Semantic Web: Knowledge Representation in Life Sciences

By

Kei Cheung

Yale Center for Medical Informatics

Genomics & Bioinformatics, February 18, 2009

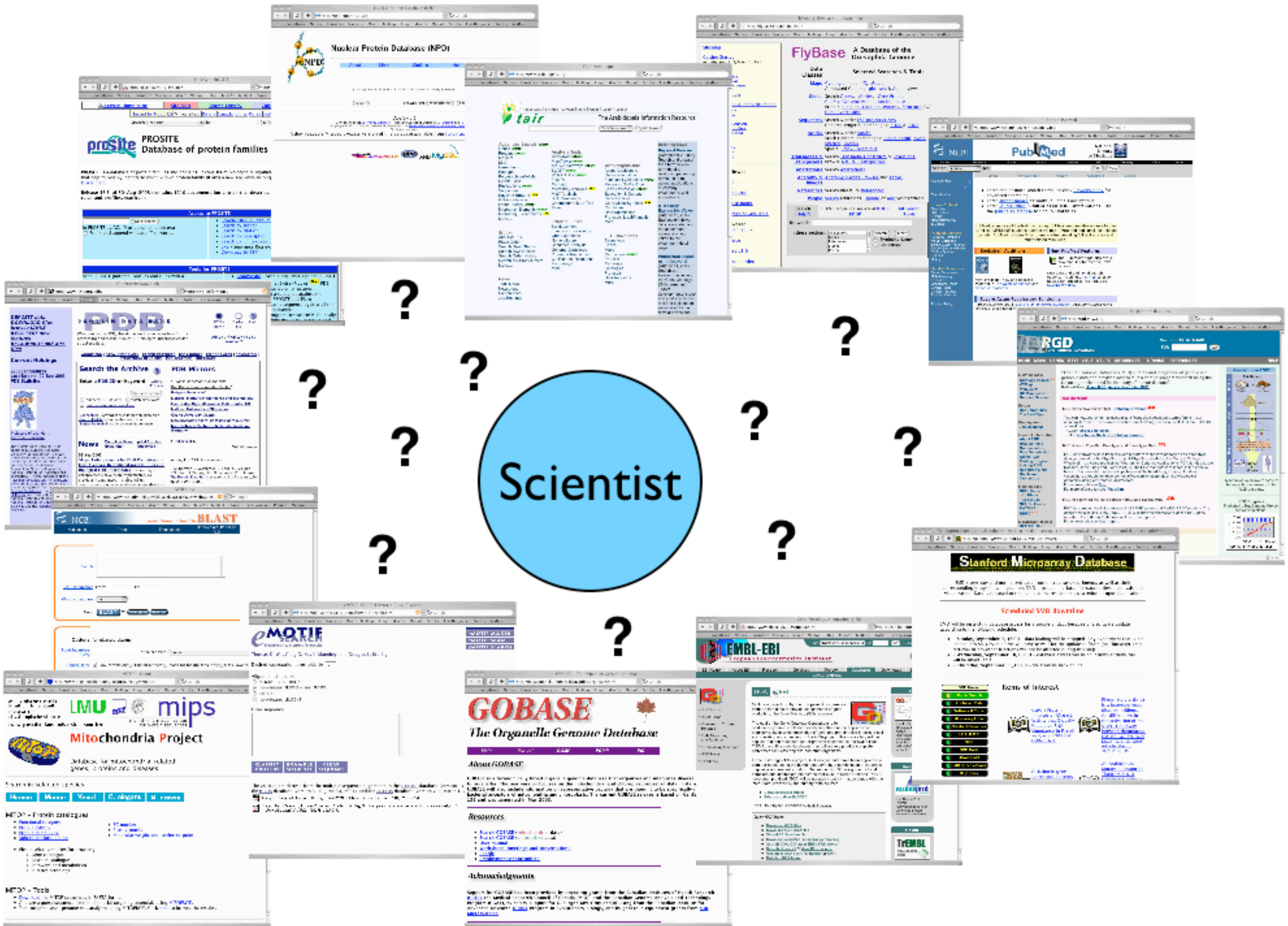
Introduction

- There has been an explosion of life science data (thanks for high-throughput genomics and proteomics technologies)
- There have been a growing number of life sciences databases including centralized repositories like GenBank, GEO, PRIDE, etc
- Data integration is an important problem in life sciences
- These databases are Web-accessible, but they not very machine-accessible

Web 1.0 vs. 2.0 vs. 3.0

- Web 1.0
  - Data display (HTML)
- Web 2.0
  - Data exchange (XML)
- Web 3.0
  - Data/knowledge modeling and integration (RDF/OWL)

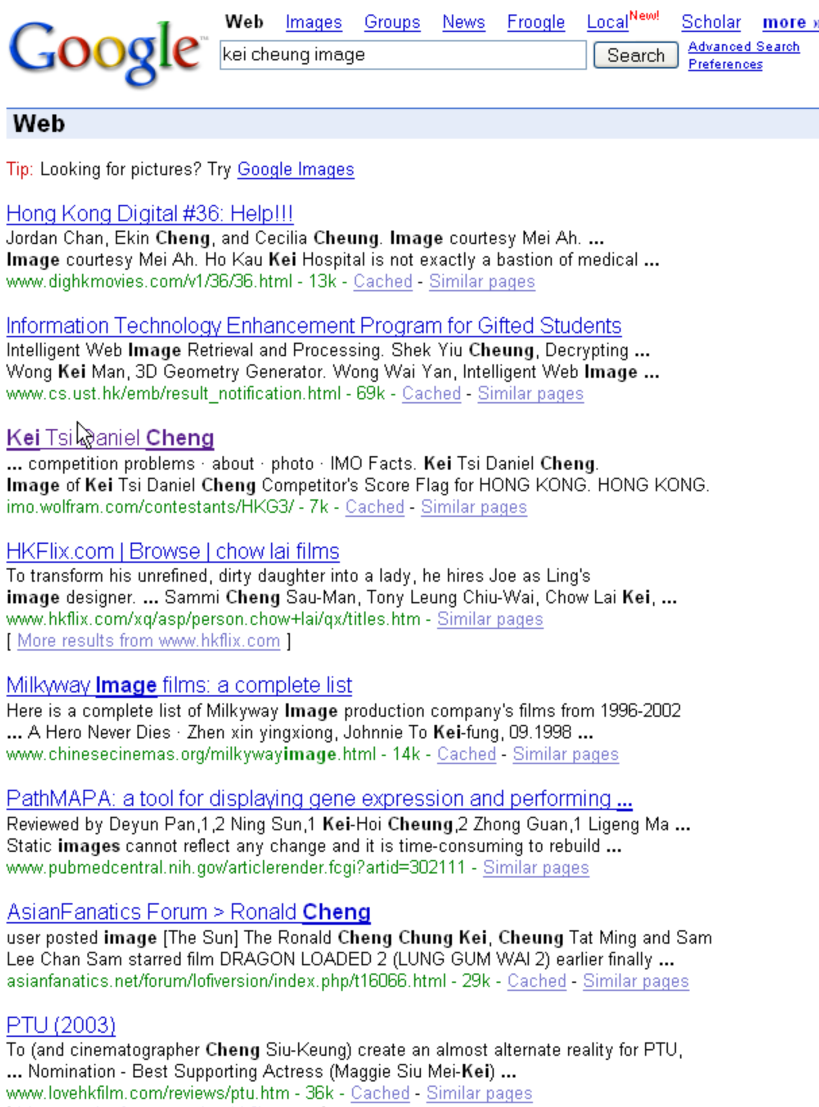
Problem with Web 1.0 (HTML)



