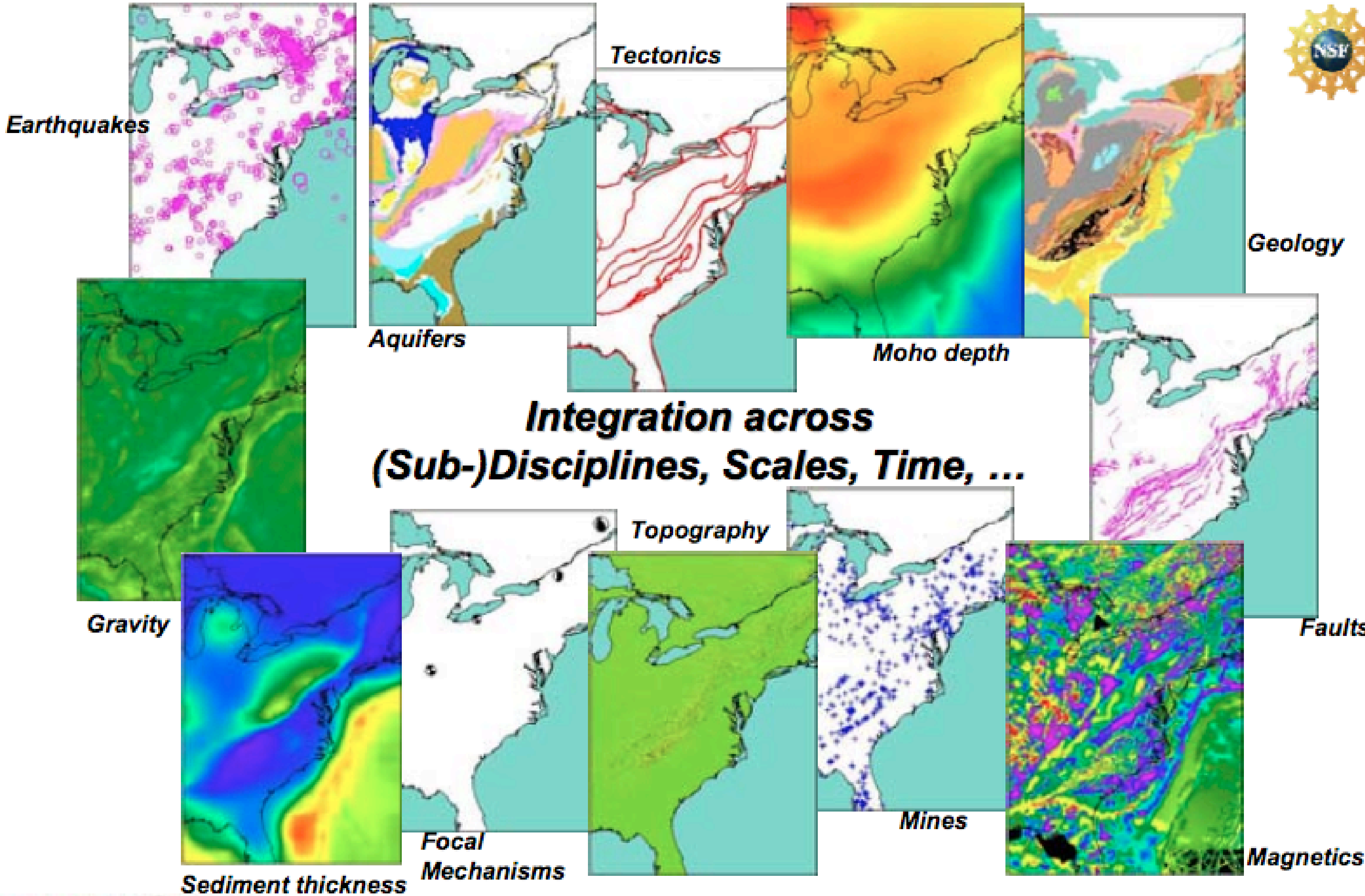
# Metadata and Semantic Technologies for Science

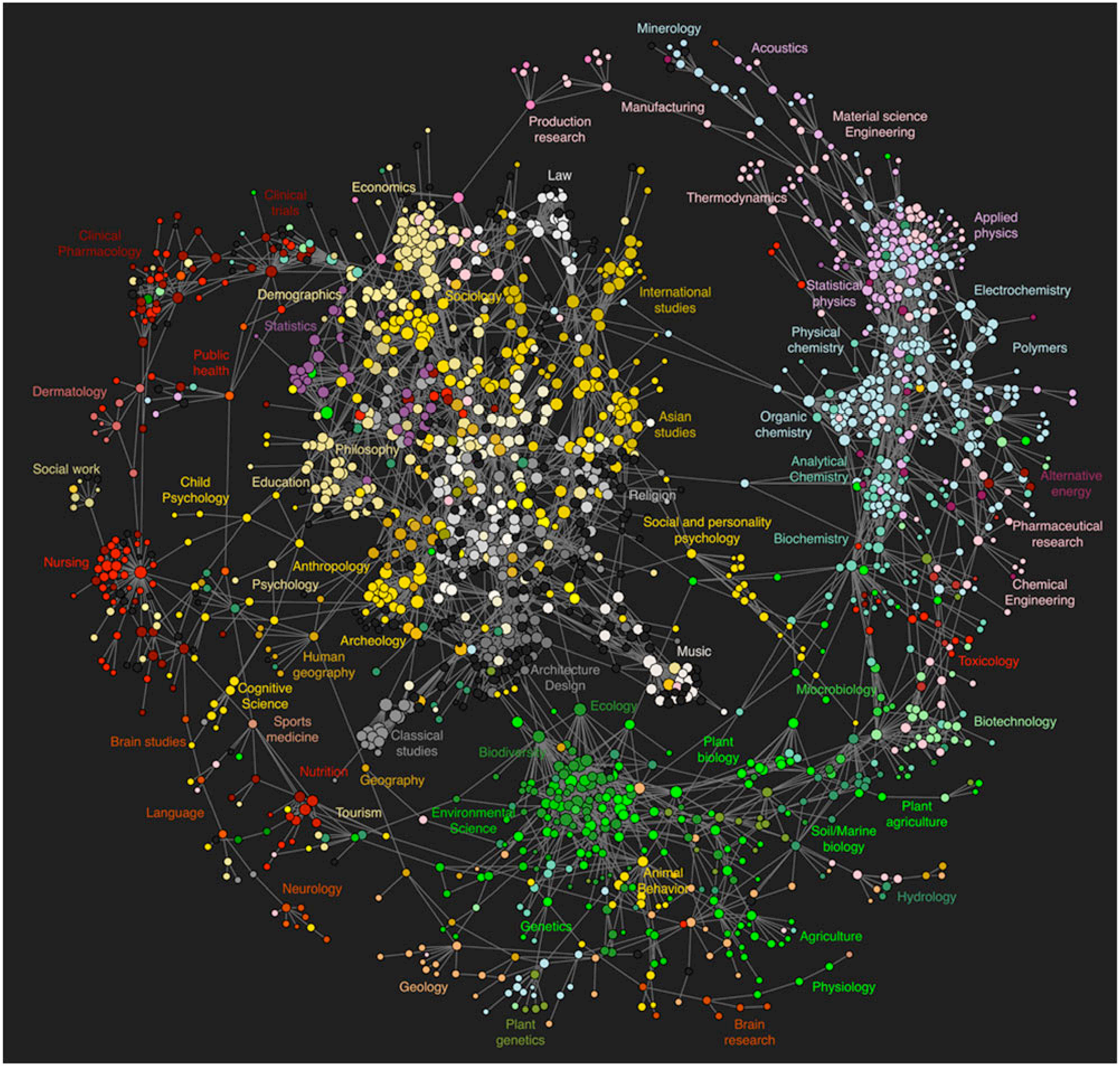
Bill Howe

# Integration across (Sub-)Disciplines, Scales, Time, …




GEON CYBERINFRASTRUCTURE FOR THE GEOSCIENCES

Src: Dogan Seber SDSC, Krishna Sinha VT




Johan Bollen
Los Alamos
PLoS ONE,
March 2009

# Metadata

- Literally: **data about data**
- Descriptive (meta-)information about the "actual" data
- Distinction between data and metadata is not always clear
- "Someone's metadata is someone else's data"
- Related notions:
  - **schema, data dictionary, conceptual model, ontology,**
- **Traditional example: Library catalog card** contains metadata about the contents and location of books

## Perspectives on Metadata

- Digital Library perspective?
  - **Resource descriptions**
  - **Operation: retrieval**
- Database perspective?
  - **Schema, describing the logical ("almost physical") structure of the data**
  - **Operation: query (query as "computation" -- more than retrieval)**

*paraphrasing Bertram Ludascher*

## Metadata Standards

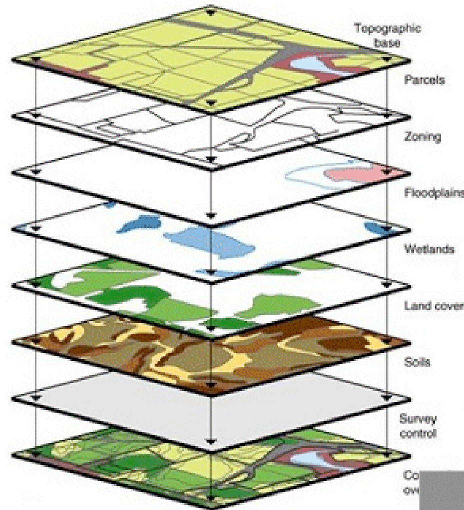
- **Content standards:**
  - which pieces of information are to be recorded (e.g. DC)
- **Value standards:**
  - how is the information to be recorded (= DC *encoding schemes)*
  - formats (ISO date format, NCA name formats, AACR)
  - lists of valid values (thesauri, controlled vocabularies, authority files)
- **Structure standards:**
  - how the information is to be grouped and labelled for use by computers and humans (XML schemas, MARC)

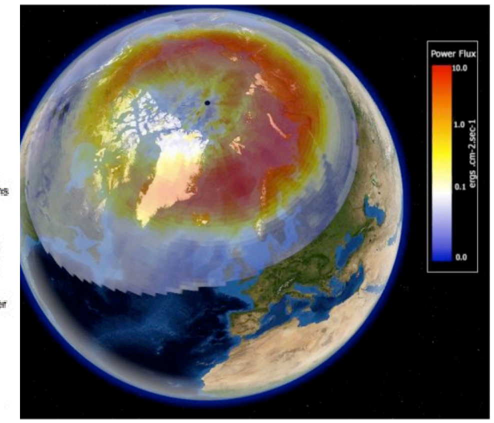
## Content Standard Example: Dublin Core

- 1. Title
- 2. Creator
- 3. Subject
- 4. Description
- 5. Publisher
- 6. Contributor
- 7. Date
- 8. Type
- 9. Format
- 10. Identifier
- 11. Source
- 12. Language
- 13. Relation
- 14. Coverage
- 15. Rights

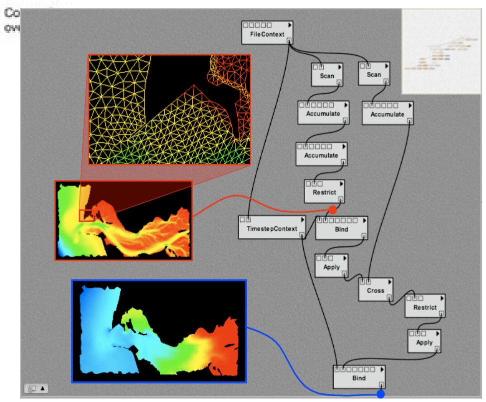
# Why Metadata?

