Knowledge Representation for the Semantic Web

Winter Quarter 2012

Slides 12 – 03/01+08/2012

Pascal Hitzler
Kno.e.sis Center
Wright State University, Dayton, OH
http://www.knoesis.org/pascal/

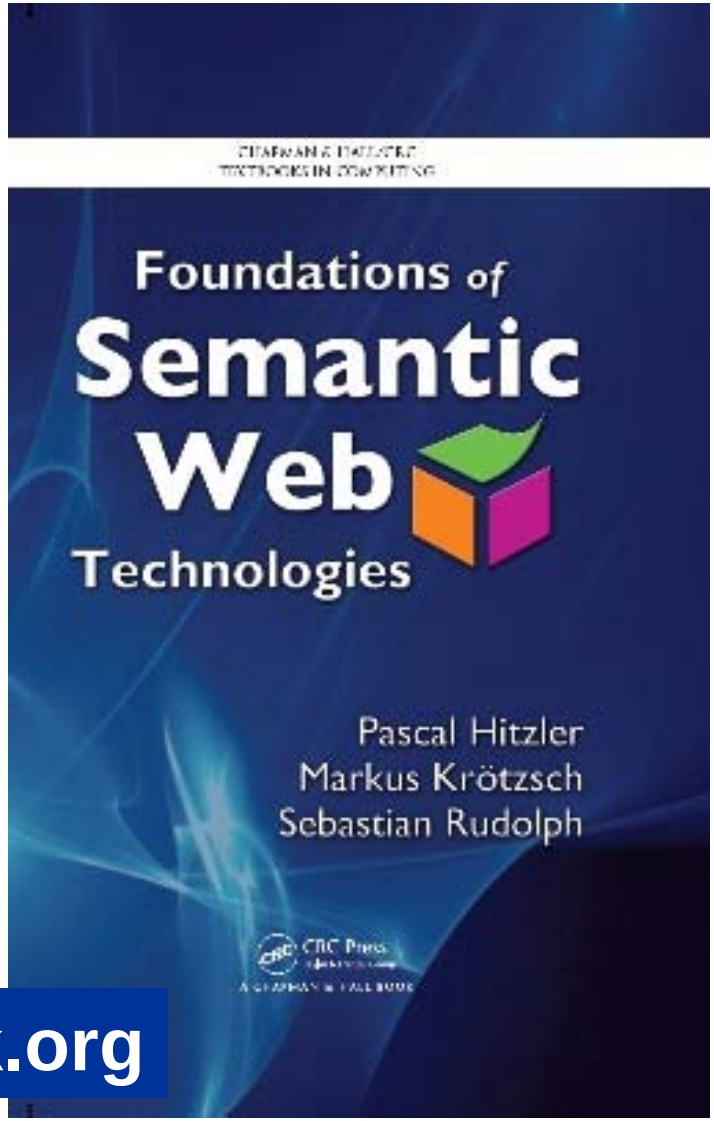
Textbook (required)

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

Foundations of Semantic Web Technologies

Chapman & Hall/CRC, 2010

Choice Magazine Outstanding Academic Title 2010 (one out of seven in Information & Computer Science)



http://www.semantic-web-book.org

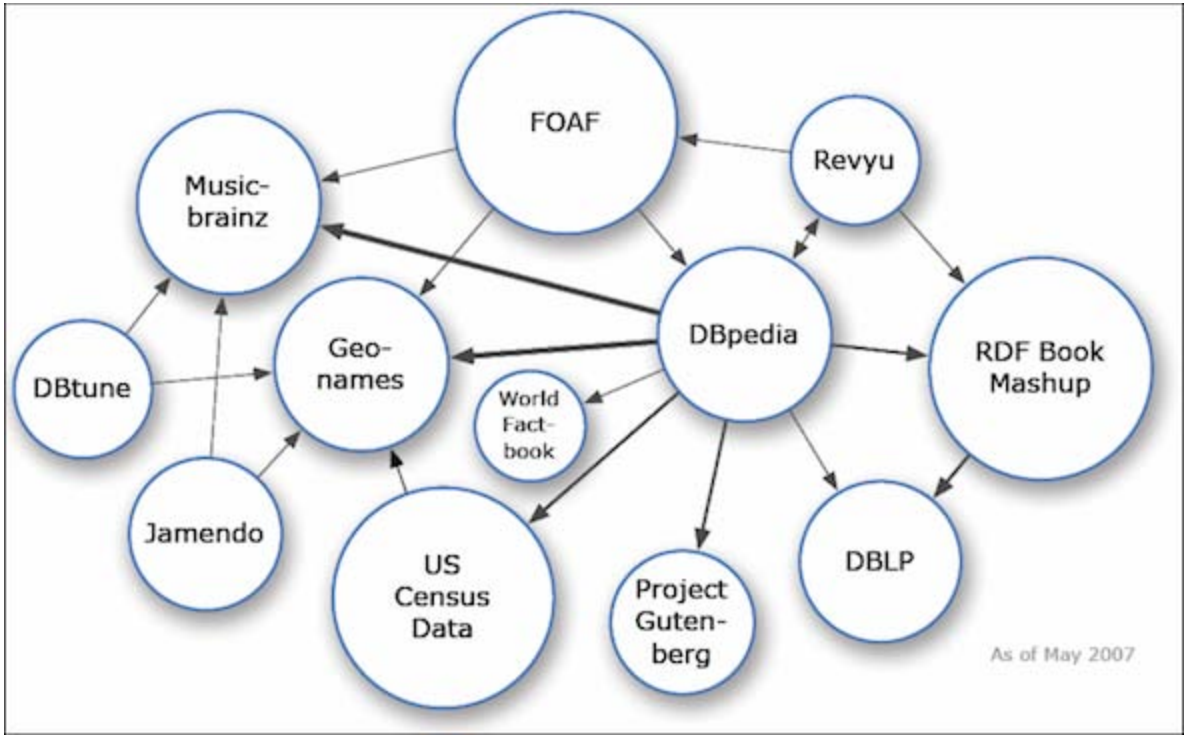
Contents

Tim Berners-Lee 2006

- **from http://www.w3.org/DesignIssues/LinkedData.html**

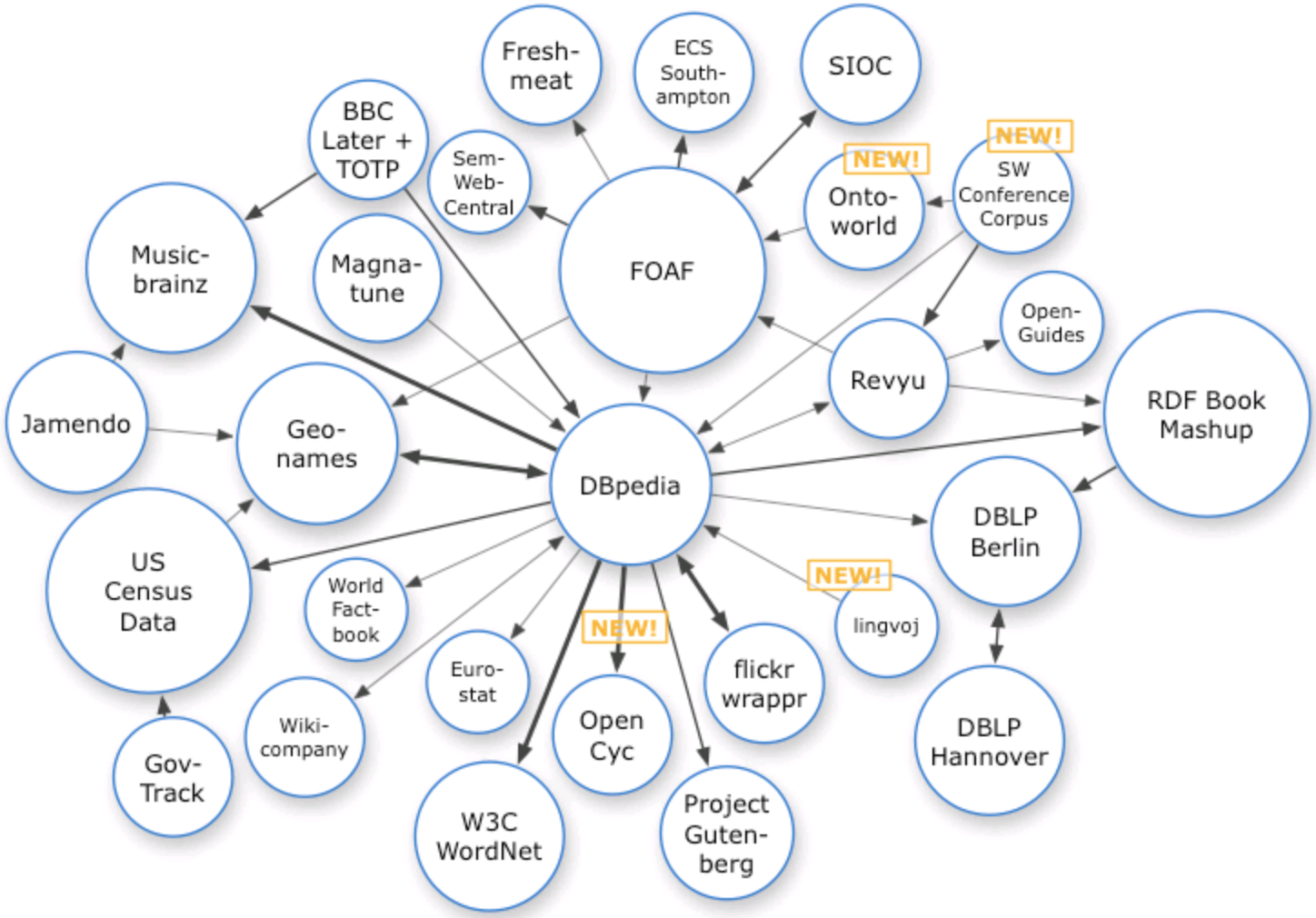
1. **Use URIs as names for things**
2. **Use HTTP URIs so that people can look up those names.**
3. **When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)**
4. **Include links to other URIs. so that they can discover more things.**