Problem with Web 1.0 (cont'd)

- Lack of annotation
- Lack of links
- Lack of link semantics
- Lack of data semantics

Lack of Semantic Annotation

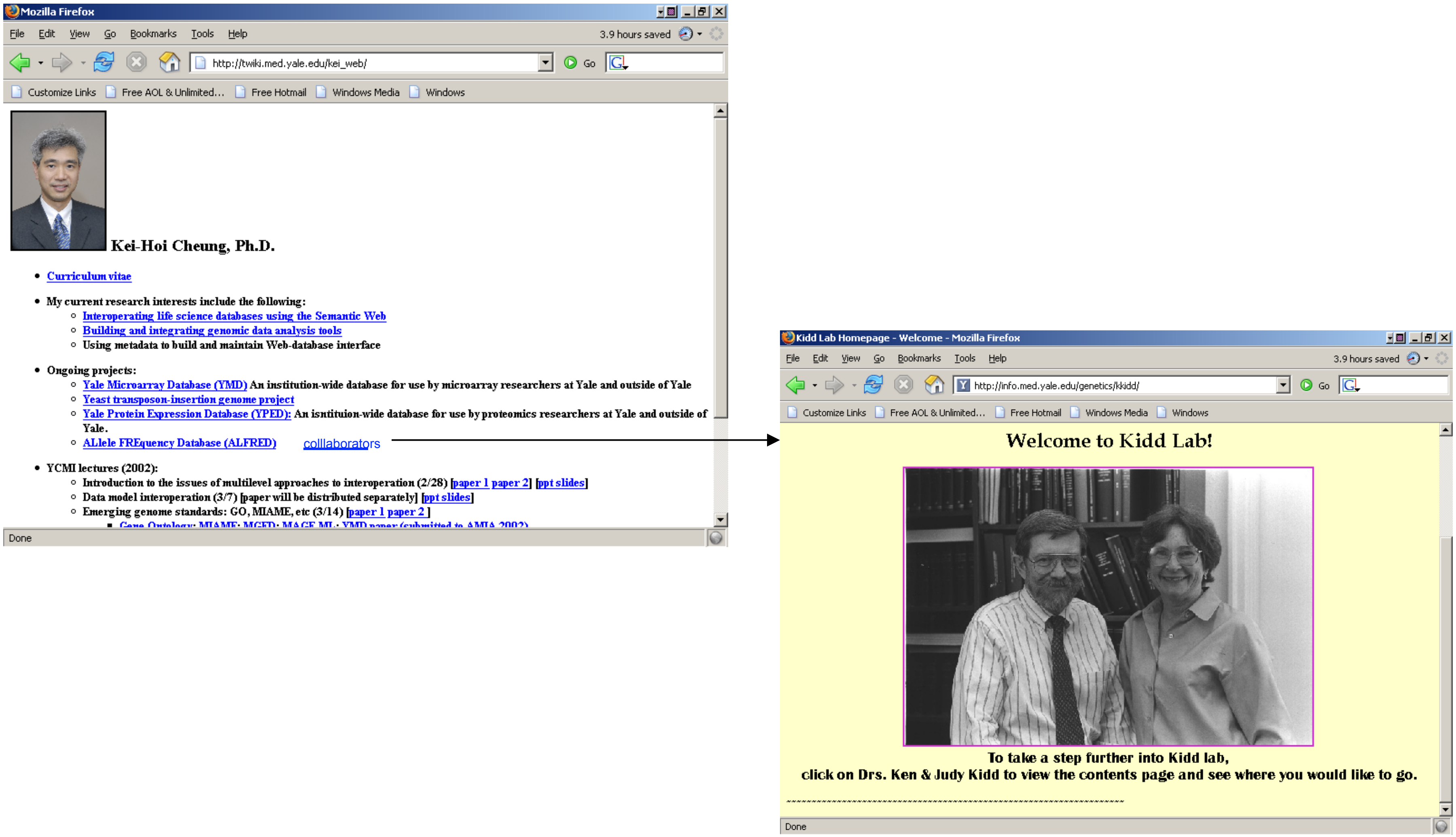


Kei Tsi Daniel Cheng
(this is not me!!)

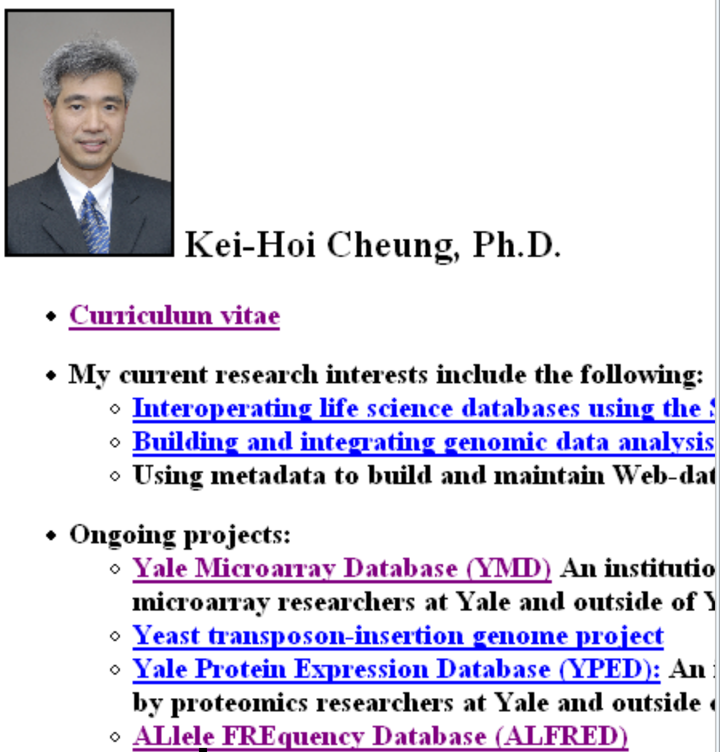
Kei Cheung
(15 years ago)