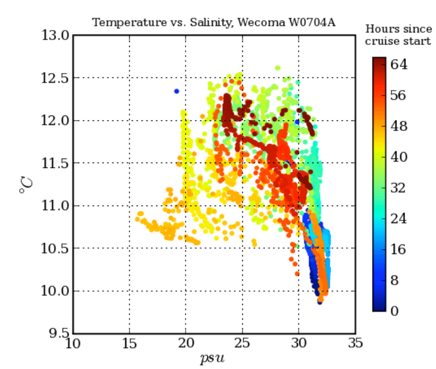
# Types of Integration

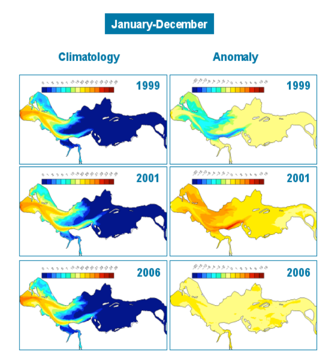


- Spatiotemporal co-registration/“overlay”
  - **“Desktop integration”**
- Schema-oriented integration
  - **View-based**
  - **Task: Derive mediated schema**
  - **Global as View or Local as View**
- Application/Process Integration
  - **Workflows, Mashups**
- “Manual”, Programmatic Integration
  - **Statistics, Visualization**





## What's Missing?

- Spatiotemporal co-registration
  - **Assumes agreement on coordinate systems**
- Schema-oriented integration
  - **Assumes agreement on (complex) data model**
- Application integration
  - **Assumes agreement on process descriptions**
- Programmtic integration
  - **Few assumptions, but no real help either**
- All of these assume agreement on semantics

# Goal: Interoperability

Resolve heterogeneity in…

| | |
|---|---|
| Synthesis | Workflow, applications, “the science” |
| Semantics | RDF, OWL |
| Structure | XML, RDBMS |
| System | files, byte-order, networking |

*Paraphrasing Bertram Ludascher*

# Example: XML

No commitment to or specification of semantics

What is a “name”?

What are the units of “length”?