Linked Open Data 2007 (May)

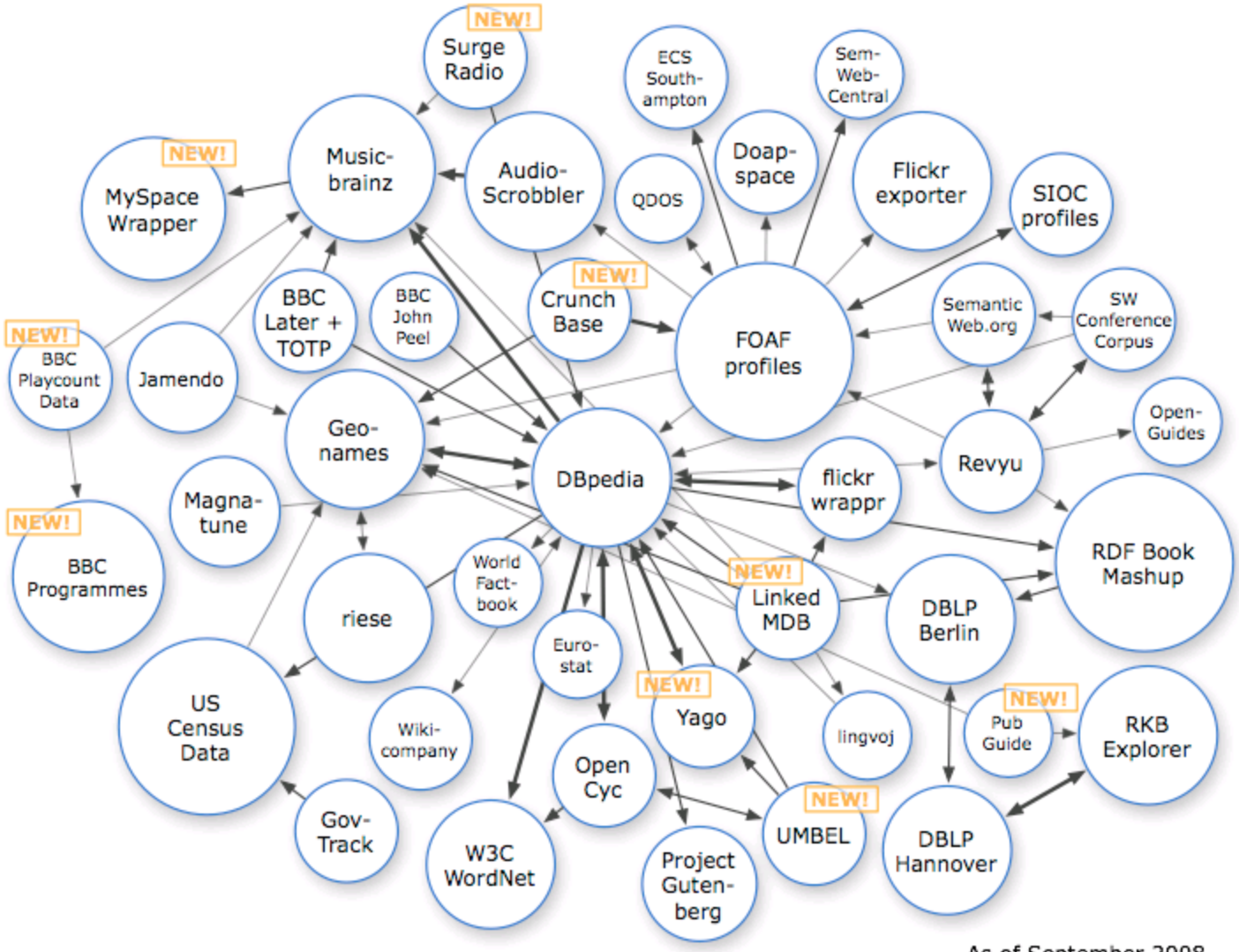


Linking Open Data cloud diagram, this and subsequent pages, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

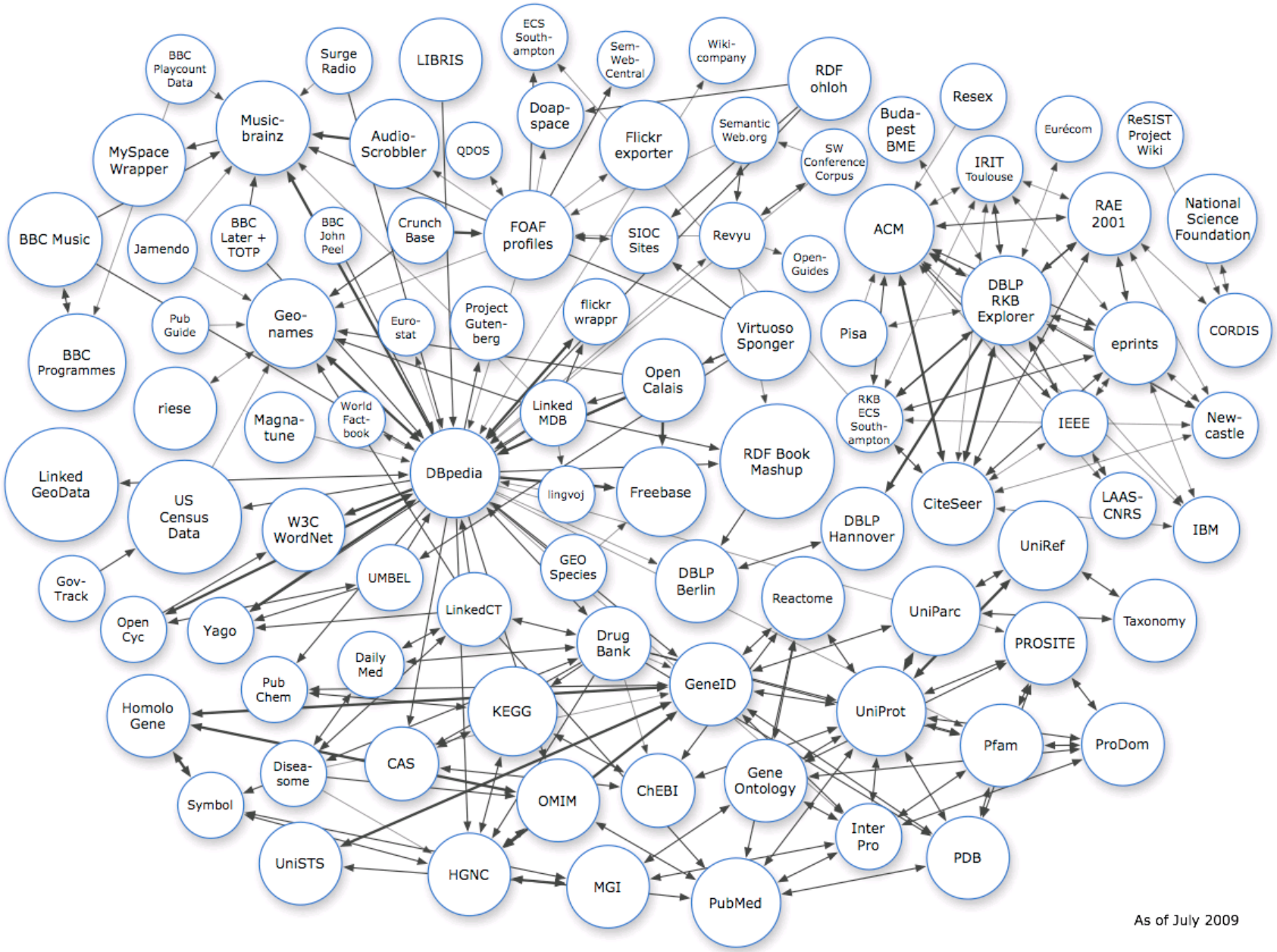
Linked Open Data 2007 (Oct)



