary

- RDFS can be used to describe semantic aspects of specific domains
- On the basis of RDFS it is possible to infer implicit knowledge
- However, the primitives of RDFS have limited expressivity
Web Ontology Language OWL

- Fragment of first-order logics
- Five variants: OWL EL, OWL RL, OWL QL, OWL DL, OWL Full
- OWL DL is decidable and has a corresponding description logics SROIQ (D)
- OWL documents are RDF documents
- Three building blocks are
  - Classes (comparable to classes in RDFS)
  - Individuals (comparable to instances in RDFS)
  - Roles (comparable to properties in RDFS)
- OWL contains primitives to specify elaborate expressions, e.g. two classes are disjoint
- OWL allows for complex reasoning tasks such as consistency check, but may be computationally expensive

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Equivalence

- OWL allows for specification of equivalence; needed in data integration scenarios
- Between individuals: owl:sameAs
  Example:

- Between properties: owl:equivalentProperty
- Between classes: owl:equivalentClass
  Example:
  dbpedia:Person owl:equivalentClass foaf:Person .

- However, equivalences are often implicitly stated in the data
Inverse Functional Properties

- Possible to define “uniquely identifying properties” useful for object consolidation
- E.g. (hypothetical) from
  ex:passportNo rdf:type owl:inverseFunctionalProperty .
  and
dbpedia:Marlene_Dietrich ex:passportNo "12033-89-5" .
- follows:
  dbpedia:Marlene_Dietrich owl:sameAs
  freebase:en.marlene_dietrich .
Further OWL Primitives

- Property characteristics: inverse properties, symmetric properties

- Property cardinality: minimum cardinality, maximum cardinality

- Class restrictions

- Property chains

- ...
LINKED DATA APPLICATION ARCHITECTURES
Data Integration System Architecture

Integration

Wrapper 1
Source 1

Wrapper 2
Source 2

Wrapper n
Source n
Semantic Web Components

User Interface & Applications

Query: SPARQL

Data interchange:
- RDF
- XML
- URI/IRI

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Linked Data: Minimal Components

1. Query
2. Answer

User Interface & Applications

Query: SPARQL

Data interchange: RDF, XML, URI/IRI

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Architecture Styles

Warehousing/Crawl-Index-Serve

Virtual Integration/Distributed Querying

1. Query
2. Answer

0. Crawl-Index

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
Basic Application: Entity Browsing

Warehousing/
Crawl-Index-Serve

Virtual Integration/
Distributed Querying

SWSE, Falcons, Sindice, Watson,
FactForge…

Tabulator, Disco, Zitgist…

Marko Grobelnik, Andreas Harth, Dumitru Roman, Big Linked Data
SUMMARY
Summary

- The Linked Data Web is a large, decentralised, complex system built on simple principles
  - identify resource via HTTP URIs
  - provide RDF that links to other URIs upon lookup
- Current trend around Linked Data allows for a re-think of components in Semantic Web Layer Cake
- Data publishers and consumers coordinate little
- Web of Data grows rapidly and covers a large variety of domains
- Algorithms operating over a common access protocol and data model
- Ontology languages provide integration and mapping between disparate sources
- First commercial applications emerging
Attribution

- Slides from my SWT-2 lectures and WWW 2010 SILD tutorial
- Slides about RDFS and OWL adapted from SWT-1 lecture (Rudolph, Kroetzsch, Harth)
- Images of Berlin, Hegel and Dietrich via Wikipedia