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- This XML document was generated by RCCOBXML -->
- <TRAINDOC>
  - <SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap"
      xmlns:uk="http://www.greenwichmeantime.co.uk"
      xmlns:us="http://www.easternstandardtime.com">
    - <SOAP-ENV:Body>
      - <TRAIN Date="18/03/2003" Time="13:00">
        - <LOCOMOTIVE>
            <Name>Thomas</Name>
            <Length>12,500.00</Length>
            <Weight>3,400</Weight>
          </LOCOMOTIVE>
        - <CARRIAGE>
            <Name>Annie</Name>
            <Length>10,000.00</Length>
            <Weight>2,000</Weight>
          - <Other_Information>
              Room for
              <count>100</count>
              standing & seated
            </Other_Information>
          </CARRIAGE>
        - <CARRIAGE>
            <Name>Clarabel</Name>
            <Length>12,500.00</Length>
            <Weight />
          </CARRIAGE>
        </TRAIN>
      </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
</TRAINDOC>
```

# Example: Relational

| type | name | length | weight | other |
|---|---|---|---|---|
| locomotive | Thomas | 12,500 | 3,400 | |
| carriage | Annie | 10,000 | 2,000 | room for 100 |
| carriage | Clarabel | 12,500 | | |

No commitment to or specification of semantics

*What is a "name"?*

*What are the units of "length"?*

## Semantic Web

- “The Semantic Web is an extension of the current Web in which information is given well-defined meaning, enabling computers and people to work in better cooperation”
- “The web will reach its full potential when it becomes an environment where data can be shared and processed by automated tools as well as by people.”

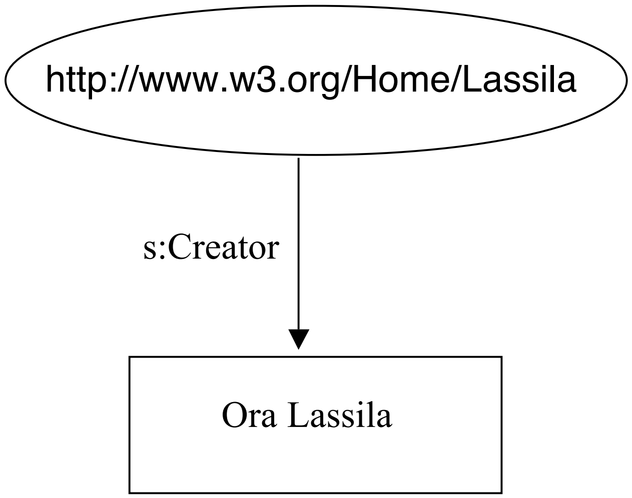
  -- Tim Berners-Lee and Eric Miller

## RDF: Web based data model

- Semantic Web: beyond machine readable to *machine understandable.*
- Resource Description Framework *is the W3C language for describing metadata on the Web.*

- **RDF** consists of two parts
  1. **RDF Model (a set of triples)**
  2. **RDF Syntax (different XML serialization syntaxes)**
  - **RDF a small set of modelling primitives + syntax**
  - **RDF *does not* commit to a domain vocabulary**
- RDF Schema for definition of Vocabularies (simple Ontologies) for RDF (and in RDF)

# A simple RDF example




**Triples**

Resource (subject) http://www.w3.org/Home/Lassila

Property (predicate) http://www.schema.org/#Creator

Value (object) "Ora Lassila"

- Resources
  - **A thing you can reference (URI)**
  - **RDF definitions are themselves Resources.**
- Properties
  - **slots, defines relationship to other resources or atomic values**
  - **Similar to Frames.**
- Statements
  - **"Resource has Property with Value"**
  - **Values can be resources or atomic XML Schema data types.**
- Directed graph

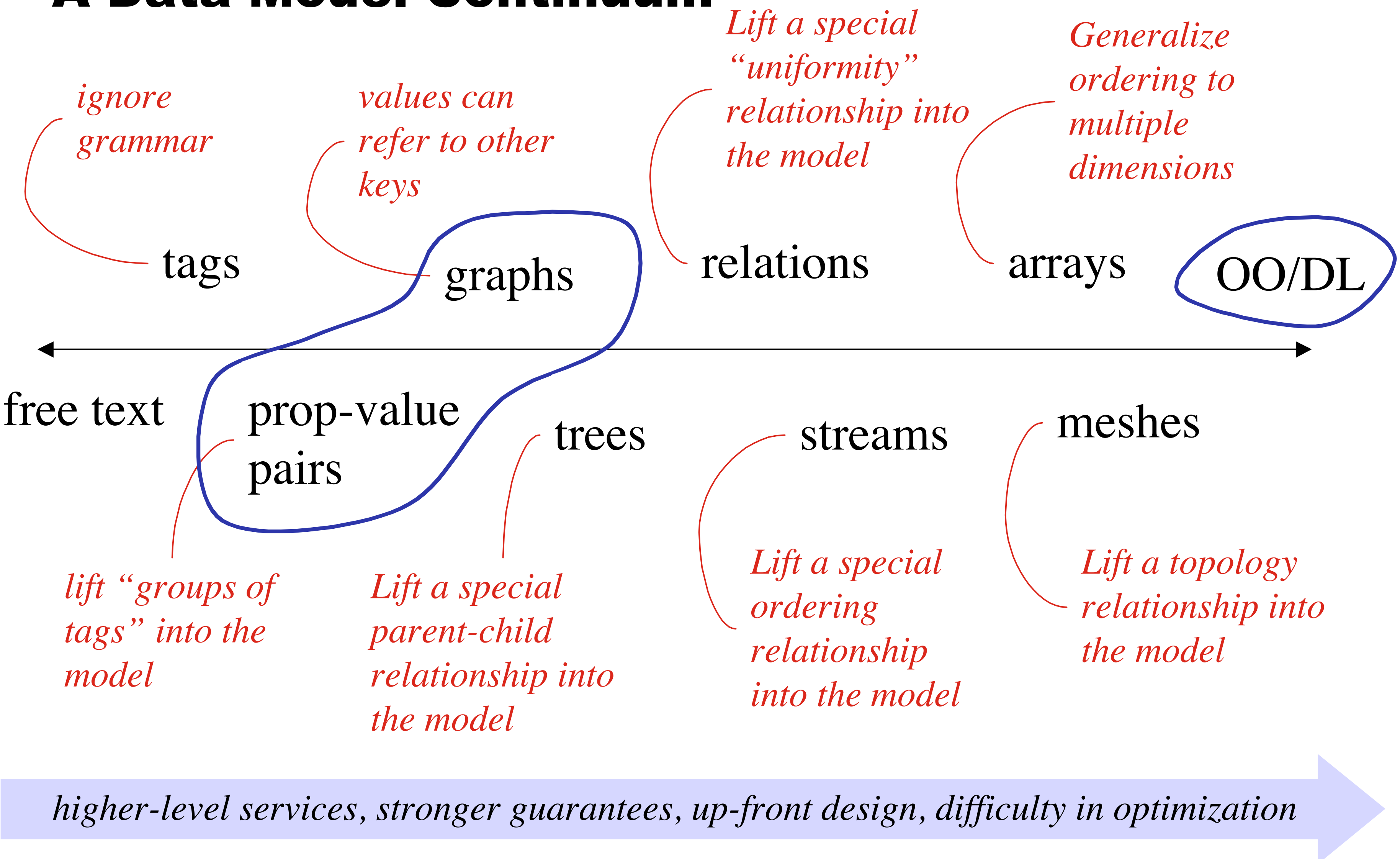
## Why RDF?  1st Attempt

- Universal machine-readable standard for representing semantics
- "Push-based" integration
  - **Describe everything precisely up-front, and integration is easier**
  - **So easy, in fact, that little autonomous agents will be able to scurry around the web booking flights on your behalf**

# An allegory

xml 30.0 rdf 1.00 json 1.00
Search Volume index
Google Trends
A
B
C
D
E
F
40.0
20.0
0
2004
2005
2006
2007
2008
2009
2010
News reference volume
0

# A Data Model Continuum



## Why RDF? 2nd attempt

- "Lowest common denominator" data model
- Just enough structure to represent explicit machine-readable relationships
  - **unlike free text, tags, key-value pairs**
- Not so much structure as to require complicated, brittle modeling decisions
  - **unlike XML, RDBMS, OO**

## Roadmap

- Background and Motivation
- Overview of Ontologies and Reasoning
- Storing and Querying RDF
- Papers

# Ontologies

- **What is an ontology? An ontology usually**
  - **specifies a theory** (a set of logic **models models**) by…
  - **defining** and **relating** …
  - **concepts** representing features of a domain of interest
- **Also overloaded (sloppy) for:**
  - **Controlled vocabularies**
  - **Database schema** (relational, XML Schema/DTD)
  - **Conceptual schema** (ER, UML)
  - **Thesauri** (synonyms, broader term/narrower term)
  - **Taxonomies** (classifications)
  - **Informal/semi-formal knowledge representations** Concept spaces, concept maps, Labeled graphs / semantic networks (RDF)
  - **Formal ontologies**, e.g., in [Description] Logic (OWL) formalization of a specification constrains possible interpretation of terms

*src: Carole Goble*

# Inference machinery

- Any knowledge is
  - **A language representation +**
  - **An inference mechanism for deduction**

- First order calculus + resolution method
- Frames + matcher