Linked Open Data 2008



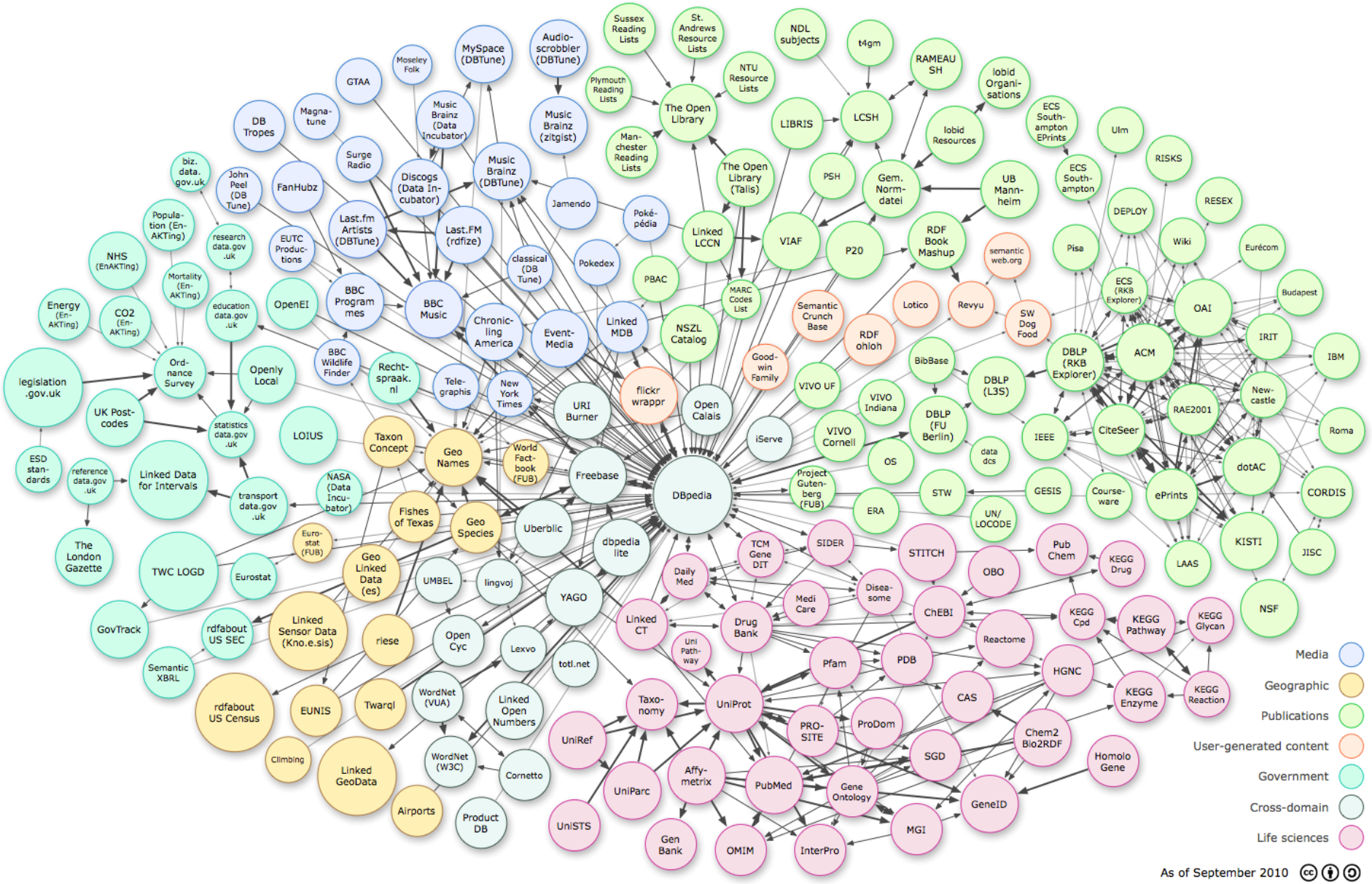
Linked Open Data 2009



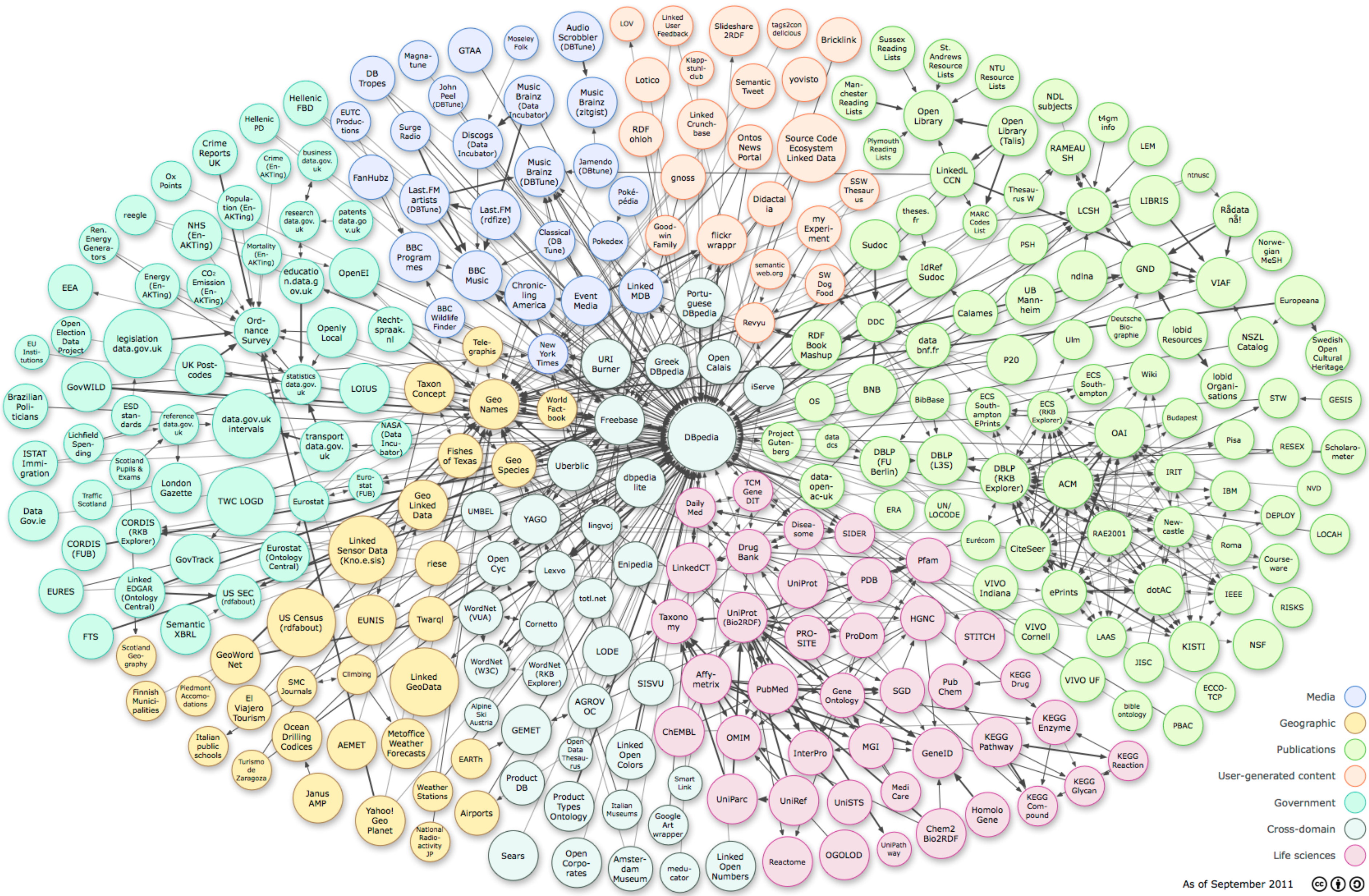
As of July 2009

Linked Open Data 2010

kno.e.sis



Linked Open Data 2011



Linked Open Data

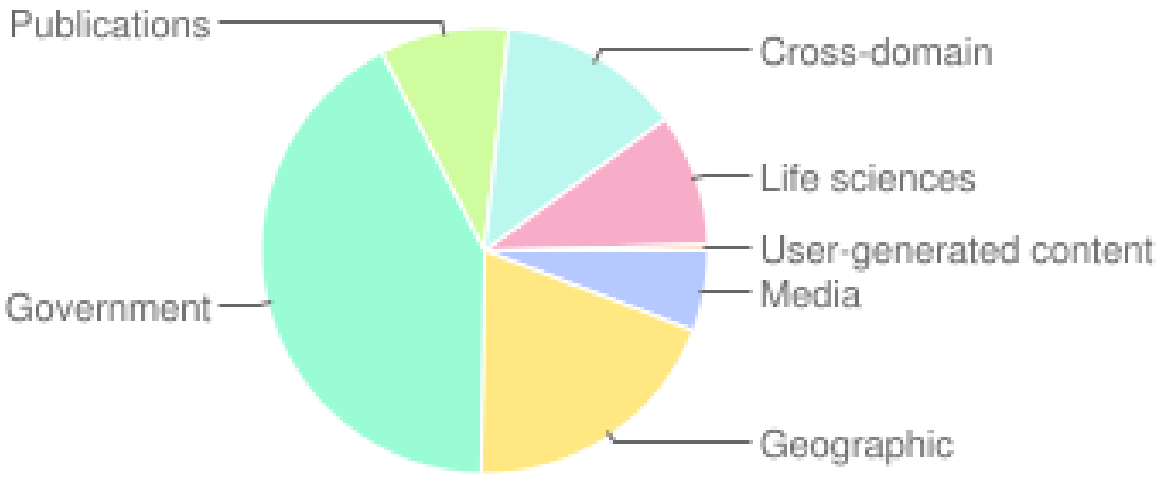
Number of Datasets

2011-09-19	**295**
2010-09-22	**203**
2009-07-14	**95**
2008-09-18	**45**
2007-10-08	**25**
2007-05-01	**12**