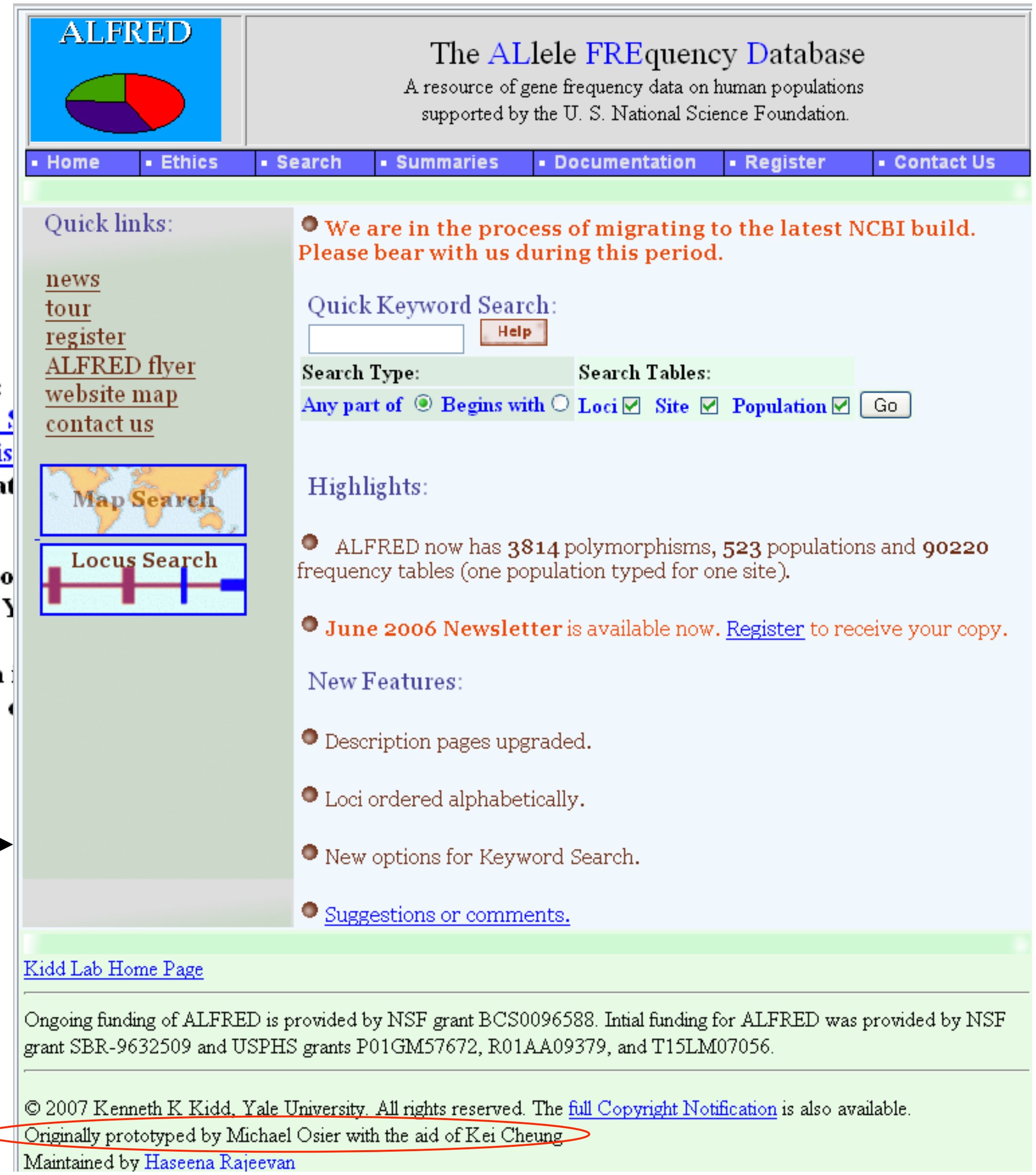
Kei Cheung
(2 months ago)

Lack of Links



Lack of Link Semantics

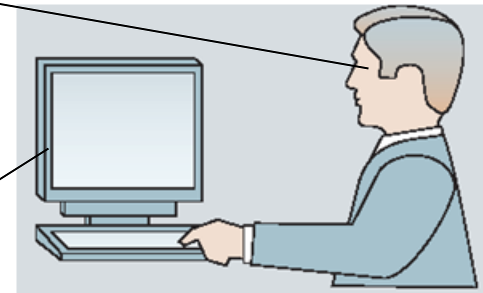




(?)
prototyped

Lack of Data Semantics

Type	Name	Synonym
Loci	Alcohol Dehydrogenase 1B (class I), beta polypeptide	ADH1B
Loci	Alcohol Dehydrogenase 1B (class I), beta polypeptide	ADH2
Loci	Solute carrier family 6 (neurotransmitter transporter, dopamine), member 3	ADHD
Loci	Alcohol Dehydrogenase 1C (class I), gamma polypeptide	ADH1C
Loci	Alcohol Dehydrogenase 1C (class I), gamma polypeptide	ADH3
Loci	Alcohol Dehydrogenase 7 (class IV), mu or sigma polypeptide	ADH-4
Loci	Alcohol Dehydrogenase 7 (class IV), mu or sigma polypeptide	ADH7



```
<html>
<body>
…
<table>
<tr>
<td><b>Type</b></td> <td><b>Name</b></td><td><b>Synonym</b></td>
</tr>
<tr>
<td>Loci</td> <td>Alcohol Dehydrogenase 1B (class I), beta polypeptide </td> <td>ADH1B </t>
</tr>
…
</table>
…
</body>
</html>
```