- Semantic net + graph traversal
- Description logics + theorem prover

# Ontologies: A Semantic Continuum

*Pump: "a device for moving a gas or liquid from one place or container to another"*

*(pump has (superclasses (...))*

| Shared human consensus | Text descriptions | Semantics hardwired; used at runtime | Semantics processed and used at runtime |
|---|---|---|---|
| *Implicit* | *Informal* *(explicit)* | *Formal* *(for humans)* | *Formal* *(for machines)* |

Further to the right means:

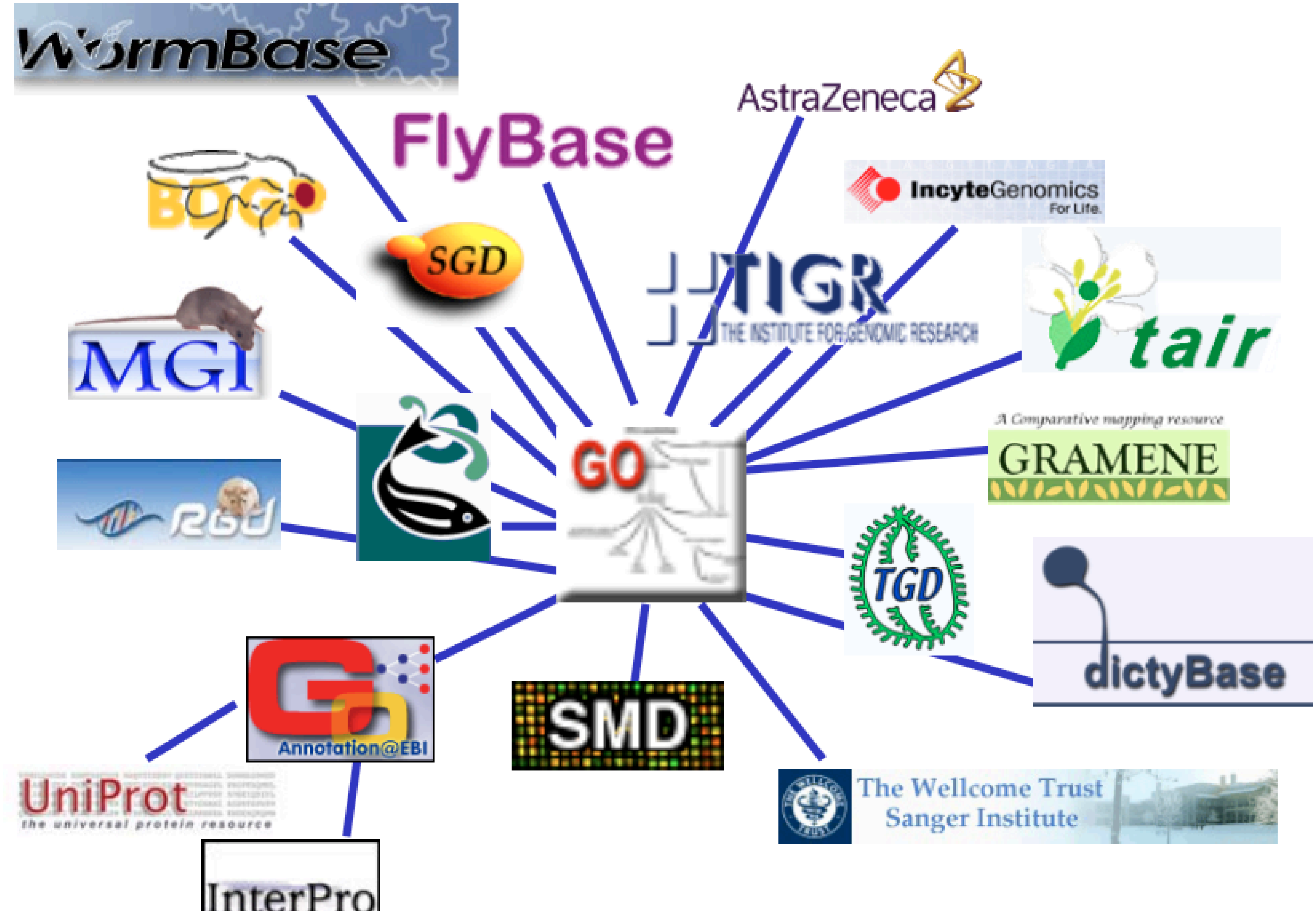
- Less ambiguity
- More likely to have correct functionality
- Better inter-operation (hopefully)

- Less hardwiring
- More robust to change
- **More difficult**

**Src: [Mike Uschold, Boeing Corp]**

## Example: Gene Ontology (GO)

- Well-known life science ontology
- Three things described
  - **sub-cellular localization**
  - **molecular function (what the gene does)**
  - **biological process (the cellular, developmental or physiological events the gene product is involved in)**
- Example:
  - Taking 4 kinases at random from 4 different organisms (Fly: ZWIM, Mouse: DAPK2 and Arabidopsis: KIPK and yeast: WEE1), based on gene nomenclature it is not clear that all of these genes encode protein kinases!
  - However, in GO, all 4 genes are annotated to the term "*protein kinaseactivity*" making it simple to find genes with similar functions in diverse species.




http://www.geneontology.org/GO.people.shtml

# RDF Schema (RDFS)

- RDF just defines the data model.
- Need for definition of vocabularies for the data model - an Ontology Language!
- RDF-Schemas describe rules for using RDF properties
  - **Define a domain vocabulary for RDF**
  - **Organise this vocabulary in a typed hierarchy**
- RDF Schemas are Web resources (and have URIs) and can be described using RDF.
- Are not to be confused with XML Schemas.
- RDFS is the framework for a vocabulary.

# RDF Schema Model

- Each property specifies what classes of subjects and objects it relates. New properties can be added to a class without modifying the class
  - **resource, class, subClassOf, type**
  - **property, subPropertyOf**
  - **domain, range, constraintResource, constraintProperty**
- Definitions can include constraints which express validation conditions
  - **domain constraints link properties with classes**
  - **range constraints limit property values**
- BUT expressive inadequacy and poorly defined semantics

## Blank Nodes

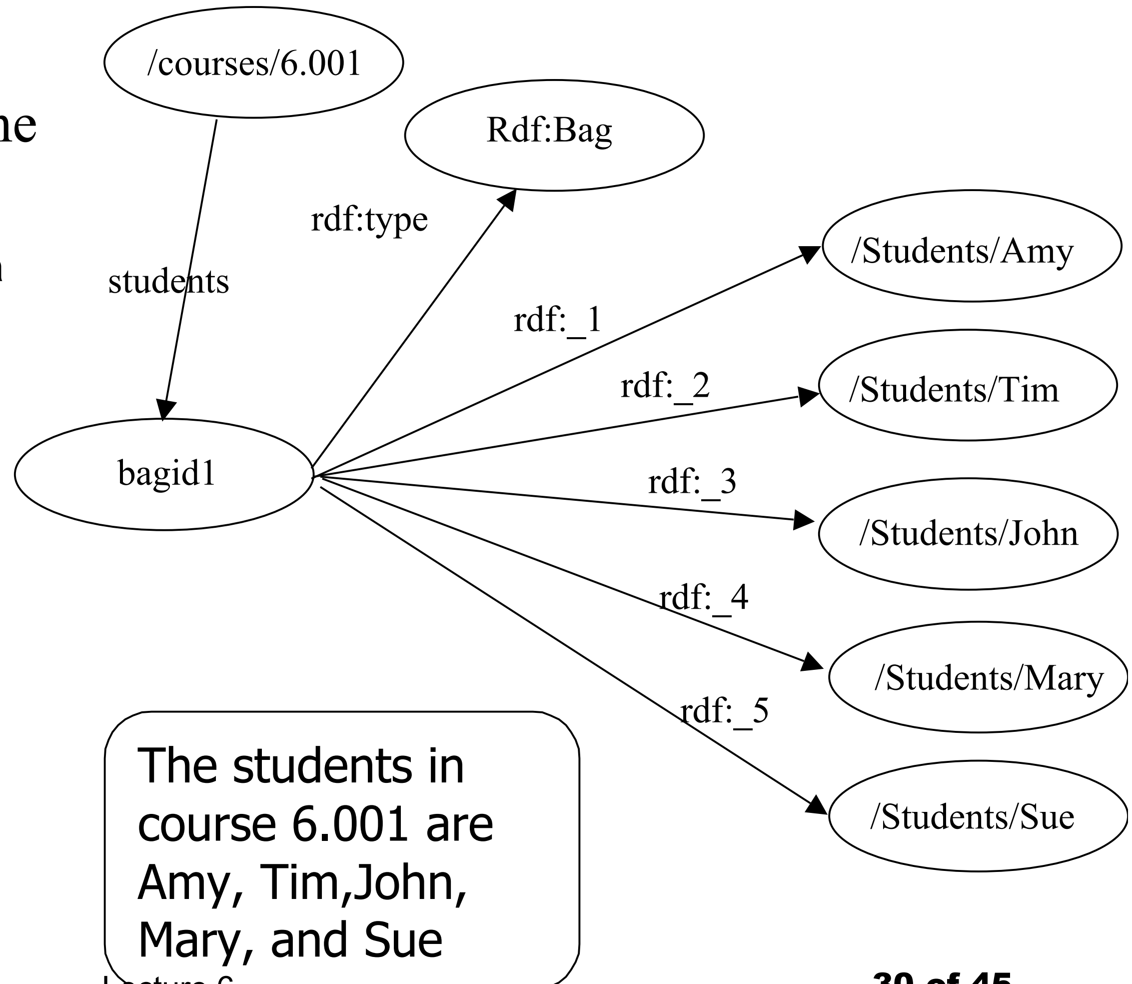
- Existentially quantified variables

| | | |
|---|---|---|
| ex:John | foaf:knows | _:p1 |
| _:p1 | foaf:birthDate | 04-21 |

- Use cases
  - **Reasoning/Graph entailment**
  - **Modeling complex structures; Collections**
  - **Anonymous classes in OWL**

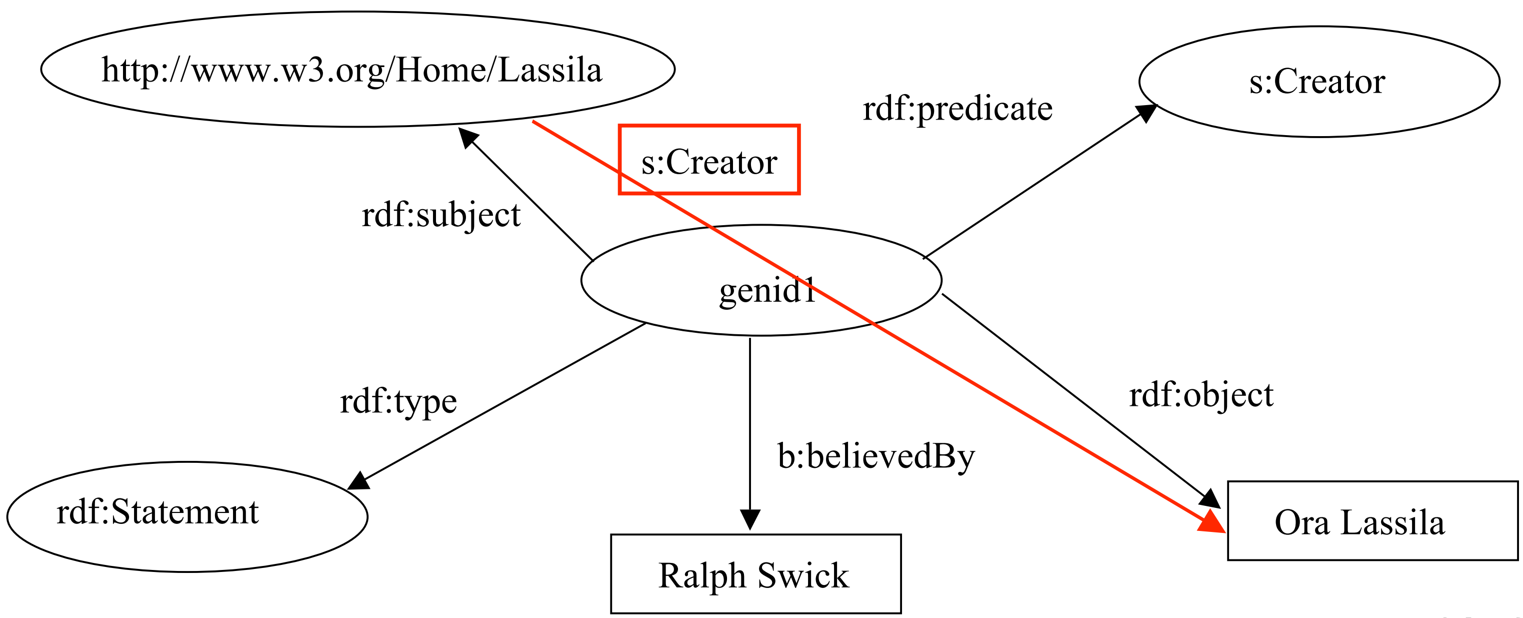
# Collection Containers

- Multiple occurrences of the same PropertyType doesn't establish a relation between the values
  - **The Millers own a boat, a bike, and a TV set**
- RDF defines three special Resources:
  - **Bag**
  - **Sequence**
  - **Alternative**



# Reification: Statements about statements

- Transform them into Resources.
- *Ralph Swick believes that*
  - ***the creator of the resource http://www.w3.org/Home/Lassila is Ora Lassila***

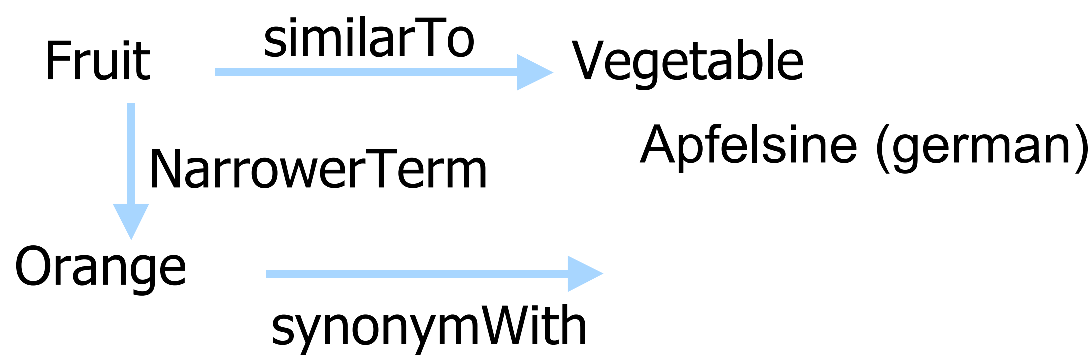


## Related Concept: Controlled Vocabulary

- Domain model
- No relationships
- Standardizes terminology to reduce semantic heterogeneity

# Related concept: Thesauri

Example:




- Graph with labels edges (similar, nt, bt, synonym)
- Fixed set of edge labels (aka relations)
- no instances

- Well known in library science
- cf. terminologies / classifications (Dewey)

## Related Concept: Topic Maps

- Topics
  - **concepts/categories/classes/etc.**
- Associations
  - **n-ary relationships (key difference with RDF)**
- Ocurrences
  - **instances**

- Standardized: ISO/IEC 13250:2003
- To enable information resources to be classified and navigated in a consistent manner

## Related Technology: Description Logics

- KL-ONE [Brachman and Schmolze, 1985]
  - **Inheritance**
- AL
  - **Atomic negation, Concept intersection, Universal restrictions, existential quantification**
- ALC (+concept complement)
- SHIQ (+cardinality restrictions, inverse relationships)
- Many more permutations…

# Related Concept: Description Logics

- DAML+ OIL equivalent to the expressive Description Logic (an extension of) SHIQ DL
- The descendants of frame systems and object hierarchies via KL-ONE.
- Core distinction between  T-Box ≈ Schema) and (A-Box ≈ Database tuples)
- Many years of DL research
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)

## OO languages

- How are they different?
- What relationships are “lifted” into the model?