Number of triples (Sept 2011)

31,634,213,770

with 503,998,829 out-links



From http://www4.wiwiss.fu-berlin.de/lodcloud/state/

Example: GeoNames

Populated Place Features (city, village,...)			
2,518,403	P.PPL	populated place	a city, town, village, or other agglomeration of buildings where people live and work
48,483	P.PPLX	section of populated place	
39,336	P.PPLL	populated locality	an area similar to a locality but with a small group of dwellings or other buildings
13,306	P.PPLQ	abandoned populated place	
2,684	P.PPLA4	seat of a fourth-order administrative division	
2,028	P.PPLA	seat of a first-order administrative division	seat of a first-order administrative division (PPLC takes precedence over PPLA)
1,847	P.PPLW	destroyed populated place	a village, town or city destroyed by a natural disaster, or by war
1,006	P.PPLF	farm village	a populated place where the population is largely engaged in agricultural activities
930	P.PPLA3	seat of a third-order administrative division	
695	P.PPLA2	seat of a second-order administrative division	
253	P.PPLS	populated places	cities, towns, villages, or other agglomerations of buildings where people live and work
249	P.STLMT	israeli settlement	
235	P.PPLC	capital of a political entity	
57	P.		
29	P.PPLR	religious populated place	a populated place whose population is largely engaged in religious occupations
6	P.PPLG	seat of government of a political entity	
2,629,547	Total for P		

rdfs:subClassOf?

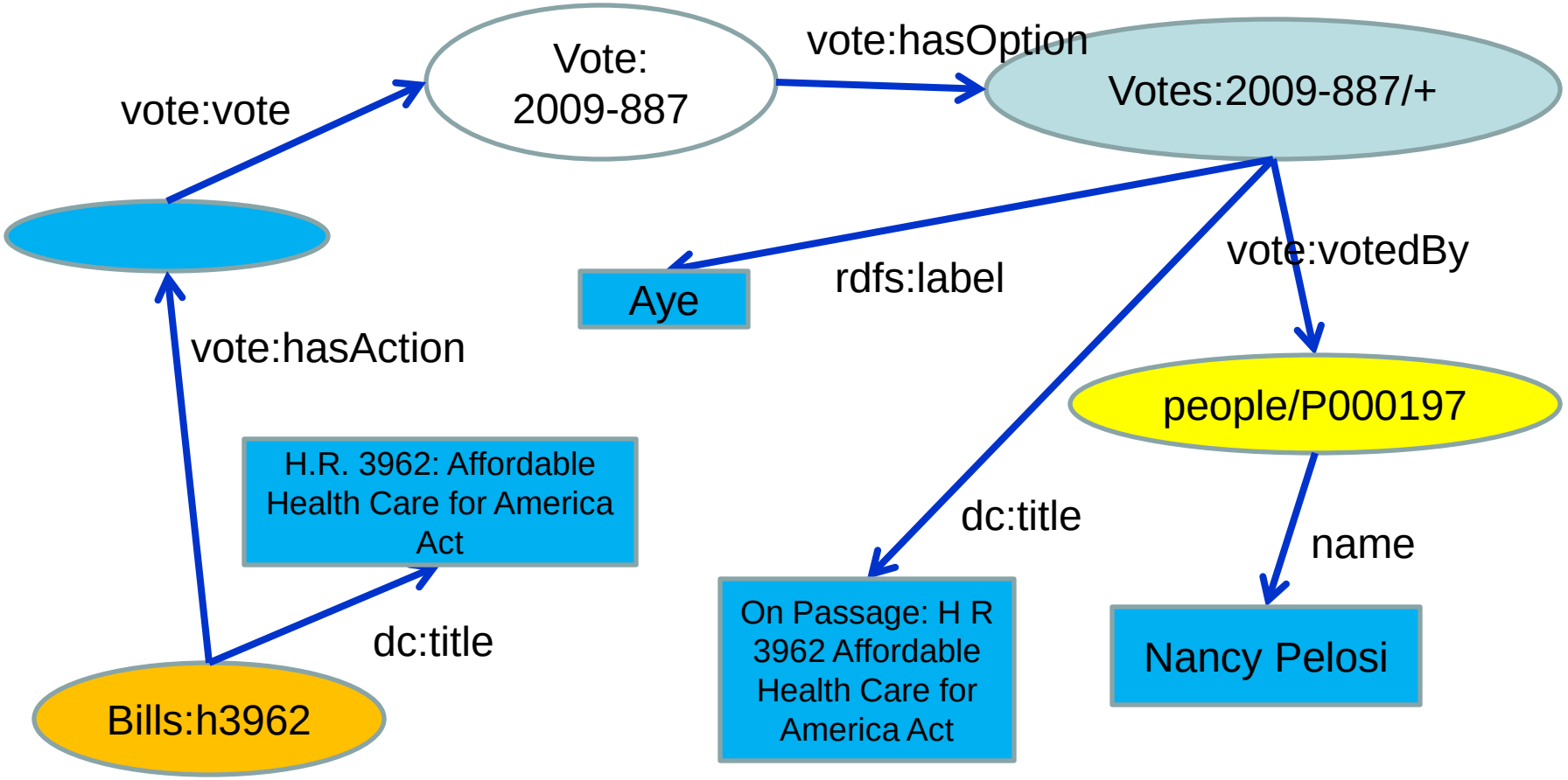
Contents

From agoogleaday.com

- What tribe has lived since 1300 AD near the canyon you'd explore from Bright Angel Trail?
- The highway that runs through Rachel, Nevada draws enthusiasts who probably enjoy what movie genre?
- If you key in international dialing code 40, how would you say "good morning" in the language of the country you're calling?
- What word will you use for "taxi" if the airport code of your destination is OSL?
- What single state is home to all of the following U.S. cities: Madrid, Toronto, Cincinnati, Denver, Hartford, and Norway?

Example: GovTrack

“Nancy Pelosi voted in favor of the Health Care Bill.”



Example: GovTrack

```
bills/h3962       dc:title          "H.R. 3962: ..." ;
                  usbill:hasAction  _:bnode0 .
_:bnode0          usbill:vote       votes/2009-887 .
votes/2009-887    vote:hasOption    votes/2009-887/+ .
                  dc:title       "On Passage: H.R. 3962 ..." ;
votes/2009-887/+  rdfs:label        "Aye" ;
                  vote:votedBy      people/P000197 .
people/P000197    usgovt:name       "Nancy Pelosi" .
```

Example querying LoD

"Identify congress members, who have voted "No" on pro environmental legislation in the past four years, with high-pollution industry in their congressional districts."

In principle, all the knowledge is there:

- **GovTrack**
- **GeoNames**
- **DBPedia**
- **US Census**

But even with LoD we cannot answer this query.

Example querying LoD

"Identify congress members, who have voted "No" on pro environmental legislation in the past four years, with high-pollution industry in their congressional districts."

Some missing puzzle pieces:

- Where is the data?
  - GovTrack
    GeoNames
    US Census
    requires intimate knowledge of the LoD data sets

Example querying LoD

"Identify congress members, who have voted "No" on pro environmental legislation in the past four years, with high-pollution industry in their congressional districts."