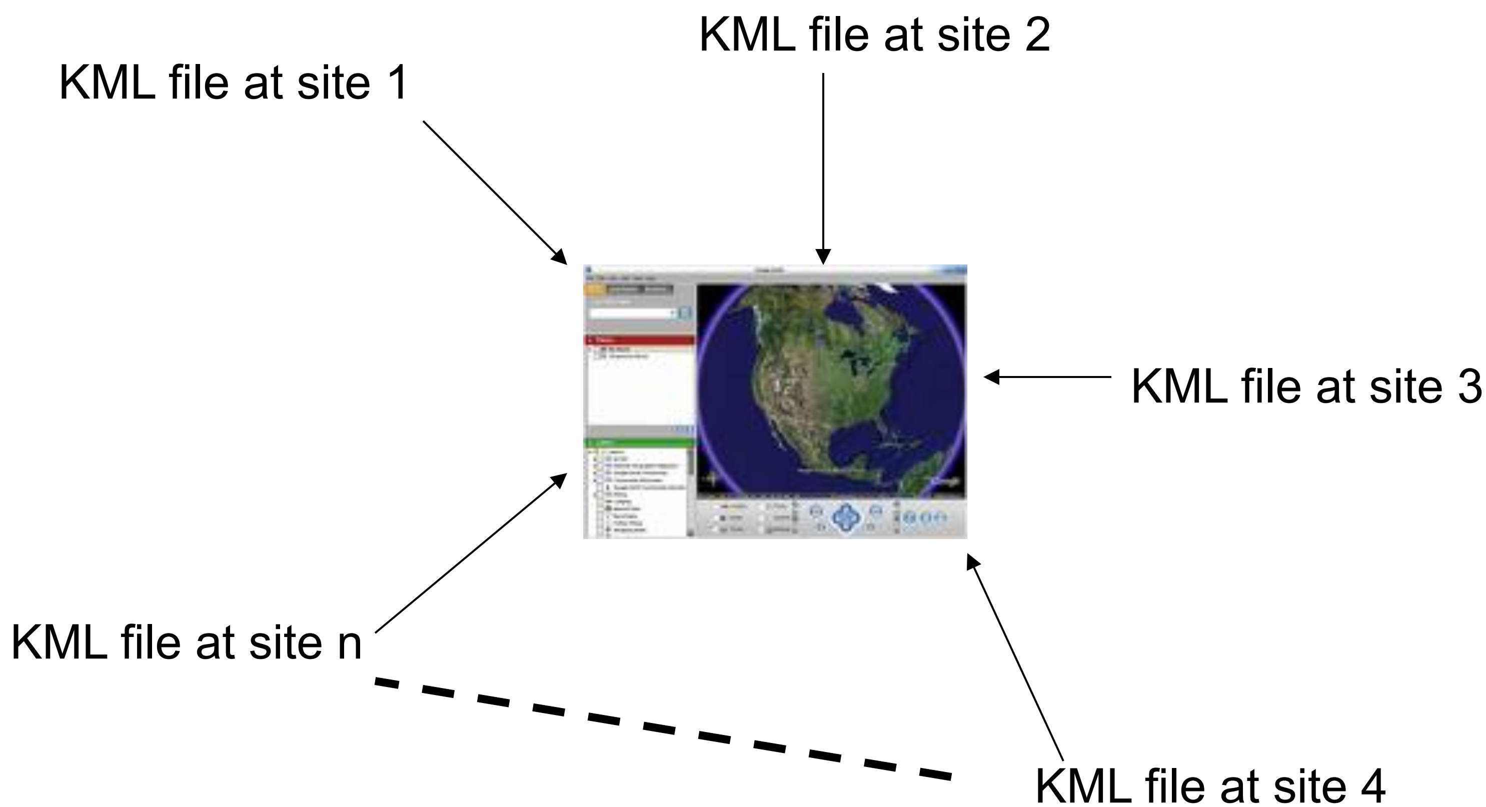
eXtensible Markup Language (XML)

- XML is designed to represent and deliver structured content over the web
- It is self descriptive by wrapping information with user-defined tags

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

- Some needs to write a program to process XML documents
- XML is a W3C Recommendation

Web 2.0 Mashup (e.g., Google Earth)



Geo-Mashup: Google Earth (tracking H5N1 virus over time)

Bio-XML

- AGAVE
- BSML
- AGML
- HUP-ML
- MAGE-ML
- SBML
- CellML
- Other …

Semantic Web

- The **Semantic Web** provides a common machine-readable framework that allows **data** to be shared and reused across application, enterprise, and community boundaries.
  - The Semantic Web is a web of data.
- The Semantic Web is about two things.
  - It is about common formats for integration and combination of data drawn from diverse sources
  - It is also about language for recording how the data relates to real world objects.

Semantic Web (cont'd)

- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- While RDFS and OWL are layered on top of RDF, they offer support for inference and axiom, making Semantic Web capable of supporting knowledge representation

RDF

- The foundation semantic web technology is the resource-description framework (RDF).
- RDF is a system to describe resources.
- RDF has a very simple yet elegant data model that can be summed up in one sentence: everything is a resource that connects with other resources via properties.
- A resource is anything that is identifiable by a uniform resource identifier (URI)

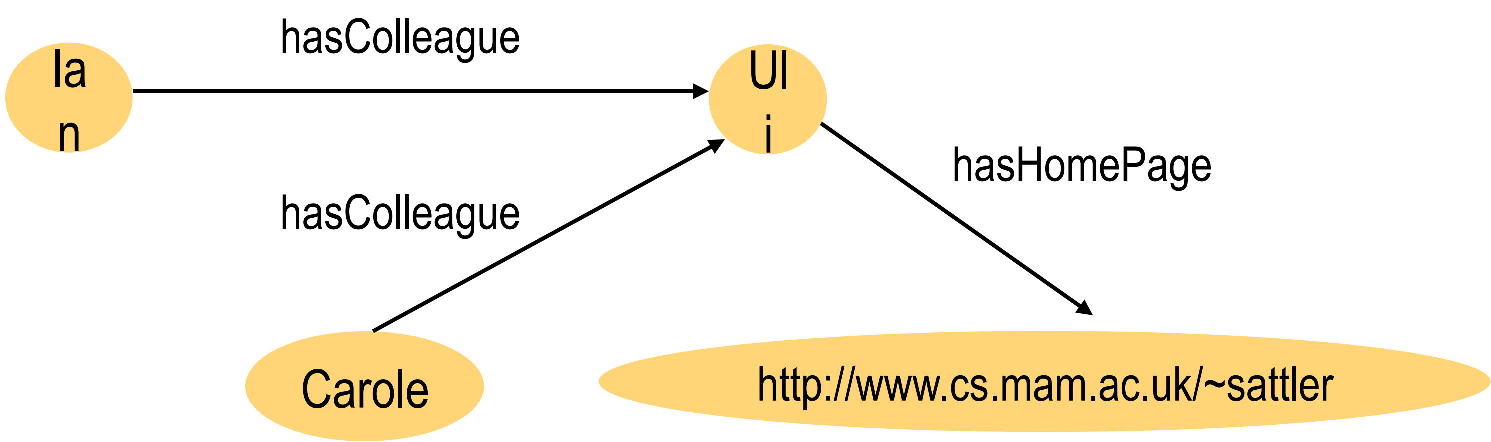
Uniform Resource Identifiers (URIs)

- "The generic set of all names/addresses that are short strings that refer to resources"
- URLs (Uniform Resource Locators) are a particular type of URI, used for resources that can be accessed on the WWW (e.g., web pages)
- In RDF, URIs typically look like “normal” URLs, often with fragment identifiers to point at specific parts of a document:
  - http://www.somedomain.com/some/path/to/file#fragmentID

RDF (cont' d)

- The basic information unit in RDF is an RDF statement in the form of
  - (subject, property, object)
- Each RDF statement can be modeled as a graph comprising two nodes connected by a directed arc

- A set of such graphs can jointly form a directed labeled graph (DLG) that can in theory model most domain knowledge.