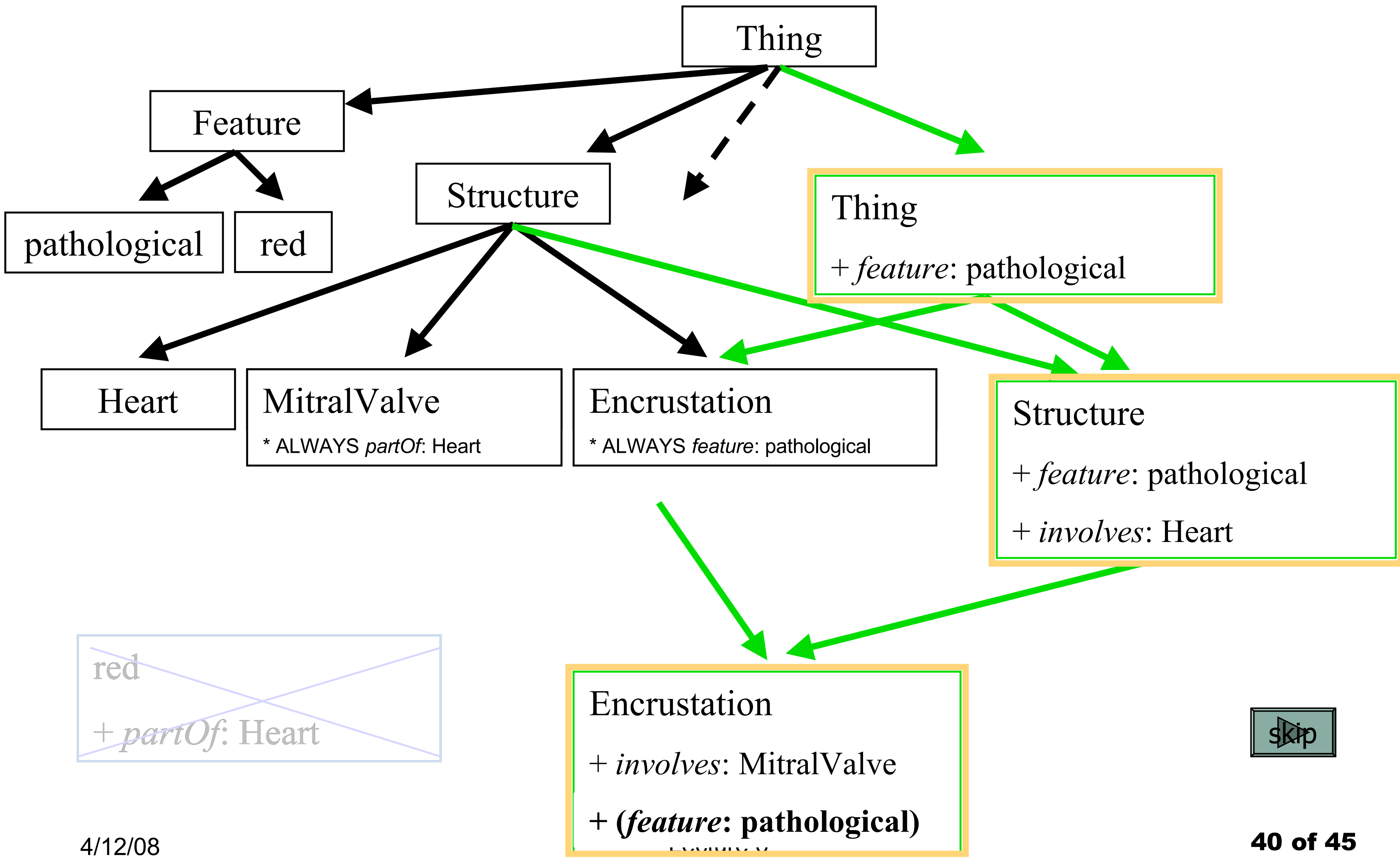
## History: DAML+OIL

- OIL : developed by group of (largely) European researchers.
- DAML- ONT: developed by group of (largely) US researchers (in DARPA DAML programme).
- Efforts merged to produce DAML+ OIL.
- Development was overseen by joint EU/ US committee.
- Now submitted to W3C as basis for standardisation WebOnt working group developing language standard.
- Subsumed by OWL

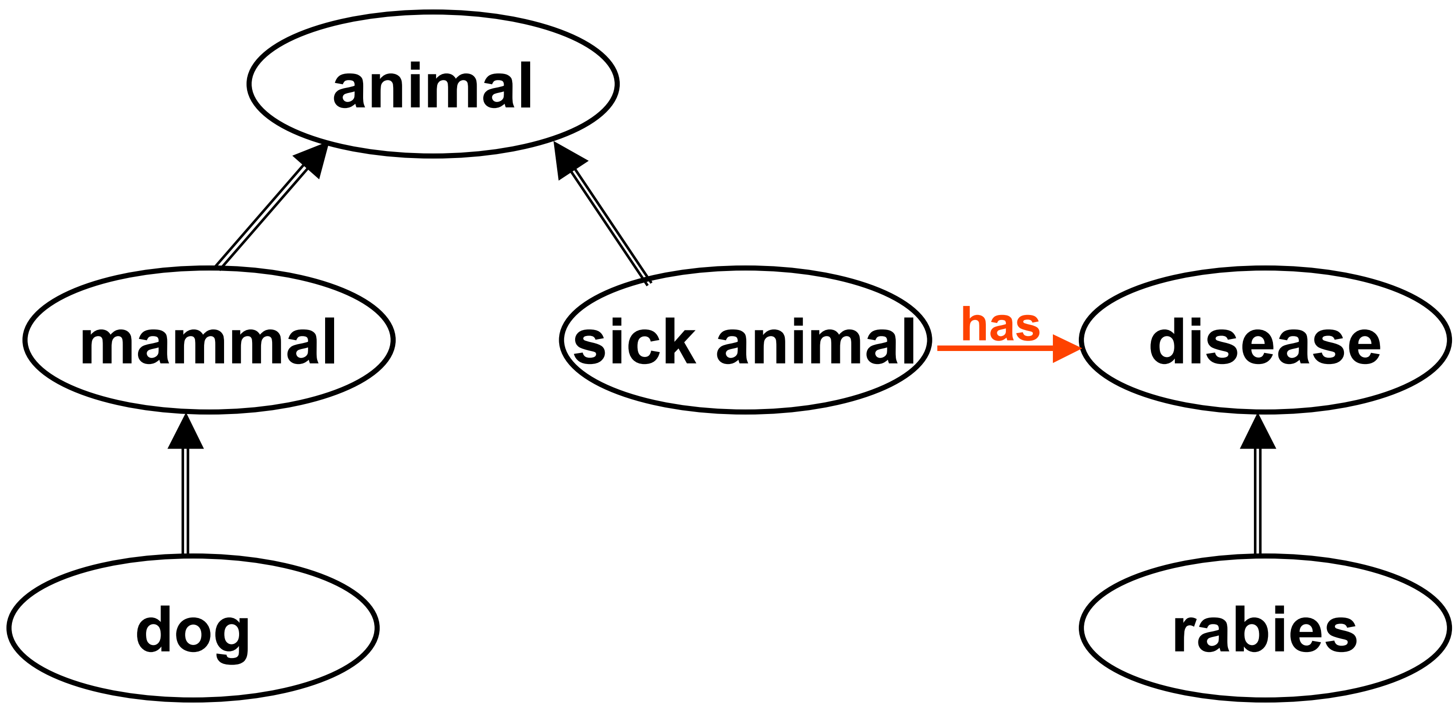
# OWL Semantics

| Abstract Syntax | DL Syntax | Semantics |
|---|---|---|
| | Descriptions ($C$) | |
| $A$ (URI Reference) | A | $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ |
| `owl:Thing` | $\top$ | $\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$ |
| `owl:Nothing` | $\bot$ | $\texttt{owl:Nothing}^{\mathcal{I}} = \emptyset$ |
| `intersectionOf`$(C_1\ C_2 \ldots)$ | $C_1 \sqcap C_2$ | $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$ |
| `unionOf`$(C_1\ C_2\ \ldots)$ | $C_1 \sqcup C_2$ | $C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$ |
| `complementOf`$(C)$ | $\neg C$ | $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$ |
| `oneOf`$(o_1 \ldots)$ | $\{o_1, \ldots\}$ | $\{o_1^{\mathcal{I}}, \ldots\}$ |
| `restriction`$(R$ `someValuesFrom`$(C))$ | $\exists R.C$ | $\{x \mid \exists y\ \ (x,y) \in R^{\mathcal{I}} \cup y \in C^{\mathcal{I}}\}$ |
| `restriction`$(R$ `allValuesFrom`$(C))$ | $\forall R.C$ | $\{x \mid \forall y\ \ (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$ |
| `restriction`$(R$ `hasValue`$(o))$ | $R : o$ | $\{x \mid (x, o^{\mathcal{I}}) \in R^{\mathcal{I}}\}$ |
| `restriction`$(R$ `minCardinality`$(n))$ | $\geqslant n R$ | $\{a \in \Delta^{\mathcal{I}} \mid \ \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$ |
| `restriction`$(R$ `maxCardinality`$(n))$ | $\leqslant n R$ | $\{a \in \Delta^{\mathcal{I}} \mid \ \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$ |
| `restriction`$(U$ `someValuesFrom`$(D))$ | $\exists U.D$ | $\{x \mid \exists y\ \ (x,y) \in U^{\mathcal{I}} \cup y \in D^{\mathcal{D}}\}$ |
| `restriction`$(U$ `allValuesFrom`$(D))$ | $\forall U.D$ | $\{x \mid \forall y\ \ (x,y) \in U^{\mathcal{I}} \rightarrow y \in D^{\mathcal{D}}\}$ |
| `restriction`$(U$ `hasValue`$(v))$ | $U : v$ | $\{x \mid (x, v^{\mathcal{I}}) \in U^{\mathcal{I}}\}$ |
| `restriction`$(U$ `minCardinality`$(n))$ | $\geqslant n U$ | $\{a \in \Delta^{\mathcal{I}} \mid \ \lvert\{b \mid (a,b) \in U^{\mathcal{I}}\}\rvert \geq n\}$ |
| `restriction`$(U$ `maxCardinality`$(n))$ | $\leqslant n U$ | $\{a \in \Delta^{\mathcal{I}} \mid \ \lvert\{b \mid (a,b) \in U^{\mathcal{I}}\}\rvert \leq n\}$ |
| | Data Ranges ($D$) | |
| $D$ (URI reference) | $D$ | $D^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^{\mathcal{I}}$ |
| `oneOf`$(v_1 \ldots,)$ | $\{v_1 \ldots,\}$ | $\{v_1^{\mathcal{I}} \ldots,\}$ |
| | Object Properties ($R$) | |
| $R$ (URI reference) | $R$ | $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ |
| | $R^-$ | $(R^{\mathcal{I}})^-$ |
| | Datatype Properties ($U$) | |
| $U$ (URI reference) | $U$ | $U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathcal{D}}^{\mathcal{I}}$ |
| | Individuals ($o$) | |
| $o$ (URI reference) | $o$ | $o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ |
| | Data Values ($v$) | |
| $v$ (RDF literal) | $v$ | $v^{\mathcal{D}}$ |

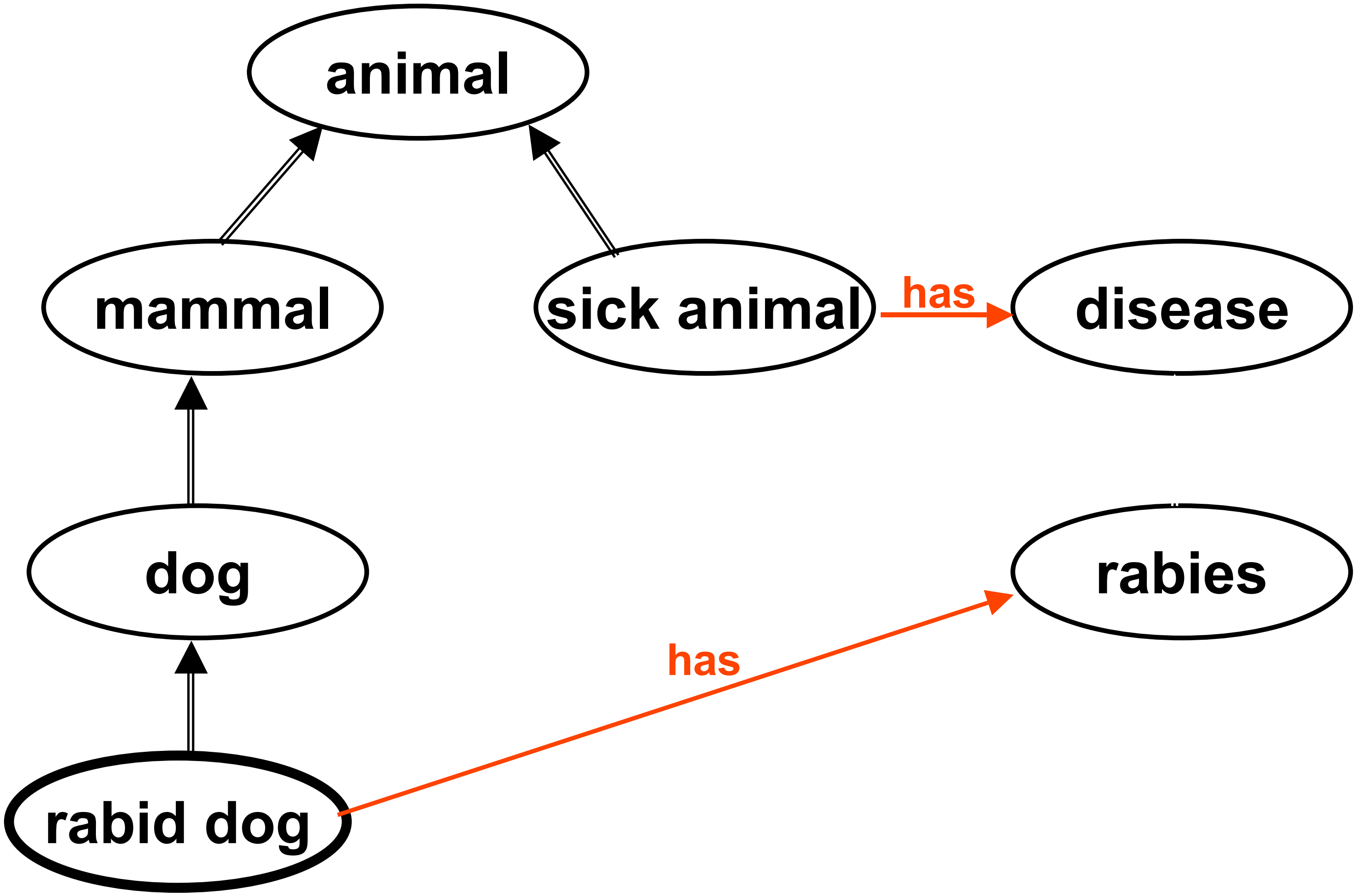
## Logic Based Ontologies



# A Simple Ontology [Swartout]



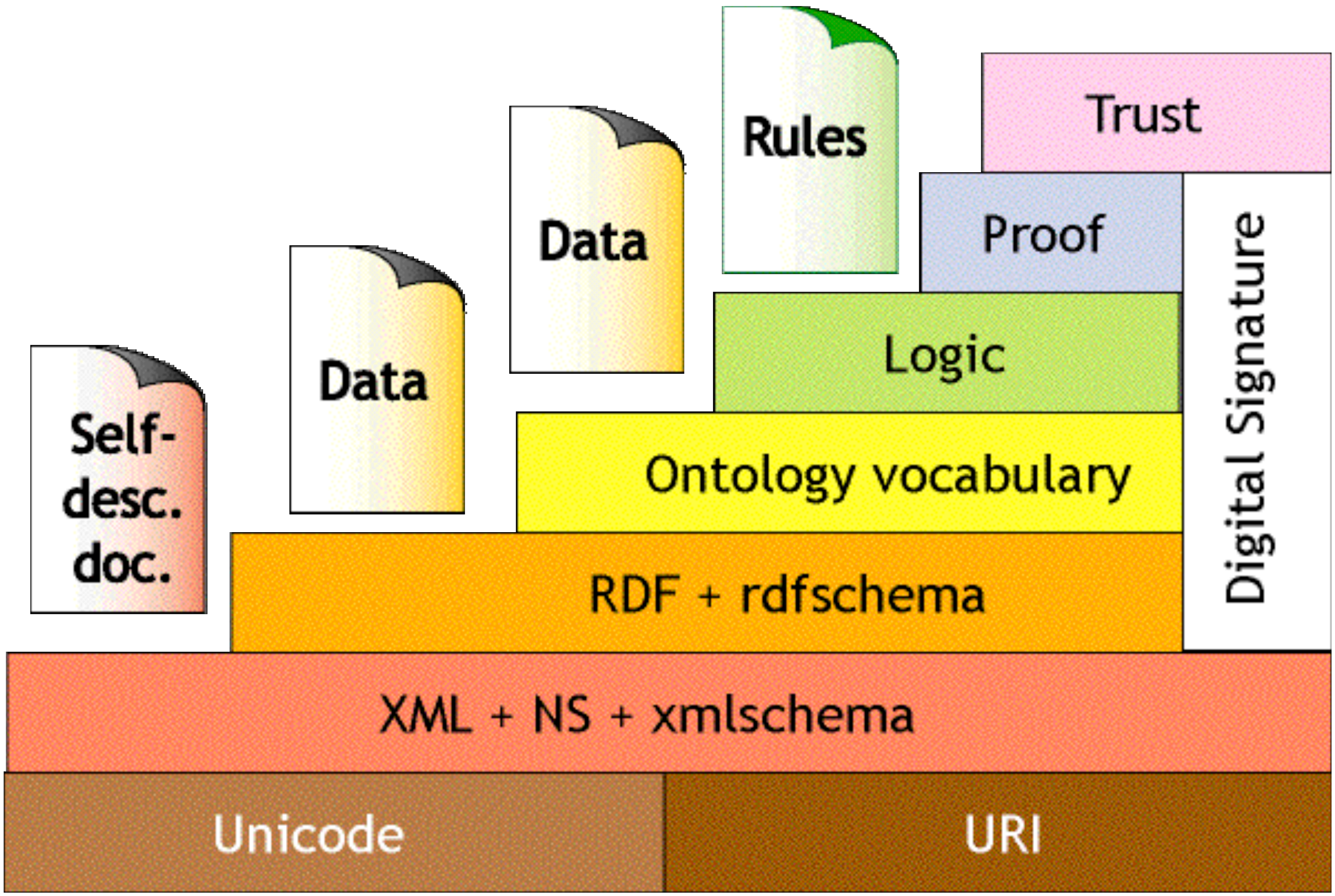
# Defining a “rabid dog”



# Reasoning Tasks

- Satisfiability of a concept
  - **determine whether a description of the concept is not contradictory, i.e., whether an individual can exist that would be instance of the concept.**
- Subsumption of concepts
  - **determine whether concept C subsumes concept D, i.e., whether description of C is more general than the description of D.**
- Consistency of individuals with respect to concept
  - **determine whether individuals in ABox do not violate descriptions and axioms described by TBox.**
- Membership
  - **check whether the individual is an instance of a concept**
- Compute Extent
  - **find all individuals that are instances of a concept**
- Realization of an individual
  - **find all concepts which the individual belongs to, especially the most specific ones**

## Stack of languages:



# Web Language Stack summary

- **XML**:
  - interchange syntax, no semantics
- **RDF**:
  - Data model, some semantics & inference
- **RDF Schema**:
  - concept modelling, more semantics & inference
- **OWL**:
  - more expressive ontology language;
  - quite expressive; expensive inference

# Roadmap

- Background and Motivation
- Ontologies and Reasoning
- Storing and Querying RDF
- Papers

# OWL Example