Some missing puzzle pieces:

- Where is the data?
  (smart federation needed)
- Missing background (schema) knowledge.
  (enhancements of the LoD cloud)
- Crucial info still hidden in texts.
  (ontology learning from texts)
- Added reasoning capabilities (e.g., spatial).
  (new ontology language features)

Don’t get me wrong

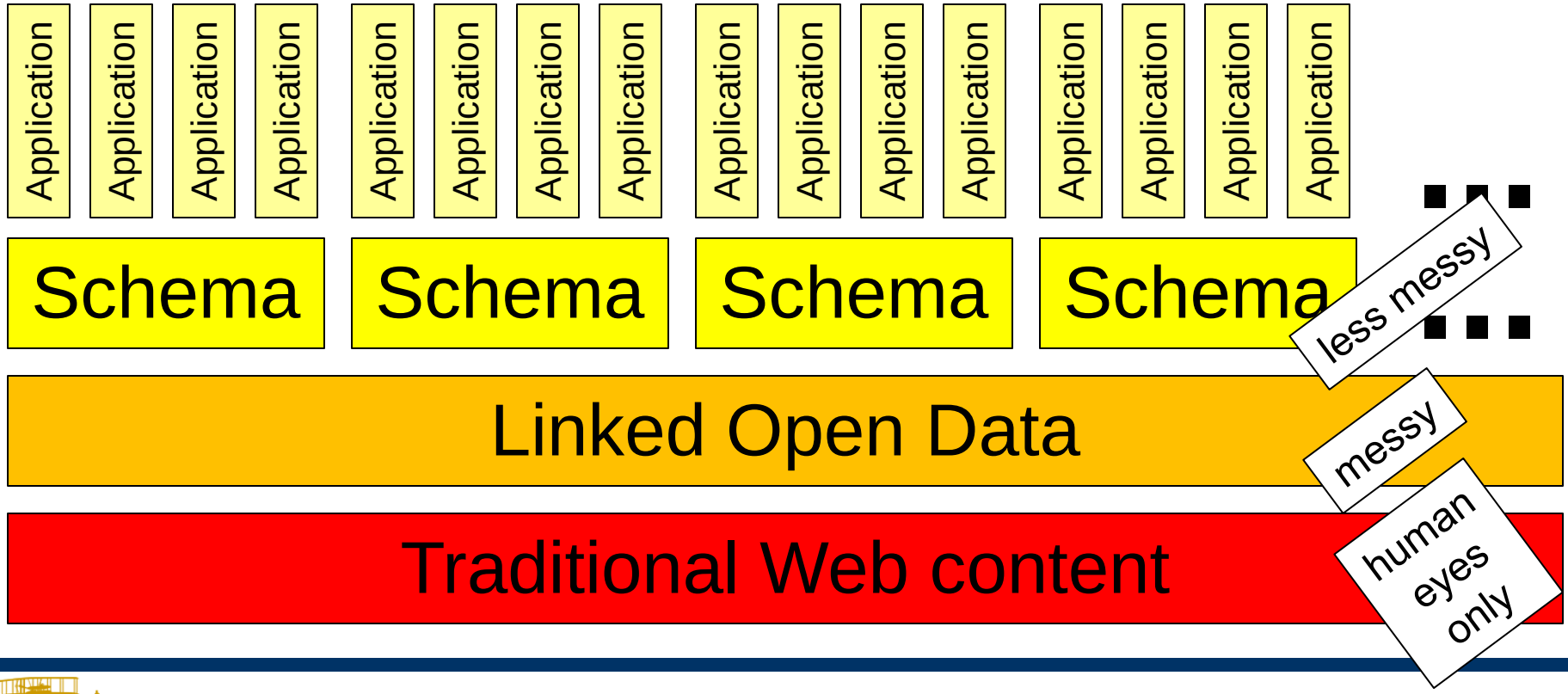
Linked Open Data is great, useful, cool, and a very important step.

But we need to make use of the added value of formal semantics in order to advance towards the Semantic Web vision!

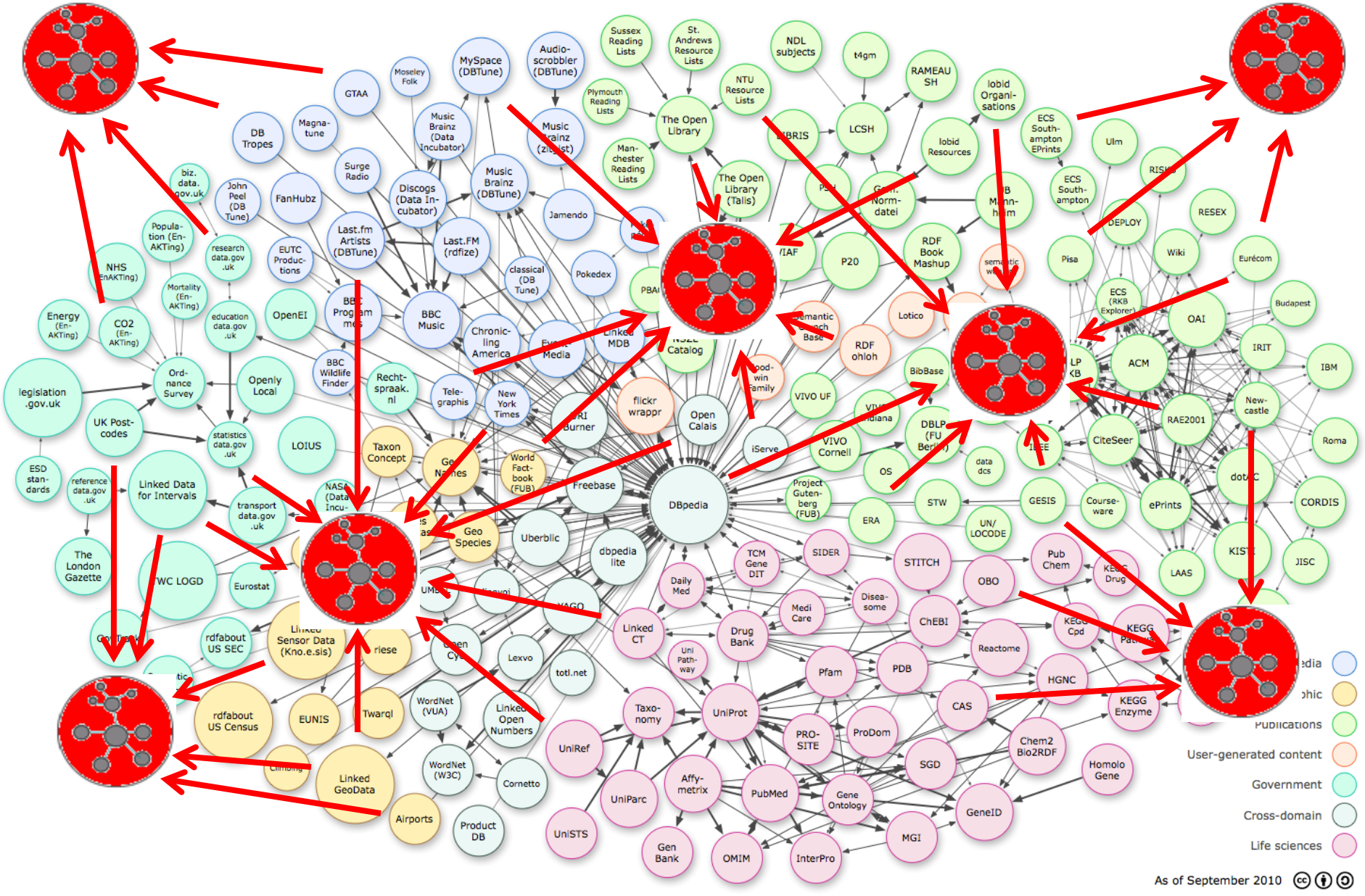
The Semantic Data Web Layer Cake

To leverage LoD, we require schema knowledge

- **application-type driven (reusable for same kind of application)**
- **less messy than LoD (as required by application)**
- **overarching several LoD datasets (as required by application)**



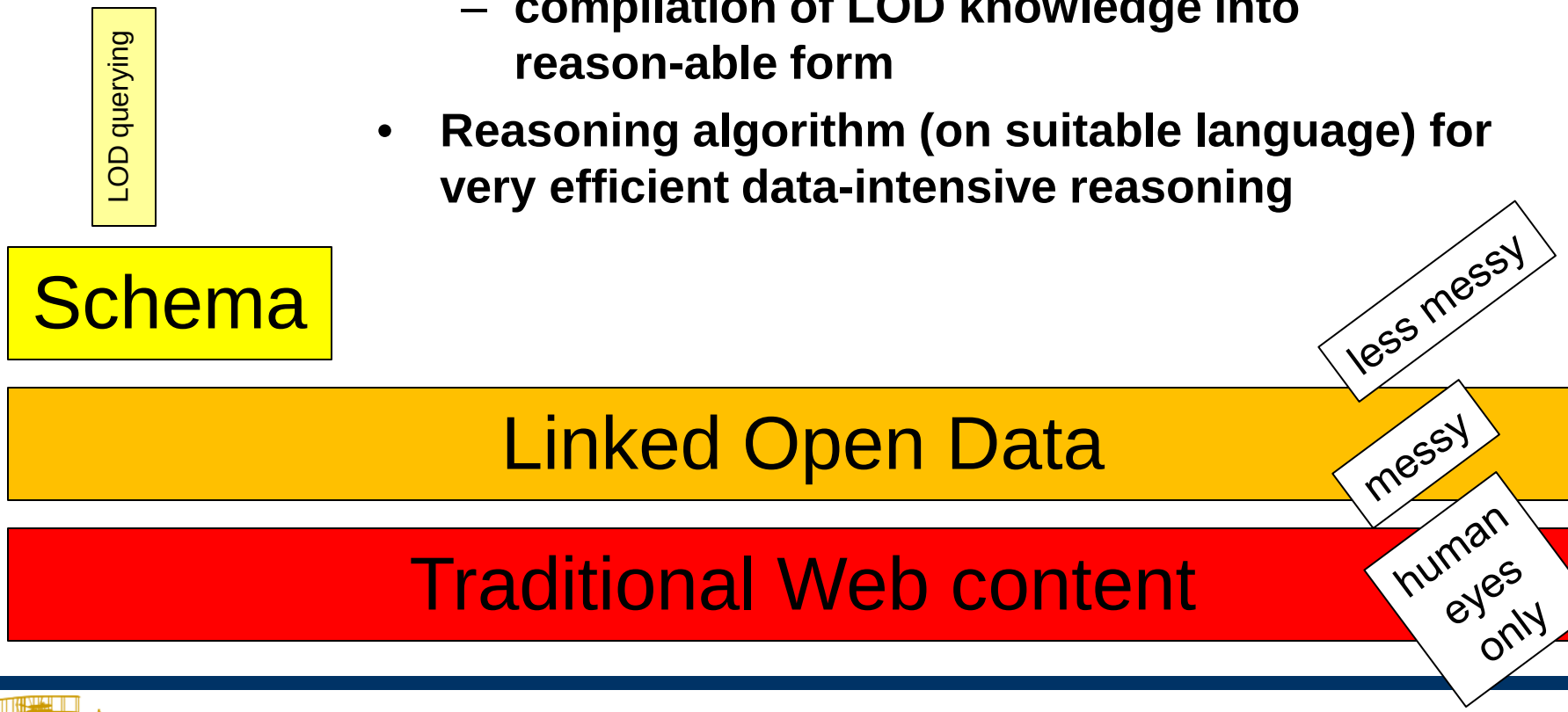
Schema on top of the LoD cloud



Idea – Querying Linked Open Data

Work in progress.