RDF/XML Syntax

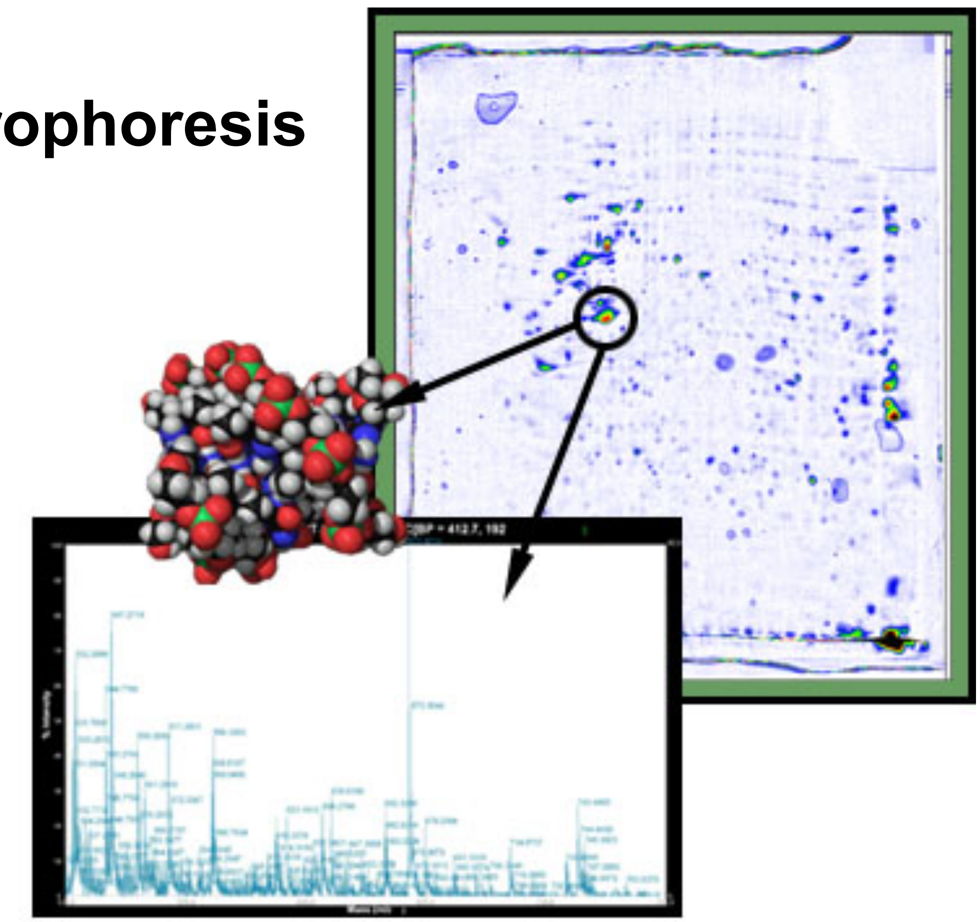
- RDF has an XML syntax that has a specific meaning:
- Every `Description` element describes a resource
- Every attribute or nested element inside a `Description` is a `property` of that Resource
- We can refer to resources by using URIs

```
<Description about="some.uri/person/ian_horrocks">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
<Description about="some.uri/person/uli_sattler">
   <hasHomePage>http://www.cs.mam.ac.uk/~sattler</hasHomePage>
</Description>
<Description about="some.uri/person/carole_goble">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
```

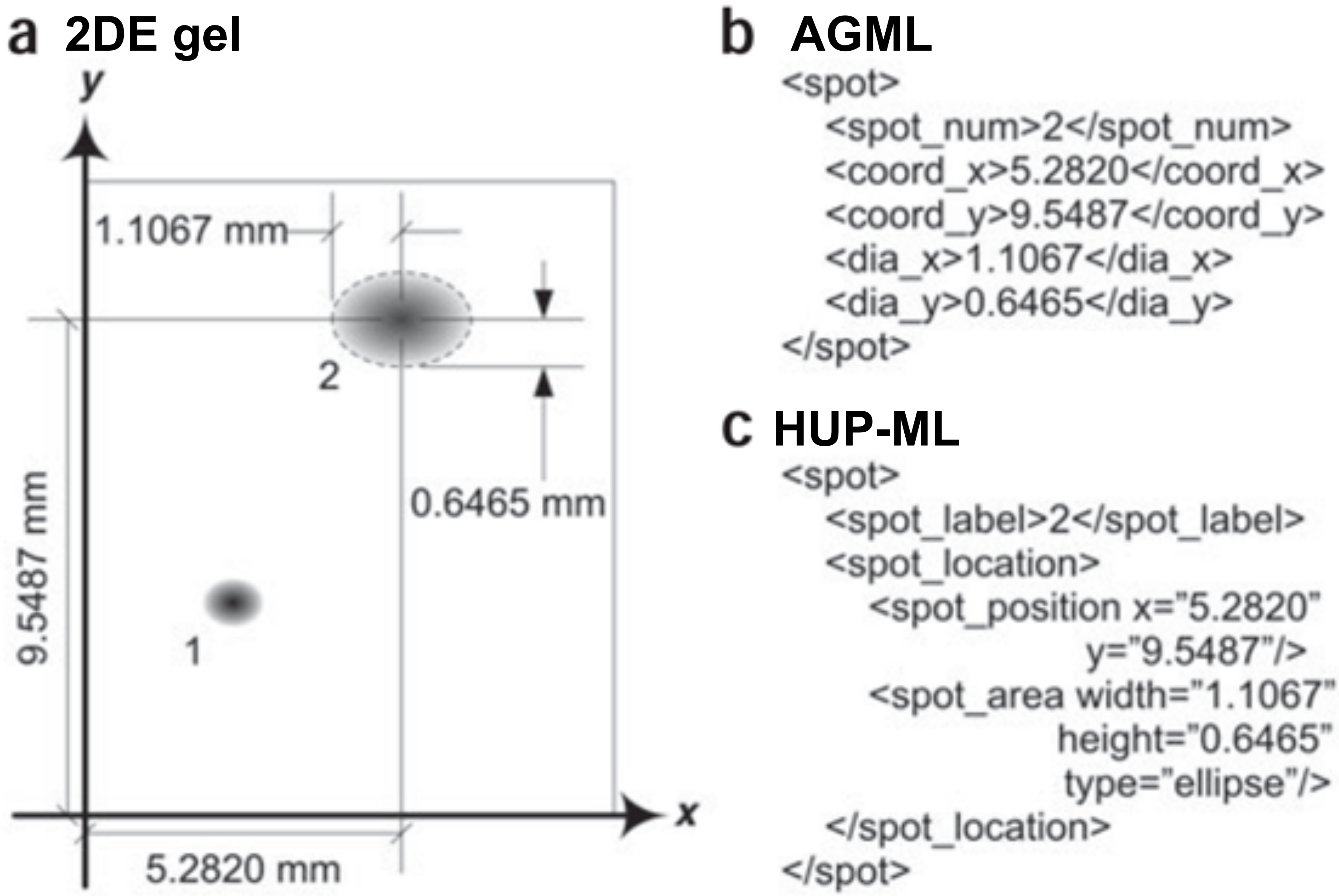
From XML to RDF

(Wang et al. (2005) Nat Biotechnol. 23(9):1099-103)