```
<owl:Class rdf:ID="Lookout">
  <owl:equivalentClass>
    <owl:Class rdf:ID="Scout"/>
  </owl:equivalentClass>
  <rdfs:label>Lookout</rdfs:label>
</owl:Class>

<owl:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="http://…"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://…"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://…/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en">
  An individual human being.
</rdfs:comment>
  <rdfs:label>Person</rdfs:label>
</owl:Class>
```

```
<owl:Class rdf:ID="PoliceOfficer">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:comment xml:lang="en">
  A warranted employee of a police force.
</rdfs:comment>
  <rdfs:label>Police Officer</rdfs:label>
</owl:Class>

<owl:Class rdf:about="#Scout">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:label>Scout</rdfs:label>
</owl:Class>
```

```
<owl:Class rdf:ID="CashierPost">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="Cashier"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="associatedWith"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Post"/>
  </rdfs:subClassOf>
  <rdfs:label>Cashier Post</rdfs:label>
</owl:Class>
```

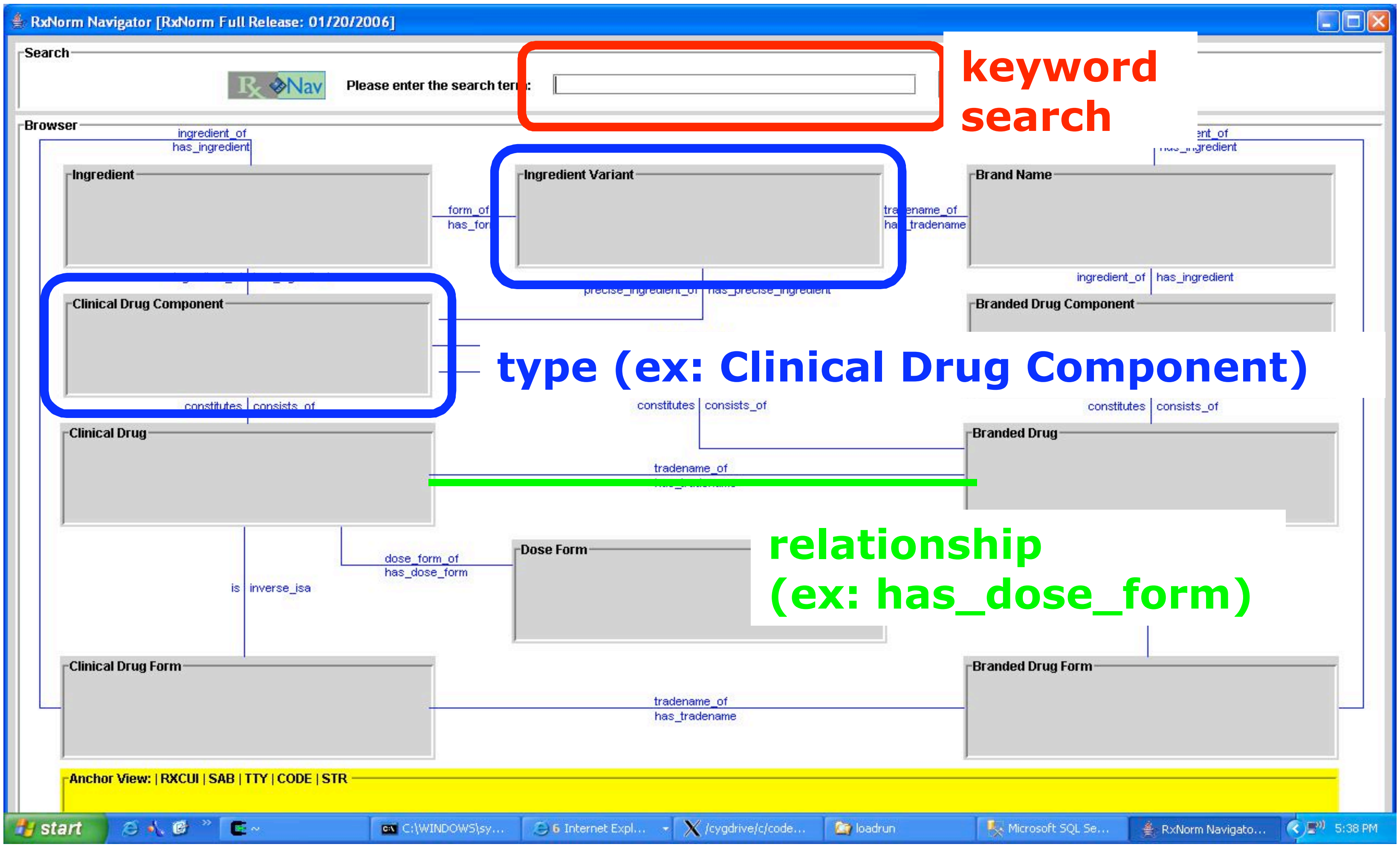
# RDF Example

```
<rdf:Description rdf:about="http://www.jhuapl.edu/merc/ba-instance-example#edge-31066">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  <rdf:subject rdf:resource="#Person-24"/>
  <rdf:predicate rdf:resource="http://bethewl1-wd1.dom1.jhuapl.edu/Ontology/BAOntology.owl#meetswith"/>
  <rdf:object rdf:resource="#Person-23"/>
  <ba:beginTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">  78397</ba:beginTime>
</rdf:Description>
<rdf:Description rdf:about="http://www.jhuapl.edu/merc/ba-instance-example#edge-31067">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  <rdf:subject rdf:resource="#Person-15"/>
  <rdf:predicate rdf:resource="http://bethewl1-wd1.dom1.jhuapl.edu/Ontology/BAOntology.owl#gf"/>
  <rdf:object rdf:resource="#Person-21"/>
  <ba:beginTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">  78399</ba:beginTime>
  <ba:endTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">78439</ba:endTime>
</rdf:Description>
<rdf:Description rdf:about="http://www.jhuapl.edu/merc/ba-instance-example#edge-31068">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  <rdf:subject rdf:resource="#Person-15"/>
  <rdf:predicate rdf:resource="http://bethewl1-wd1.dom1.jhuapl.edu/Ontology/BAOntology.owl#signals"/>
  <rdf:object rdf:resource="#Person-21"/>
  <ba:beginTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">  78399</ba:beginTime>
</rdf:Description>
```

# SPARQL Example

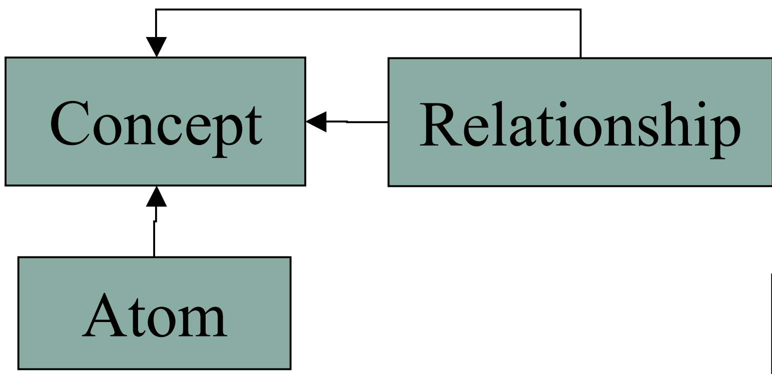
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ba: <http://bethewl1-wd1.dom1.jhuapl.edu/Ontology/BAOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX op: <http://www.w3.org/2005/xpath-functions#>

SELECT ?a ?b ?c ?t1 ?t2 ?t3
WHERE
{
?s1 rdf:type rdf:Statement .
?s1 rdf:subject ?a .
?s1 rdf:predicate ba:meetswith .
?s1 rdf:object ?b .
?s1 ba:beginTime ?t1 .
?s2 rdf:type rdf:Statement .
?s2 rdf:subject ?b .
?s2 rdf:predicate ba:signals .
?s2 rdf:object ?c .
?s2 ba:beginTime ?t2 .
?s3 rdf:type rdf:Statement .
?s3 rdf:subject ?c .
?s3 rdf:predicate ba:meetswith .
?s3 rdf:object ?a .
?s3 ba:beginTime ?t3 .
FILTER (
((?t1 < ?t2) && (?t2 < ?t3))
&& (((xsd:integer(?t2) - xsd:integer(?t1)) <= "1200"^^xsd:integer)
&& ((xsd:integer(?t3) - xsd:integer(?t2)) <= "1200"^^xsd:integer))
)
}
```