- **Schema creation for**
  - **query federation**
  - **utilizing background knowledge**
  - **compilation of LOD knowledge into reason-able form**
- **Reasoning algorithm (on suitable language) for very efficient data-intensive reasoning**



Contents

BLOOMS

Table 4. Results of various systems for LOD Schema Alignment. Legends: Prec=Precision, Rec=Recall, M=Music Ontology, B=BBC Program Ontology, F=FOAF Ontology, D=DBpedia Ontology, G=Geonames Ontology, S=SIOC Ontology, W=Semantic Web Conference Ontology, A=AKT Portal Ontology, err=System Error, NA=Not Available

Linked Open Data Schema Ontology Alignment												
	Alignment API		OMViaUO		RiMoM		S-Match		AROMA		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
M,B	0.4	0	1	0	err	err	0.04	0.28	0	0	0.63	0.78
M,D	0	0	0	0	err	err	0.08	0.30	0.45	0.01	0.39	0.62
F,D	0	0	0	0	err	err	0.11	0.40	0.33	0.04	0.67	0.73
G,D	0	0	0	0	err	err	0.23	1	0	0	0	0
S,F	0	0	0	0	0.3	0.2	0.52	0.11	0.30	0.20	0.55	0.64
W,A	0.12	0.05	0.16	0.03	err	err	0.06	0.4	0.38	0.03	0.42	0.59
W,D	0	0	0	0	err	err	0.15	0.50	0.27	0.01	0.70	0.40
Avg.	0.07	0.01	0.17	0	NA	NA	0.17	0.43	0.25	0.04	0.48	0.54

Jain, Hitzler et al, ISWC2010

BLOOMS

Table 1. Results on the oriented matching track. Results for RiMOM and AROMA have been taken from the OAEI 2009 website. Legends: Prec=Precision, A-API=Alignment API, OMV=OMViaUO, NaN=division by zero, likely due to empty alignment.

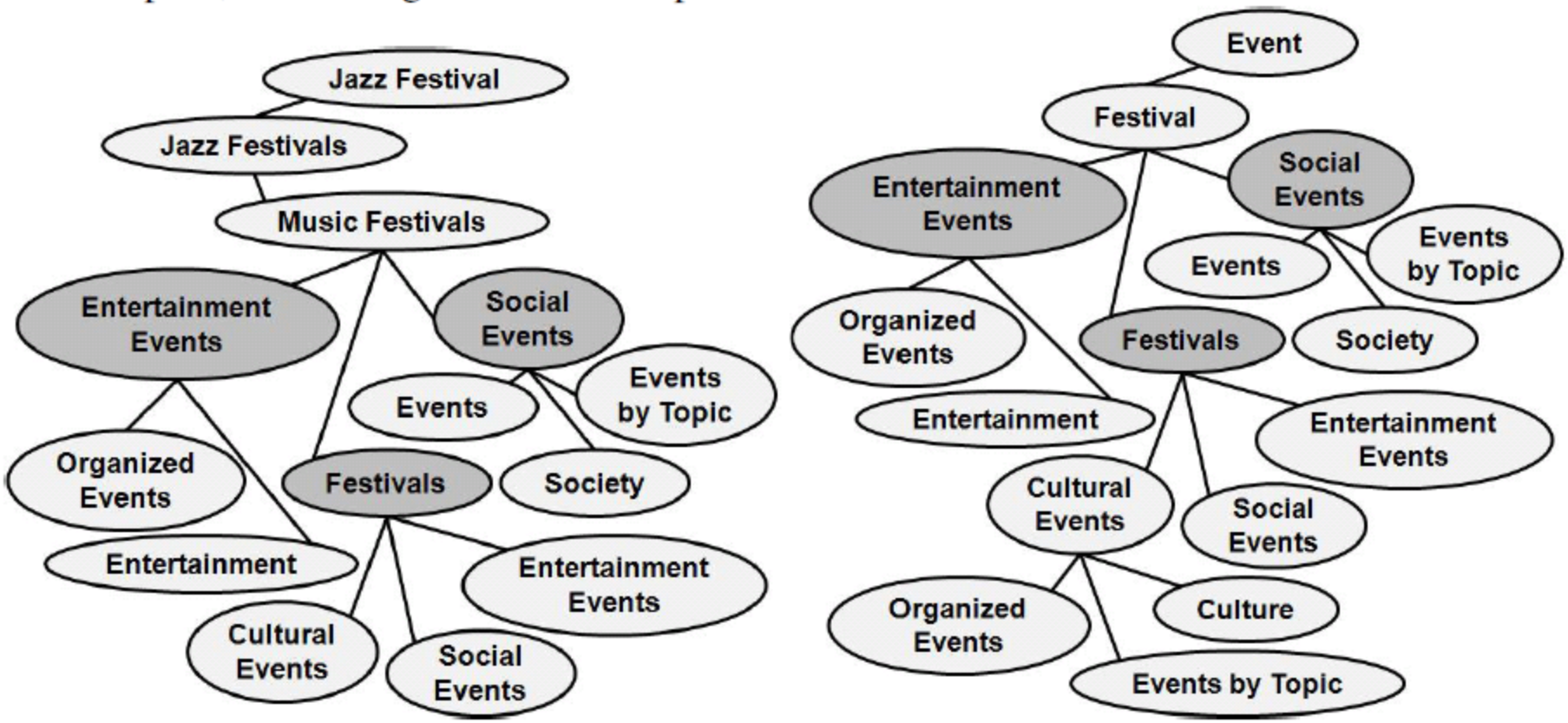
Ontology Alignment Initiative—Oriented Matching Track												
	A-API		OMV		S-Match		AROMA		RiMoM		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
1XX	0	0	0.02	0.06	0.01	0.71	NaN	0	1	1	1	1
2XX	0	0	0.01	0.03	0.05	0.30	0.84	0.08	0.67	0.85	0.52	0.51
3XX	0.01	0.03	0.02	0.047	0.01	0.14	0.72	0.11	0.59	0.81	1	0.84
Avg.	0.00	0.01	0.02	0.04	0.03	0.38	0.63	0.07	0.75	0.88	0.84	0.78

BLOOMS

1. **Pre-processing of the input ontologies** in order to (i) remove property restrictions, individuals, and properties, and to (ii) tokenize composite class names to obtain a list of all simple words contained within them, with stop words removed.
2. **Construction of the BLOOMS forest** T_C for each class name C, using information from Wikipedia.
3. **Comparison of constructed BLOOMS forests**, which yields decisions which class names are to be aligned.
4. **Post-processing** of the results with the help of the Alignment API and a reasoner.

BLOOMS trees

Fig. 1. BLOOMS trees for Jazz Festival with sense Jazz Festival and for Event with sense Event. To save space, some categories are not expanded to level 4.



BLOOMS

1. **Pre-processing of the input ontologies** in order to (i) remove property restrictions, individuals, and properties, and to (ii) tokenize composite class names to obtain a list of all simple words contained within them, with stop words removed.
2. **Construction of the BLOOMS forest** T_C for each class name C, using information from Wikipedia.
3. **Comparison of constructed BLOOMS forests**, which yields decisions which class names are to be aligned.
4. **Post-processing** of the results with the help of the Alignment API and a reasoner.

BLOOMS decisions

1. Remove from T_s all nodes for which there is a parent node which occurs in T_t. All leaves of the resulting tree T'_s are either of level 4 or occur in T_t. Note that due to the way BLOOMS trees are constructed, we removed only nodes from T_s which actually occur in T_t—we remove them because they do not give us any essential additional information for comparing T_s with T_t.
2. $o(T_s, T_t) = \frac{n}{k-1}$, where n is the number of nodes in T'_s which occur also in T_t, and k is the total number of nodes in T'_s (we do not count the root).

The decision on an alignment is then made as follows.