2D Gel Electrophoresis



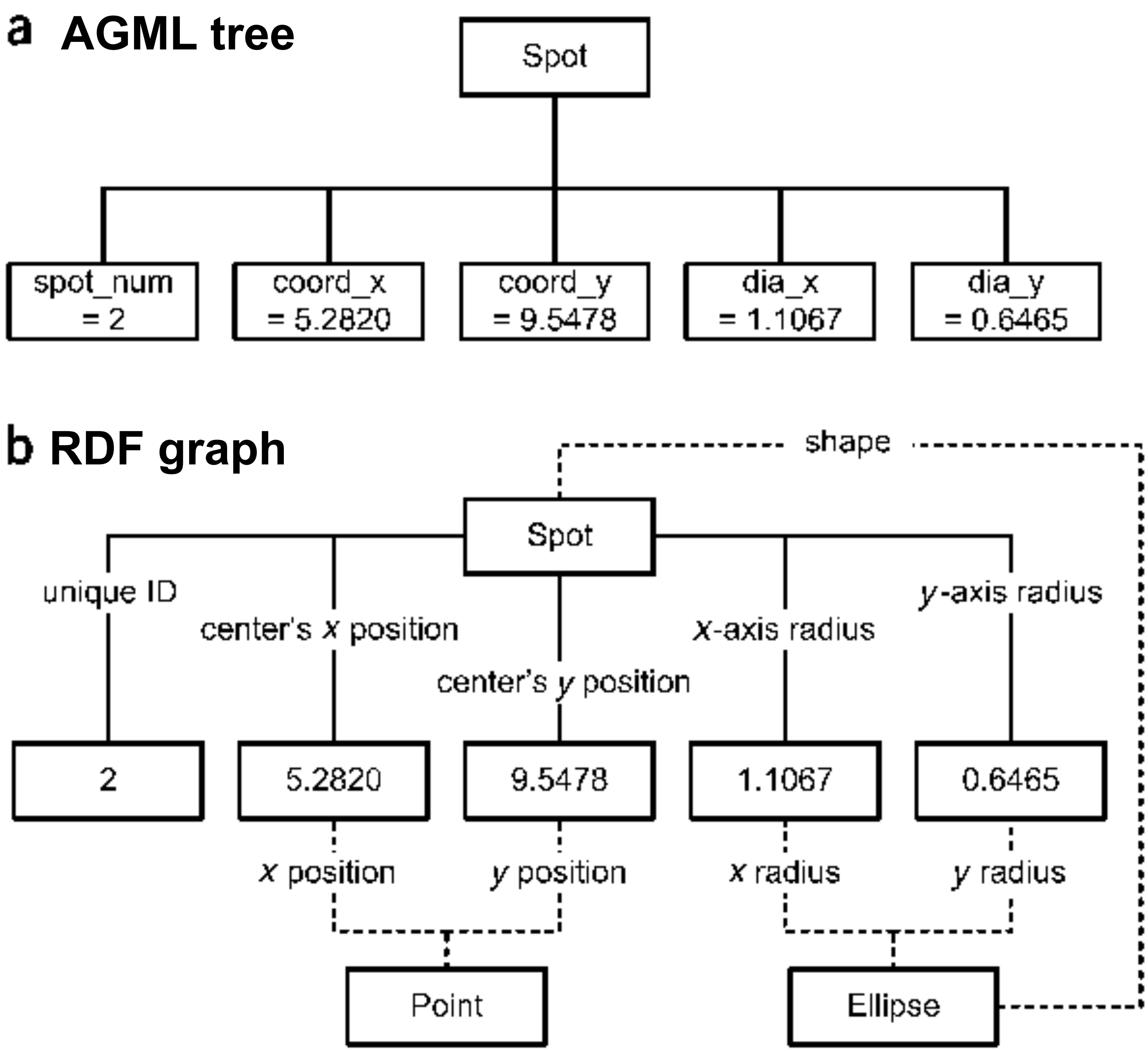
XML Representations of 2DE gel



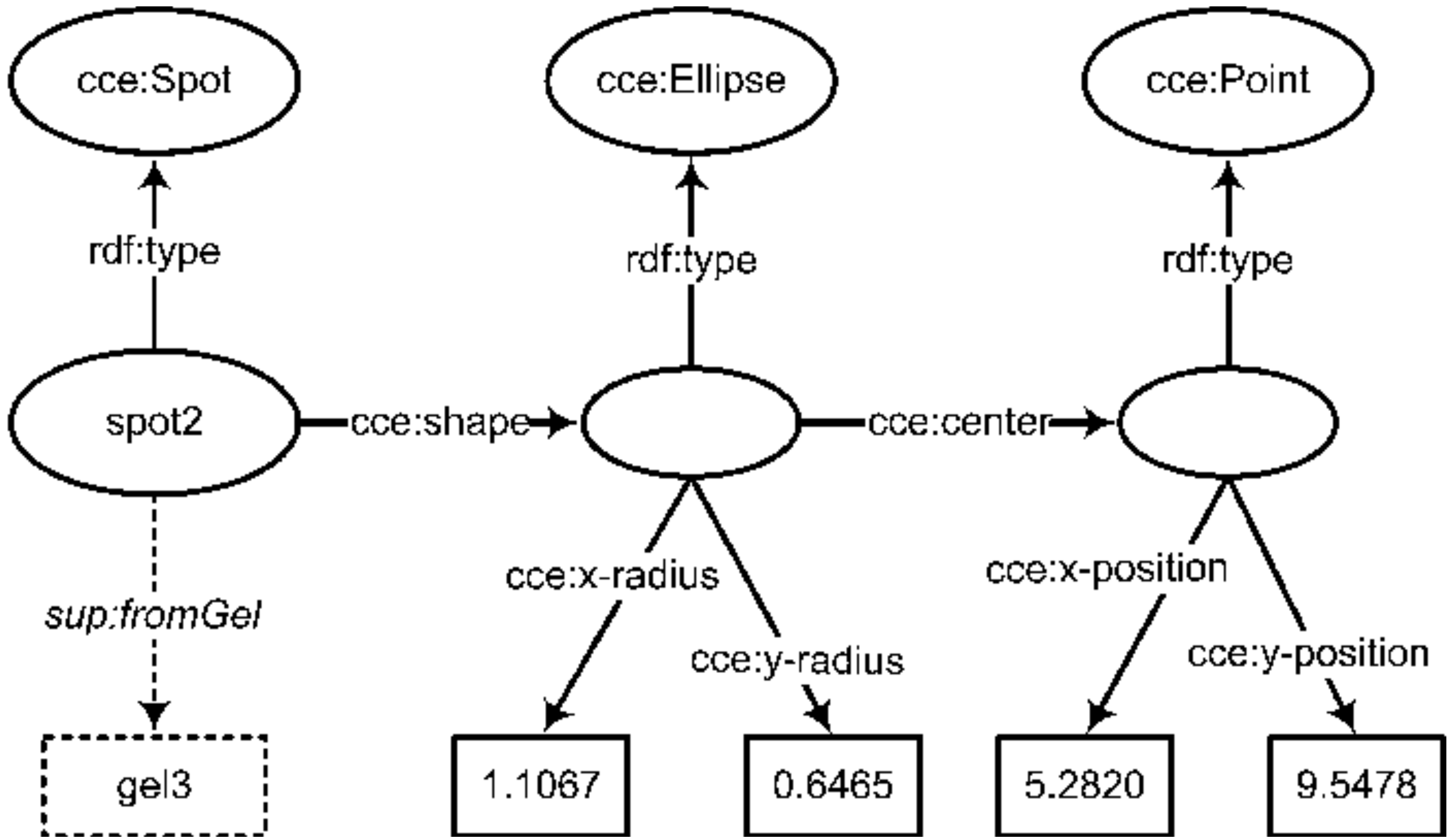
Problems with XML

- Limited expressiveness of the XML language.
- XML is designed as a language for message encoding
- XML is only self-descriptive about the following structural relationships:
  - containment, adjacency, co-occurrence, attribute and opaque reference.
  - All these relationships are useful for serialization, but are not optimal for modeling objects of a problem domain
  - For example, the relationship between the <spot> and <coord_*> of AGML tags is no different from that between <spot> and <dia_*>.
  - A computer algorithm must treat them differently to develop meaningful applications. To calculate the distance between two <spot>s, an algorithm shall use the value of <coord_*>, but to calculate the area of each <spot>, it shall retrieve the value of <dia_*> instead

XML vs. RDF


a AGML tree
Spot
spot_num = 2
coord_x = 5.2820
coord_y = 9.5478
dia_x = 1.1067
dia_y = 0.6465
b RDF graph
shape
Spot
unique ID
center's x position
center's y position
x-axis radius
y-axis radius
2
5.2820
9.5478
1.1067
0.6465
x position
y position
x radius
y radius
Point
Ellipse

An RDF Model for a spot on a 2DE gel



Characteristics of RDF

- The DLG structure offered by RDF makes it extensible and evolvable. Adding nodes and edges to a DLG doesn't change the structure of any existing subgraph.