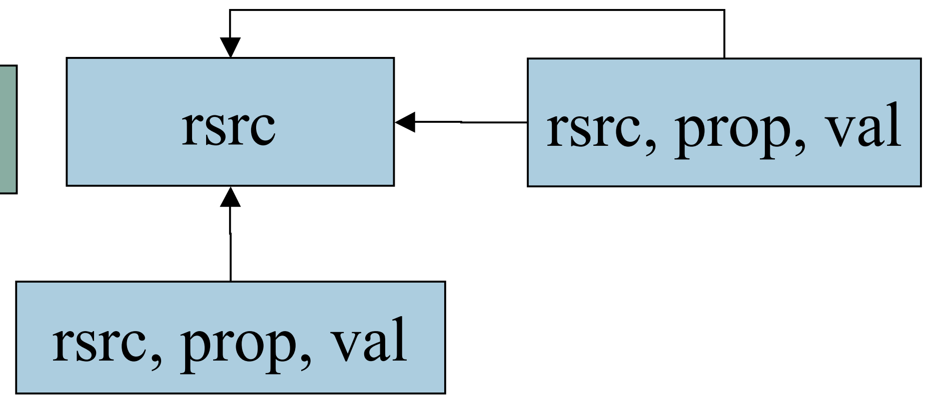
# Example: Medical Nomenclature

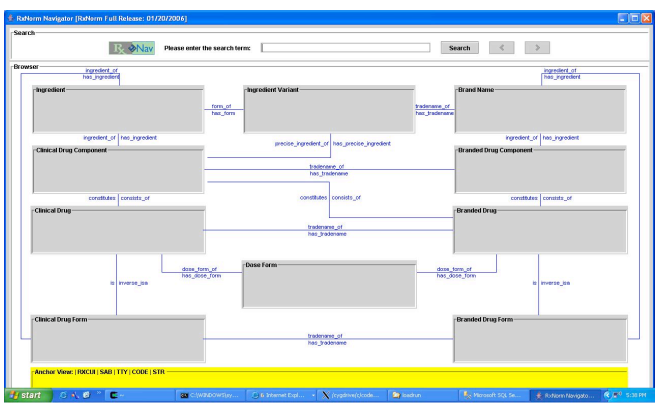



"RxNav" Interface developed by the National Library of Medicine

# Example: UMLS



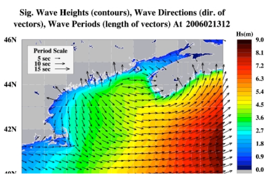



| concept | prop | value |
|---|---|---|
| 10001 | NDC | 1 |
| 10001 | ORIG_CODE | 123 |
| 10001 | ingredient_of | 10004 |
| 10001 | type | DC |

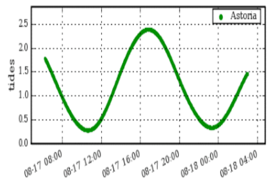
up to 23M triples describing 0.6M “concepts”

# Example: Ocean Circulation Forecasting System

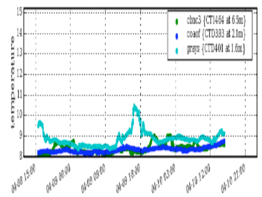
Atmospheric models



Tides

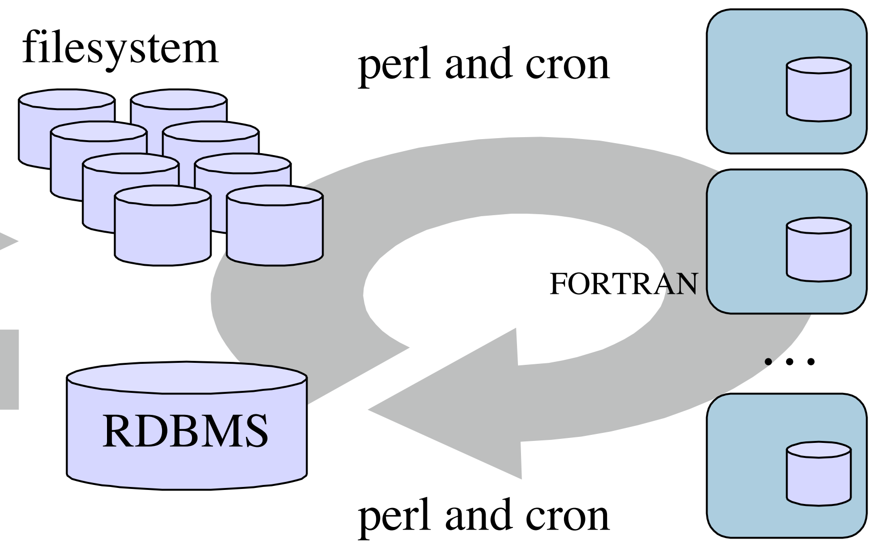


River discharge

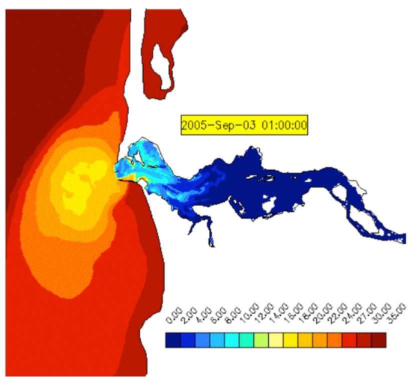


forcings (i.e., inputs)

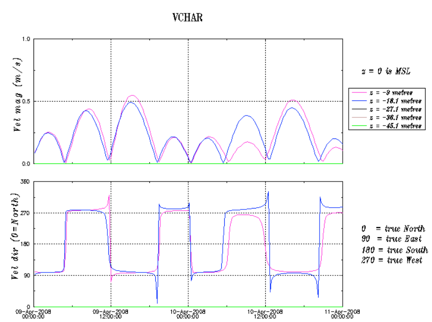
products via the web




cluster




*salinity isolines*



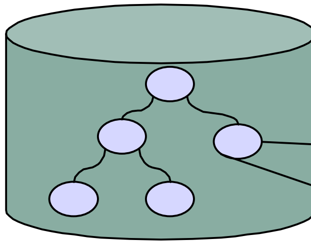

*station extractions*

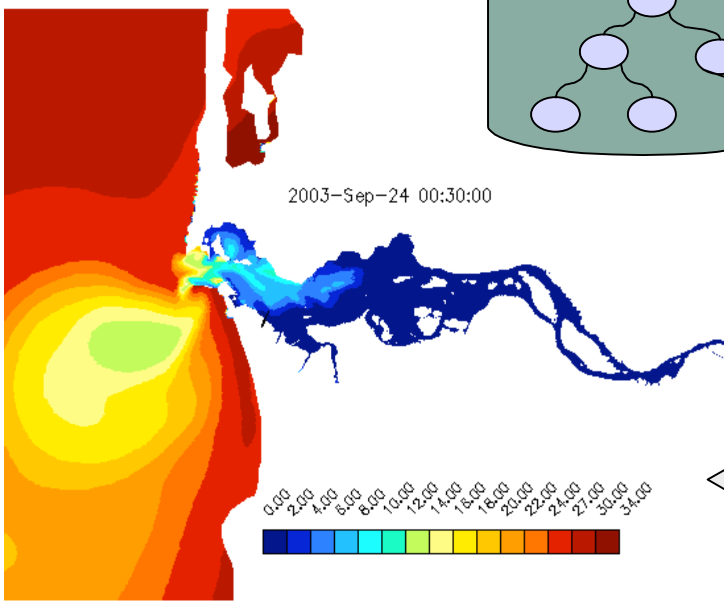
*model-data comparisons*

*Simulation results*
*Config and log files*
*Intermediate files*
*Annotations*
*Data Products*
*Relations*

CMOP Center for Coastal Margin Observation & Prediction

# Example: Ocean Circulation Forecasting






Variable = "Salinity" Depth = "7"

**…/anim-sal_estuary_7.gif**

Type = "Animation" Region = "Estuary"

| path | prop | value |
|---|---|---|
| …/anim-sal_estuary_7.gif | depth | 7 |
| …/anim-sal_estuary_7.gif | variable | salt |
| …/anim-sal_estuary_7.gif | region | estuary |
| …/anim-sal_estuary_7.gif | type | anim |

***7.5M triples describing 1M files***

# Exercise

- data1
  - **year=2004, week=24, day=1, variable=salt**
- data2
  - **year=2004, week=24, day=1, variable=temp**
- image1
  - **variable=salt, depth=7**
  - **src=data1**
- image2
  - **type=animation, variable=temp, depth=12**
  - **src=data2**

*Q1: Find all resources with variable = "salt"*

*Q2: Find all images generated from week 24 data (use "src")*

# Some Storage Models

- Schema-dependent storage (RDFS)
  - **Chen et al 06**
  - **Pan and Heflin 03**
- Indexed Triple Store
  - **RDF-3X, 10**
  - **Sesame, Broekstra 02**
  - **YARS, Harth and Decker 05**
  - **3store, Harris and Gibbons 03**
  - **Oracle, Chong et al 05 (and ICDE 08)**
- Property Tables
  - **Jena, Wilkinson 06**
  - **C-Store, Abadi et al 07**
- Horizontal DB
  - **Agrawal 01**
- Signature-based
  - **Howe 04, 06**
  - **RDFBroker 06**

## Simple Idea: Cluster by Signature

- Resources expressing the same properties clustered together
- Posit that |Signature| << |Resource|
- Queries evaluated transparently over Signature Extents

# 1) Triple Store

Triples

| rsrc | prop | value |
|---|---|---|
| 101 | depth | 7 |
| 336 | variable | temp |
| 101 | path | …/iso_e_s_7.gif |
| 101 | variable | salt |
| 843 | channel | north |
| 843 | variable | salt |
| 336 | path | …/trans_s_t.gif |
| 843 | path | …/trans_n_s.gif |
| 336 | channel | south |
| 101 | region | estuary |

*One join per condition*

A Query in SPARQL/RDQL:

```
select ?v
where
   (?r, <s:region>,   <s:estuary>),
   (?r, <s:variable>, <s:salt>),
   (?r, <s:depth>,    <s:7>),
   (?r, <s:path>, ?v)
```

… and in SQL:

```
SELECT p.value as path
  FROM Triples r, Triples v,
       Triples d, Triples p
 WHERE r.property = 'region'
   AND v.property = 'variable'
   AND d.property = 'depth'
   AND p.property = 'path'
   AND r.rsrc = v.rsrc
   AND v.rsrc = d.rsrc
   AND d.rsrc = p.rsrc
```

# 1) Triple Store, single pass trick

Triples

| rsrc | prop | value |
|---|---|---|
| 101 | depth | 7 |
| 336 | variable | temp |
| 101 | path | …/iso_e_s_7.gif |
| 101 | variable | salt |
| 843 | channel | north |
| 843 | variable | salt |
| 336 | path | …/trans_s_t.gif |
| 843 | path | …/trans_n_s.gif |
| 336 | channel | south |
| 101 | region | estuary |

```
select ?v
where
   (?r, <s:region>,   <s:estuary>),
   (?r, <s:variable>, <s:salt>),
   (?r, <s:depth>,    <s:7>)
   (?r, <s:path>, ?v)
```

```
SELECT MAX(CASE WHEN property='path' THEN value END) as path
FROM Triples
GROUP BY rsrc
HAVING
    MAX(CASE WHEN property='region' THEN value END) = 'estuary'
AND MAX(CASE WHEN property='variable' THEN value END) = 'salt'
AND MAX(CASE WHEN property='region' THEN value END) = '7'
```

# 2) Property Tables

depth

| rsrc | value |
| --- | --- |
| 101 | 7 |

variable

| rsrc | value |
| --- | --- |
| 336 | temp |
| 101 | salt |
| 843 | salt |

channel

| rsrc | value |
| --- | --- |
| 843 | north |
| 336 | south |

region

| rsrc | value |
| --- | --- |
| 101 | estuary |

path

| rsrc | value |
| --- | --- |
| 101 | …/iso_e_s_7.gif |
| 336 | …/trans_s_t.gif |
| 843 | …/trans_n_s.gif |

```
select ?p
where
   (?r, <s:region>,   <s:estuary>),
   (?r, <s:variable>, <s:salt>),
   (?r, <s:depth>,    <s:7>)
   (?r, <s:path>,     ?p)
```

```
select p.value
  from region r, variable v,
       depth d, path p
 where r.value = 'estuary'
   and v.value = 'salt'
   and d.value = '7'
   and r.rsrc = v.rsrc
   and v.rsrc = d.rsrc
   and d.rsrc = p.rsrc
```