- If, for any choice of $T_s \in T_C$ and $T_t \in T_D$, we have that $T_s = T_t$, then we set C owl:equivalentClass D.
- If $\min\{o(T_s, T_t), o(T_t, T_s)\} \geq x$ for any choice of $T_s \in T_C$ and $T_t \in T_D$, and for some pre-defined threshold x,[8] then set C rdfs:subClassOf D if $o(T_s, T_t) \leq o(T_t, T_s)$, and set D rdfs:subClassOf C if $o(T_s, T_t) \geq o(T_t, T_s)$.

BLOOMS Test Data

Table 3. LOD datasets=LOD datasets utilizing this schema, D=taxonomic depth, # C=number of classes, Linked datasets=LOD datasets they are linked to at the instance level

Schema	LOD datasets	D	# C	Linked datasets
DBpedia[20]	DBpedia	4	204	Geonames, US Census, Freebase
Geonames[21]	Geonames, Geospecies	2	11	DBpedia, Jamendo, FOAF Profiles
Music Ontology[22]	Jamendo, Music Brainz, DBTunes	4	136	GovTrack, DBpedia, Geonames
BBC Program[23]	BBC Programs, BBC Music	4	100	BBC Music, BBC Playcount Data
FOAF Profiles[24]	FOAF, Music Brainz	3	16	Crunch Base, QDOS, SIOC Sites
SIOC[25]	DBpedia, LinkedMDB	2	14	Virtuoso Sponger, FOAF Profiles, SemanticWeb.org
AKT Reference Ontology[26]	ACM, DBLP	5	17	Pisa, IEEE, eprints
Semantic Web Conference Ontology[27]	SW Conference Corpus	5	177	SemanticWeb.org, Revyu

Further enhancements

- **e.g. with partonomies [Jain et al, submitted]**
- **So far we have a partonomies crator, called PLATO.**
- **Evaluations on next slides.**

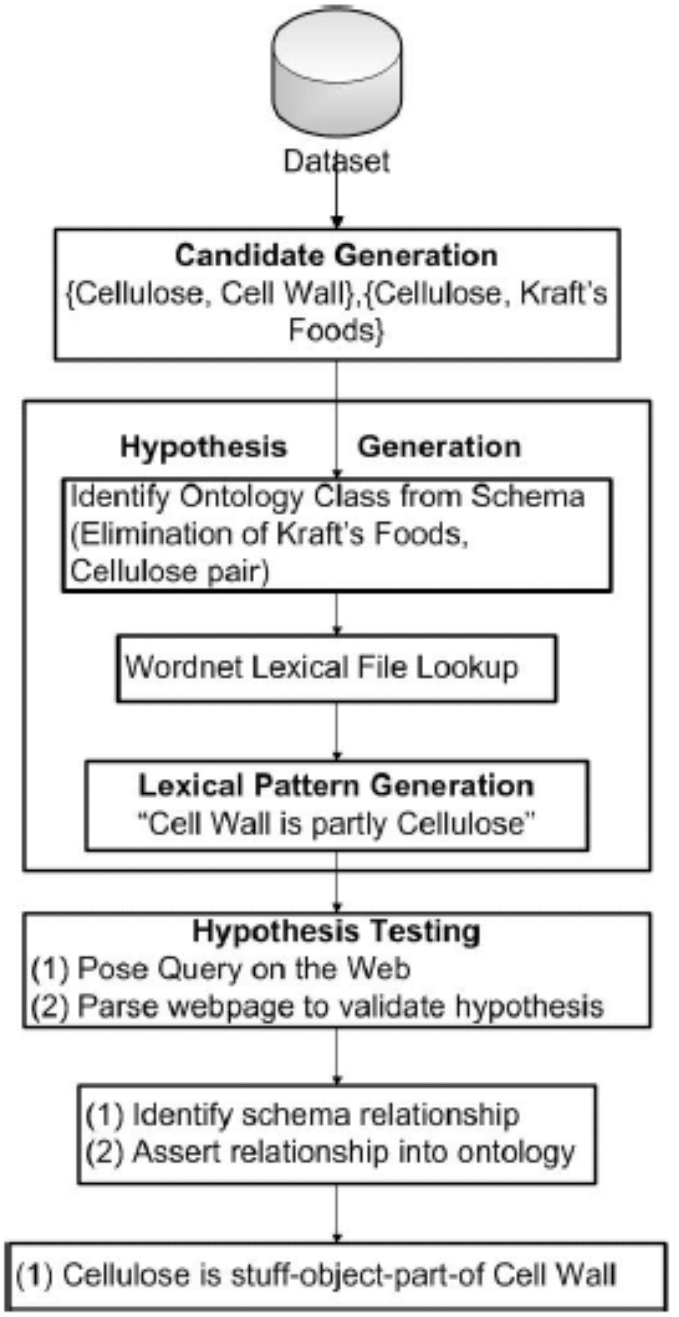


Figure 1: PLATO system flow chart

PLATO partonomy detection system

- **Intra-dataset partonomy detection**

Relation Type	Distinct Entity Pairs	Correctly Found	Precision
Stuff-Object-Part-Of	4178	3427	0.82
Component-Integral-Part-Of	3126	27931	0.89
Feature-Activity-Part-Of	1287	464	0.85
Member-Collection-Part-Of	1912	803	0.85
Portion-Mass-Part-Of	0	0	NA
Place Area-Part-Of	3350	1248	0.48
Total	13853	10557	0.76

Table 2: Precision of the six different relation types between DBpedia entities

PLATO partonomy detection system

- **Intra-dataset partonomy detection**
  - **Between Freebase dishes and DBPedia ingredients**
  - **Between Freebase human anatomy parts and DBPedia organs**

Task	Recall	Precision
Dish-Ingredient Task	0.72	0.53
Anatomy-Organ Task	N/A	0.86

PLATO partonomy detection system

- **Discovery of schema-level links**

In addition to adding properties at the instance-level (i.e. between entities), PLATO also enriches the schema by generalizing from the instance level assertions. To explain this step, let C and D be two classes about which we want to find out whether they should be related on the schema level by one of the partonomic relationships R. From the process just described, we obtain a set $M_{R,C,D}$ of instance level assertions of the form $R(a, b)$, where $a \in C$ and $b \in D$.[13] We now add schema level axioms according to the following rules: (1) If, for all $a \in C$, there is a $b \in D$ with $R(a, b) \in M_{R,C,D}$, then add the axiom $C \sqsubseteq \exists R.D$, which can be expressed in OWL/RDF serialization using the *owl:someValuesFrom* property restriction. (2) If, for all $b \in D$, there is a $a \in C$ with $R(a, b) \in M_{R,C,D}$, then add the axiom $D \sqsubseteq \exists R^{-}.C$, were R^{-} indicates the inverse (using *owl:inverseOf*) property of R. While this approach seems to be rather crude compared to schema learning methods based on inductive paradigms,[14] it already achieves good results, as can be seen from our evaluation in Section 4.3.

Schema-level links evaluation

Total # of Class Pairs	Correctly Identified	Precision
93	81	0.87

Table 4: Precision as measured on Schema Level Links Between DBpedia entities

```
dbpedia-owl:Band rdfs:subClassOf  [
   rdf:type              owl:Restriction ;
   owl:onProperty        :hasMember ;
   owl:someValuesFrom    dbpedia-owl:Artist
 ] .
```