- RDF has an open-world assumption in that allows anyone to make statements about any resource
- RDF is monotonic in that new statements neither change nor negate the validity of previous assertions, making it particularly suitable in an academic environment, in which consensus and disagreement about the same resources have a useful coexistence that needs to be formally recorded.
- All RDF terms share a global naming scheme in URI, making distributed data and ontologies possible
- The combined effect of global naming, universal data structure and open-world assumption is that resources exist independently but can be readily linked with little precoordination.

RDF can be helpful to omic approaches to biology

- The decoupled nature of RDF makes it a natural choice for defining an omic standard.
- The essence of omic science resides in its "holistic" description of the subject of interest
- RDF makes it possible to connect all omic-specific data as a whole without necessarily turning them into a "whole".

Ontology: Origins and History

Ontology in Philosophy

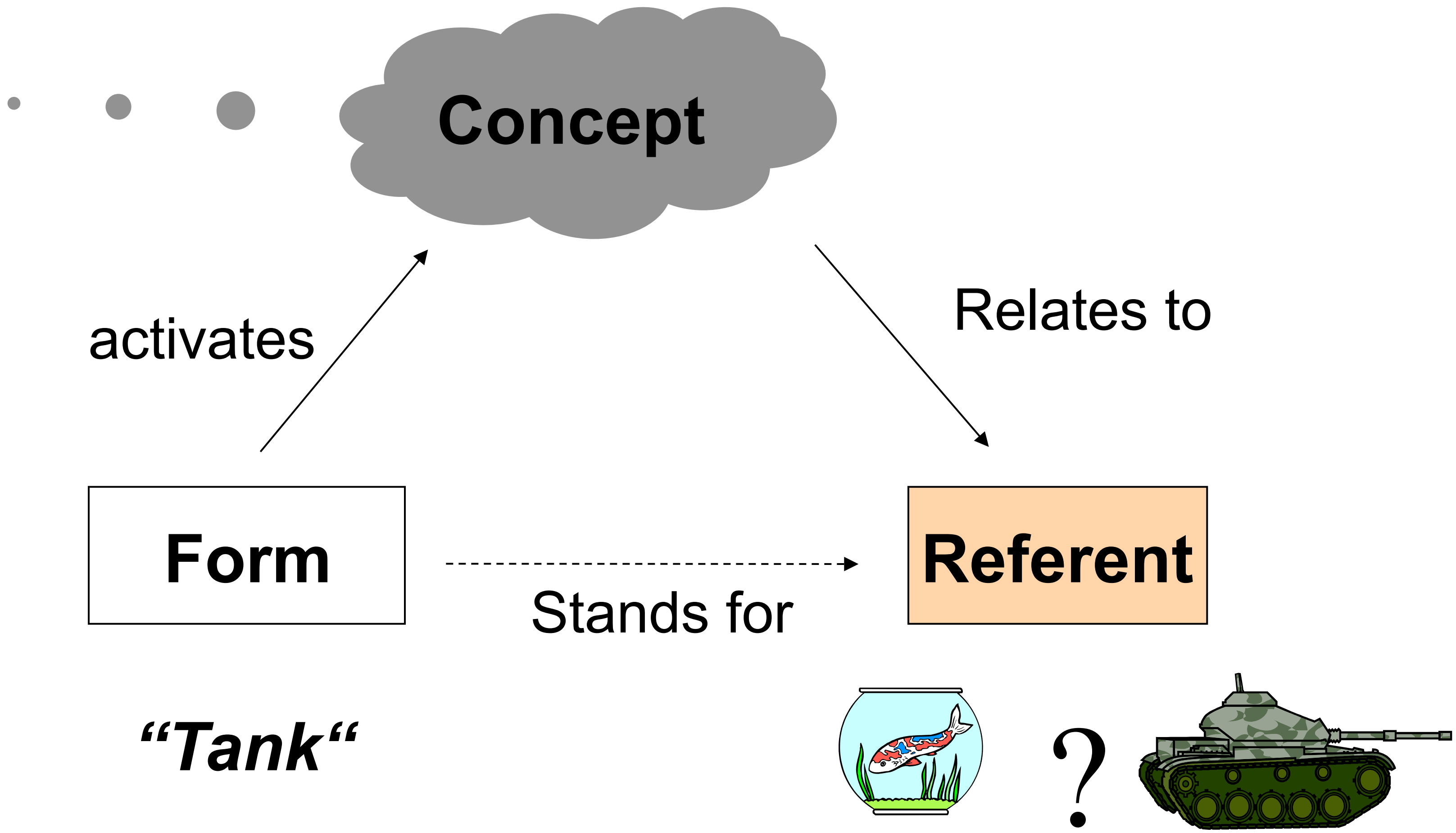
a philosophical discipline—a branch of philosophy that
deals with the nature and the organisation of reality

- Science of Being (Aristotle, Metaphysics)
- Tries to answer the questions:

  What characterizes being?

  Eventually, what is being?