# 3) Signature Tables

S1: variable, channel, path

| rsrc | variable | channel | path |
|---|---|---|---|
| 336 | temp | south | …/trans_s_t.gif |
| 843 | salt | north | …/trans_n_s.gif |

```
select ?p
where
   (?r, <s:region>,   <s:estuary>),
   (?r, <s:variable>, <s:salt>),
   (?r, <s:depth>,    <s:7>)
   (?r, <s:path>,     ?p)
```

S2: depth, region, variable, path

| rsrc | depth | region | variable | path |
|---|---|---|---|---|
| 101 | 7 | estuary | salt | …/iso_e_s_7.gif |

```
select path
  from S2
 where region = 'estuary'
   and variable = 'salt'
   and depth = '7'
```

# 3) Signature Tables (2)

S1: variable, channel, path

| rsrc | variable | channel | path |
|---|---|---|---|
| 336 | temp | south | …/trans_s_t.gif |
| 843 | salt | north | …/trans_n_s.gif |

```
select ?v
where
  (?r, <s:variable>, ?v)
```

S2: depth, region, variable, path

| rsrc | depth | region | variable | path |
|---|---|---|---|---|
| 101 | 7 | estuary | salt | …/iso_e_s_7.gif |

```
select variable
  from S2
UNION
select variable
  from S1
```

# Computing Signatures

| | | |
|---|---|---|
| r0 | p0 | v(0,0) |
| r2 | p1 | v(2,1) |
| r0 | p2 | v(0,2) |
| r0 | p1 | v(0,1) |
| r1 | p3 | v(1,3) |
| r1 | p1 | v(1,1) |
| r2 | p3 | v(2,3) |

External Sort

| | | |
|---|---|---|
| r0 | p0 | v(0,0) |
| | p1 | v(0,1) |
| | p2 | v(0,2) |
| r1 | p1 | v(1,1) |
| | p3 | v(1,3) |
| r2 | p1 | v(1,1) |
| | p3 | v(1,3) |

Nest

| | | | |
|---|---|---|---|
| r0 | hash(S0) | p0, p1, p2 | v(0,0), v(0,1), v(0,2) |
| r1 | hash(S1) | p1, p3 | v(1,1), v(1,3) |
| r2 | hash(S2) | p1, p3 | v(1,1), v(1,3) |

# Computing Signatures

| | | | |
|---|---|---|---|
| hash(S0) | p0, p1, p2 | r0 | v(0,0), v(0,1), v(0,2) |
| hash(S1) | p1, p3 | r1 | v(1,1), v(1,3) |
| | | r2 | v(1,1), v(1,3) |

signatures

| sighash | signature |
|---|---|
| hash(S0) | p0, p1, p2 |
| hash(S1) | p1, p3 |

hash(S0)

| rsrc | p0 | p1 | p2 |
|---|---|---|---|
| r0 | v(0,0) | v(0,1) | v(0,2) |

hash(S1)

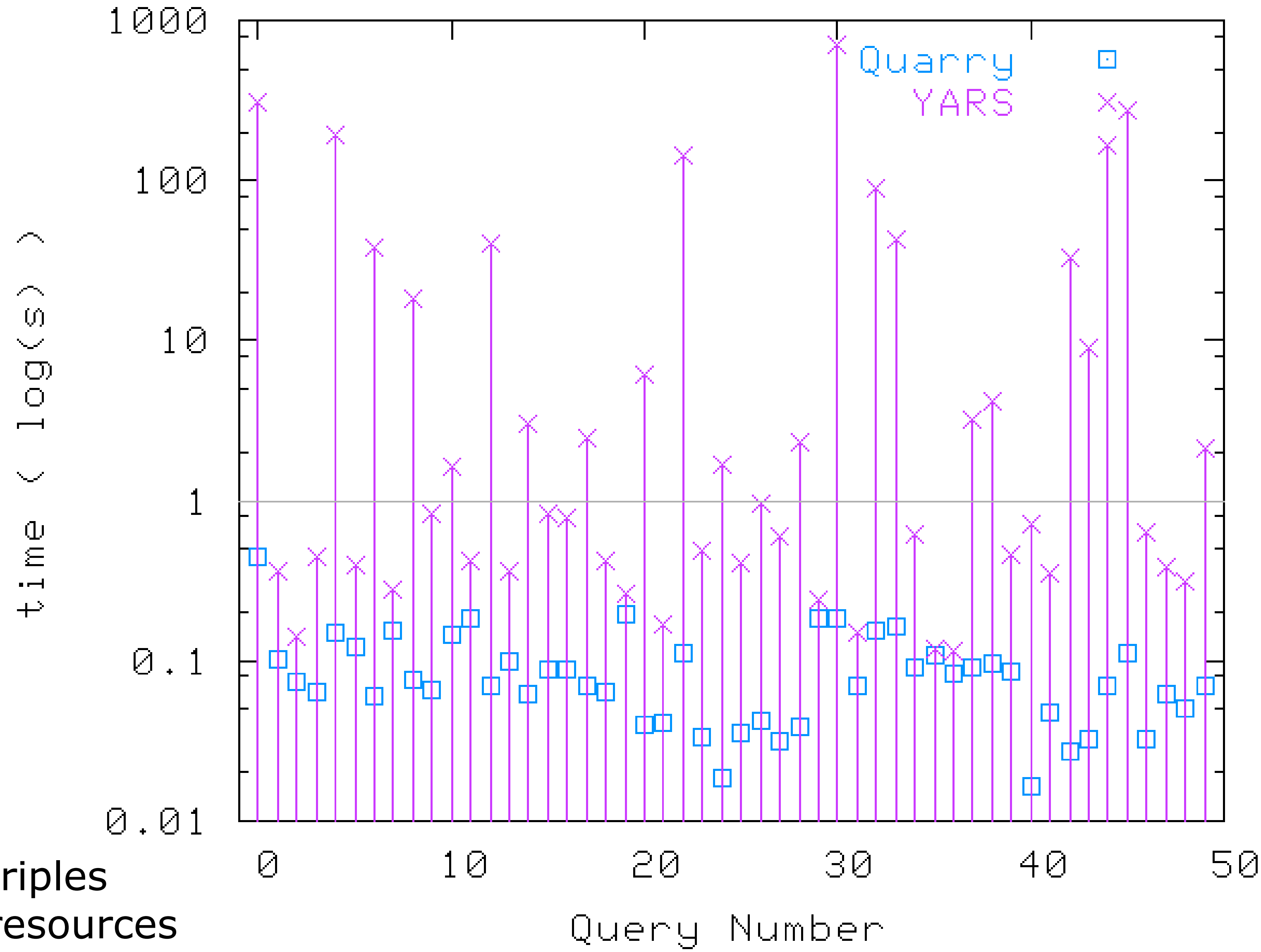
| rsrc | p1 | p3 |
|---|---|---|
| r1 | v(1,1) | v(1,3) |
| r2 | v(1,1) | v(1,3) |

## Experimental Results

- **Yet Another RDF Store (YARS)**
  - **9 B-Tree indexes in Berkeley DB**
    - rpv → _
    - pv → r
    - vr → p
    - etc.
  - **Authors report performance gain over Redland and Sesame**
    - ~3M triples, single term queries

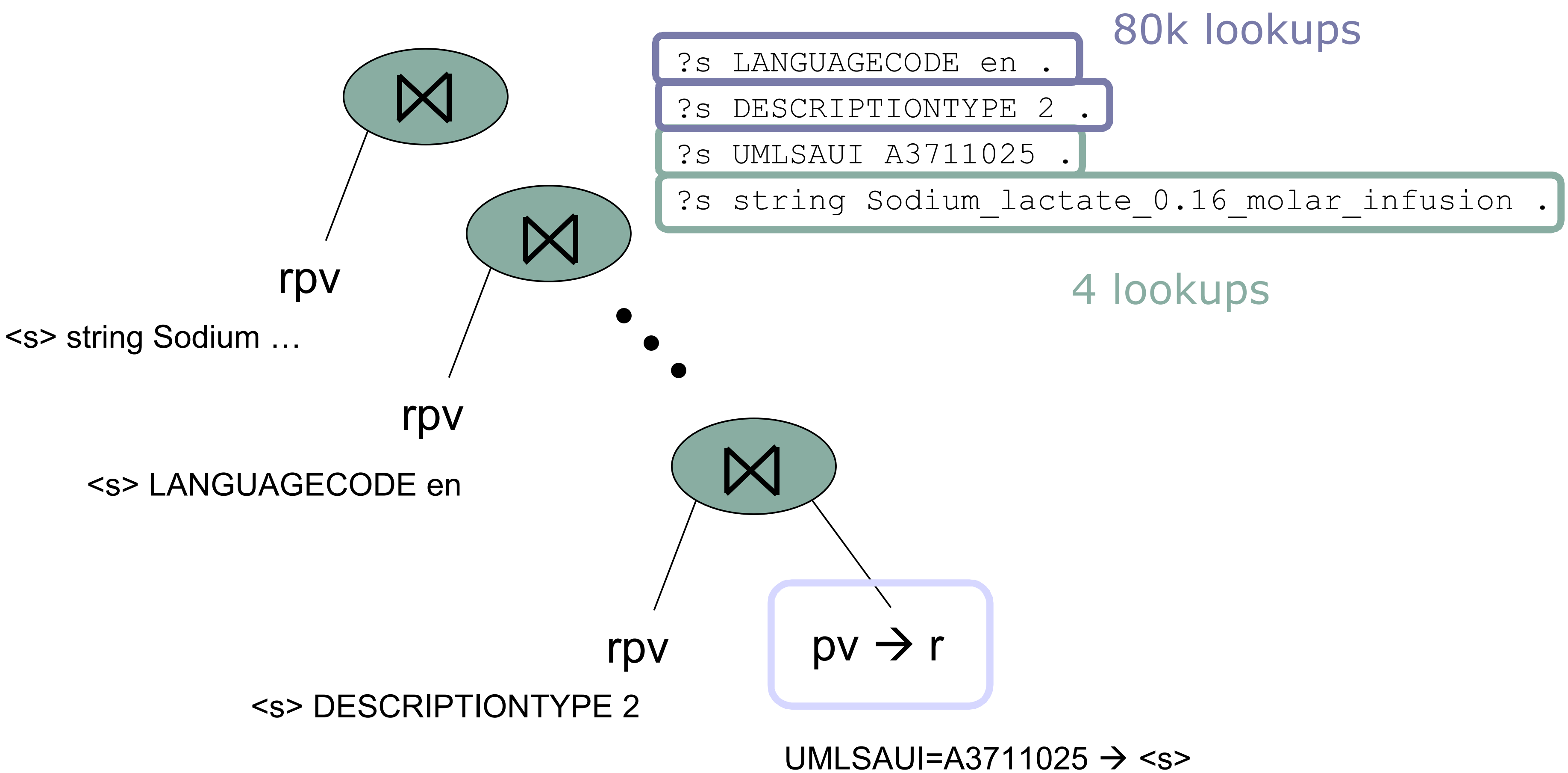
- Random multi-term conjunctive queries

```
?s <p0> <o0>
?s <p1> <o1>
:
?s <pn> <on>
```




3.6M triples
606k resources
149 signatures

# A Common YARS Query Plan



# YARS Plan Speed