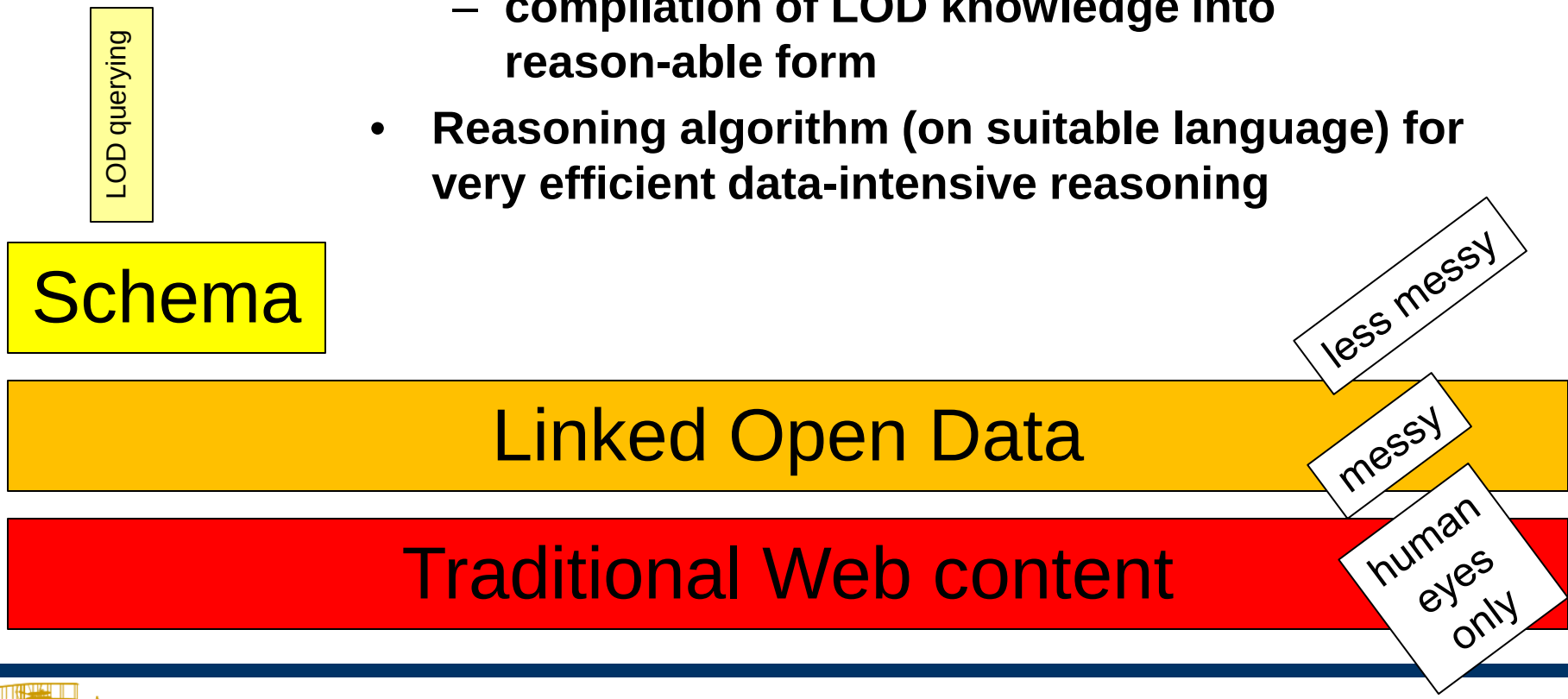
Contents

1. Linked Data
2. Linked Data Querying: The problem
3. Linked Data Alignment: BLOOMS and PLATO
4. Linked Data Querying with ALOQUS
5. References

Idea – Querying Linked Open Data

Work in progress.

- **Schema creation for**
  - **query federation**
  - **utilizing background knowledge**
  - **compilation of LOD knowledge into reason-able form**
- **Reasoning algorithm (on suitable language) for very efficient data-intensive reasoning**



Linked Data Query

```
#find the landlocked countries and their population
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>

SELECT ?countryName ?population WHERE
{
?country rdf:type     type:LandlockedCountries ;
              rdfs:label  ?countryName ;
              prop:populationEstimate ?population .
   FILTER ( lang(?countryName) = "en" )
}
```

countryName	population
Burkina Faso	15746232
Laos	6800000
Liechtenstein	35789
Niger	15306252
Nepal	29331000
Zambia	12935000
Malawi	14901000
Rwanda	11370425
Uganda	32369558
Turkmenistan	5110000
Serbia	7306677
Serbia	9496000
Bhutan	691141
Swaziland	1185000

ALOQUS Illustration

"Identify films, the nations where they were shot and the population of these countries"

```
SELECT ?film ?nation ?pop

WHERE {
?film     protonu:ofCountry          ?nation.
?film     rdf:type                   protonu:Movie.
?film     rdfs:label                 ?film_name.
?nation protont:populationCount      ?pop.
}
```

ALOQUS Illustration – Blooms Alignment

protonu:ofCountry	maps to	lmdb:country
protonu:Movie	maps to	lmdb:film
protont:populationCount	maps to	dbprop:populationCount

Alignment confidence > 0.9

ALOQUS Illustration – Sub Queries

```
(a) SELECT ?film ?nation ?pop
WHERE {
?film lmdb:country ?nation.
?film rdf:type lmdb:film.
?film rdfs:label ?film_name.
}

(b) SELECT ?nation ?pop
WHERE {
?nation dbprop:populationCensus ?pop.
```

ALOQUS Illustration

How to preserve the upper level ontology terms?

Use CONSTRUCT instead of SELECT

ALOQUS Illustration – Construct Query

```
CONSTRUCT {
    ?film protonu:ofCountry ?nation.
    ?film rdf:type protonu:Movie.
    ?film rdfs:label ?film_name.
}
WHERE {
    ?film lmdb:country ?nation.
    ?film rdf:type lmdb:film.
    ?film rdfs:label ?film_name.
}
```

ALOQUS Illustration – Query Results

(a)

lmdb-film:11446	protonu:ofCountry	lmdb-country:IN.
lmdb-film:11446	rdf:type	protonu:Movie.
lmdb-film:11446	rdfs:label	"Run".
lmdb-film:17091	protonu:ofCountry	lmdb-country:LK.
lmdb-film:17091	rdf:type	protonu:Movie.
lmdb-film:17091	rdfs:label	"Getawarayo".
lmdb-film:16973	protonu:ofCountry	lmdb-country:IN.
lmdb-film:16973	rdf:type	protonu:Movie.
lmdb-film:16973	rdfs:label	"Kabeela".

(b)

dbpedia:Sri_Lanka	protont:PopulationCount	21324791.
dbpedia:India	protont:PopulationCount	1210193422.

ALOQUS Illustration – Entity Co-reference

- **Detect similar entities**
- **lmdb:IN is equivalent to :**

  http://data.linkedmdb.org/resource/country/IN
  http://sws.geonames.org/1269750/
  http://rdf.freebase.com/ns/m.03rk0
  http://dbpedia.org/resource/India
  http://data.nytimes.com/india_geo
  http://dbtune.org/musicbrainz/resource/country/IN
  http://umbel.org/umbel/ne/wikipedia/India
  http://www.ontologyportal.org/SUMO.owl#India
  http://www4.wiwiss.fu-berlin.de/factbook/resource/India

- **Map to a proxy identifier aloqus:9bc35a1**

ALOQUS Illustration – Result I

- **With proxy identifiers**

film	name	nation	population
lmdb-film:17091	"Getawarayo"	aloqus:2908ba82	21324791
lmdb-film:16973	"Kabeela"	aloqus:9bc35ca1	1210193422
lmdb-film:11446	"Run"	aloqus:9bc35ca1	1210193422

ALOQUS Illustration – Result II

- **Without proxy identifiers**

film	name	nation	population
lmdb-film:17091	"Getawarayo"	lmdb-country:LK	21324791
lmdb-film:16973	"Kabeela"	lmdb-country:IN	1210193422
lmdb-film:11446	"Run"	lmdb-country:IN	1210193422
lmdb-film:11446	"Run"	nytimes:india_geo	1210193422

Comparison with other systems

Features	ALOQUS	DARQ	SQUIN
Approach	Uses upper level ontology(PROTON) or any other ontology as primary ontology for query serialization and execution.	Requires formal description of datasets in the form of Service Description.	Requires an initial URI to execute queries.
Query Creation	Creates query corresponding to every mapping for a concept.	Creates queries only corresponding to the concepts mentioned in the query.	Creates queries only corresponding to the concepts mentioned in the query.
Failsafe	Executes all sub-queries for multiple mappings. Hence retrieves at least partial answers if a specific endpoint doesn't work.	X	X
Detect Entity co-references	Crawls and also consumes sameAs.org webservices.	X	X
Result Processing	Query answers, retrieved from different datasets are merged and presented to user.	Retrieves answers from multiple dataset based on service description.	Retrieves answers from multiple dataset through link traversal.
Write queries using ontology not present in LOD	Yes	X	X
Support for open-ended queries like ?s ?p ?o	Yes	X	X
Result Storage for later Retrieval	Yes	X	X
DESCRIBE Query Form	Yes	N/A	Yes

Contents

1. Linked Data
2. Linked Data Querying: The problem
3. Linked Data Alignment: BLOOMS and PLATO
4. Linked Data Querying with ALOQUS
5. References

References

- http://linkeddata.org
- Tim Berners-Lee: http://www.w3.org/DesignIssues/LinkedData.html
- Christian Bizer, Tom Heath, Tim Berners-Lee: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. 5(3): 1-22 (2009)
- Pascal Hitzler, Frank van Harmelen, A reasonable Semantic Web. Semantic Web 1(1-2), 39-44, 2010.
- Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, Amit P. Sheth, Linked Data is Merely More Data. In: Dan Brickley, Vinay K. Chaudhri, Harry Halpin, Deborah McGuinness: Linked Data Meets Artificial Intelligence. Technical Report SS-10-07, AAAI Press, Menlo Park, California, 2010, pp. 82-86. ISBN 978-1-57735-461-1. Proceedings of LinkedAI at the AAAI Spring Symposium, March 2010.

References

- Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, Peter Z. Yeh, Ontology Alignment for Linked Open Data. In P. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Pan, I. Horrocks, B. Glimm (eds.), The Semantic Web - ISWC 2010. 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I. Lecture Notes in Computer Science Vol. 6496. Springer, Berlin, 2010, pp. 402-417.
- Amit Krishna Joshi, Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, Amit P. Sheth, Mariana Damova, Alignment-based Querying of Linked Open Data. Submitted.
- Prateek Jain, Pascal Hitzler, Kunal Verma, Peter Yeh, Amit Sheth, Moving beyond sameAs with PLATO: Partonomy detection for Linked Data. Submitted.