Ontology in Linguistics



Ontology in Computer Science

- An ontology is an engineering artifact:
  - It is constituted by a specific vocabulary used to describe a certain reality, plus
  - a set of explicit assumptions regarding the intended meaning of the vocabulary.
- Thus, an ontology describes a formal specification of a certain domain:
  - Shared understanding of a domain of interest
  - Formal and machine manipulable model of a domain of interest

Structure of an Ontology

Ontologies typically have two distinct components:

- Names for important concepts in the domain
  - Elephant is a concept whose members are a kind of animal
  - Herbivore is a concept whose members are exactly those animals who eat only plants or parts of plants
  - Adult_Elephant is a concept whose members are exactly those elephants whose age is greater than 20 years
- Background knowledge/constraints on the domain
  - Adult_Elephants weigh at least 2,000 kg
  - All Elephants are either African_Elephants or Indian_Elephants
  - No individual can be both a Herbivore and a Carnivore

A Semantic Web — First Steps

Make web resources more accessible to automated processes

- Extend existing rendering markup with semantic markup
  - Metadata annotations that describe content/function of web accessible resources
- Use Ontologies to provide vocabulary for annotations
  - "Formal specification" is accessible to machines

- A prerequisite is a standard web ontology language
  - Need to agree common syntax before we can share semantics

Ontology Design and Deployment

- Given key role of ontologies in the Semantic Web, it will be essential to provide tools and services to help users:
  - Design and maintain high quality ontologies, e.g.:
    - Meaningful — all named classes can have instances
    - Correct — captured intuitions of domain experts
    - Minimally redundant — no unintended synonyms
    - Richly axiomatised — (sufficiently) detailed descriptions
  - Store (large numbers) of instances of ontology classes, e.g.:
    - Annotations from web pages
  - Answer queries over ontology classes and instances, e.g.:
    - Find more general/specific classes
    - Retrieve annotations/pages matching a given description
  - Integrate and align multiple ontologies

Ontology Languages for the Semantic Web

Ontology Languages

- Wide variety of languages for "Explicit Specification"
  - Graphical notations
    - RDF/RDFS
  - Logic based
    - Description Logics (e.g., OIL, DAML+OIL, OWL)
    - Rules (e.g., RuleML, LP/Prolog)
    - First Order Logic (e.g., KIF)
    - Conceptual graphs
    - (Syntactically) higher order logics (e.g., LBase)
    - Non-classical logics (e.g., Flogic, modalities)
  - Probabilistic/fuzzy
- Degree of formality varies widely
  - Increased formality makes languages more amenable to machine processing (e.g., automated reasoning)

RDF Schema (RDFS)

- RDF is graphical formalism (+ XML syntax + semantics)
  - for representing metadata
  - for describing the semantics of information in a machine- accessible way
- RDFS extends RDF with "schema vocabulary", e.g.:
  - Class, Property
  - type, subClassOf, subPropertyOf
  - range, domain

RDFS (cont'd)

- RDF gives a formalism for meta data annotation, and a way to write it down in XML, but it does not give any special meaning to vocabulary such as subClassOf or type
- RDF Schema allows you to define vocabulary terms and the relations between those terms
  - it gives "extra meaning" to particular RDF predicates and resources
  - this "extra meaning", or semantics, specifies how a term should be interpreted

RDFS Examples

- Example RDF Schema terms:
  - Class
  - Property
  - type
  - subClassOf
  - range
  - domain
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

```
<Person,type,Class>
<hasColleague,type,Property>
<Professor,subClassOf,Person>
<Carole,type,Professor>
<hasColleague,range,Person>
<hasColleague,domain,Person>
```

RDF/RDFS "Liberality"

- No distinction between classes and instances (individuals)

```
<Species,type,Class>
<Lion,type,Species>
<Leo,type,Lion>
```

- Properties can themselves have properties

```
<hasDaughter,subPropertyOf,hasChild>
<hasDaughter,type,familyProperty>
```

- No distinction between language constructors and ontology vocabulary, so constructors can be applied to themselves/each other

```
<type,range,Class>
<Property,type,Class>
<type,subPropertyOf,subClassOf>
```

Problems with RDFS

- RDFS too weak to describe resources in sufficient detail
  - No localized range and domain constraints
    - Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can’t say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No transitive, inverse or symmetrical properties
    - Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
  - …
- Difficult to provide reasoning support

Web Ontology Language Requirements

Desirable features identified for Web Ontology Language:

- Extends existing Web standards
  - Such as XML, RDF, RDFS
- Easy to understand and use
  - Should be based on familiar KR idioms
- Formally specified
- Of "adequate" expressive power
- Possible to provide automated reasoning support

From RDF to OWL

- Two languages developed to satisfy above requirements
  - OIL: developed by group of (largely) European researchers (several from EU OntoKnowledge project)
  - DAML-ONT: developed by group of (largely) US researchers (in DARPA DAML programme)
- Efforts merged to produce DAML+OIL
  - Development was carried out by "Joint EU/US Committee on Agent Markup Languages"
  - Extends ("DL subset" of) RDF
- DAML+OIL submitted to W3C as basis for standardisation
  - Web-Ontology (WebOnt) Working Group formed
  - WebOnt group developed OWL language based on DAML+OIL
  - OWL language now a W3C Candidate Recommendation
  - Will soon become Proposed Recommendation

OWL Language

- Three species of OWL
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to FOL fragment (¼ DAML+OIL)
  - OWL Lite is "easier to implement" subset of OWL DL
- Semantic layering
  - OWL DL ¼ OWL full within DL fragment
  - DL semantics officially definitive
- OWL DL based on SHIQ Description Logic
  - In fact it is equivalent to SHOIN$(\mathbf{D}_n)$ DL
- OWL DL Benefits from many years of DL research
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)

OWL Class Constructors

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1 \wedge \ldots \wedge C_n$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1 \vee \ldots \vee C_n$
complementOf	$\neg C$	¬Male	$\neg C$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x_1 \vee \ldots \vee x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leqslant nP$	⩽1hasChild	$[P]_{n+1}$
minCardinality	$\geqslant nP$	⩾2hasChild	$\langle P \rangle_n$

- XMLS datatypes as well as classes in 8P.C and 9P.C

OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+ \sqsubseteq$ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant 1$hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant 1$hasSSN$^-$

Data/ontologies available in RDF/ OWL format

- UniProt
- Gene Ontology
- NCI Metathesaurus
- MGED Ontology
- Sequence Ontology
- Protein Ontology
- Many more …

Semantic Web/Ontology Resources

- Semantic Web for Health Care and Life Sciences Interest Group
  – http://www.w3.org/2001/sw/hcls/
- National Center for Biomedical Ontologies
  – http://bioontology.org/
- Open Biomedical Ontologies (OBO) Foundry
  – http://www.obofoundry.org/

Enabling Technologies

- Ontology viewers/editors (e.g., Protégé)
- SPARQL
- OWL reasoners (e.g., Pellet, RacerPro, FaCT++)
- Triplestores (Sesame, Virtuoso, Oracle, Allegro Graph) – SPARQL Endpoint

Related Technologies

- RDF attribute (RDFa)
- GRDDL
- Semantic Wiki

The End