Building the Semantic Web

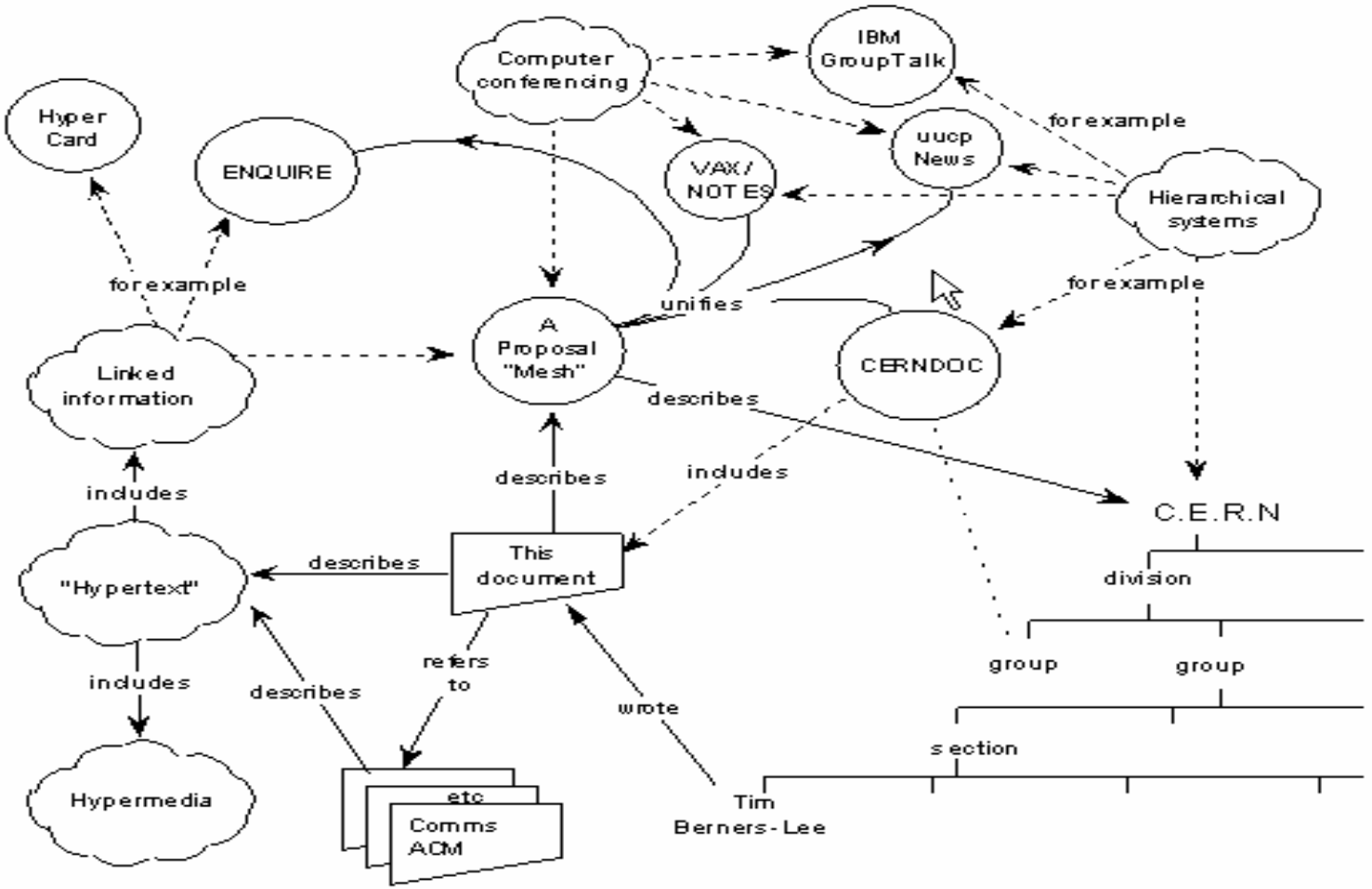
CS 431 – March 27, 2006

Carl Lagoze – Cornell University

Acknowledgements for various slides and ideas

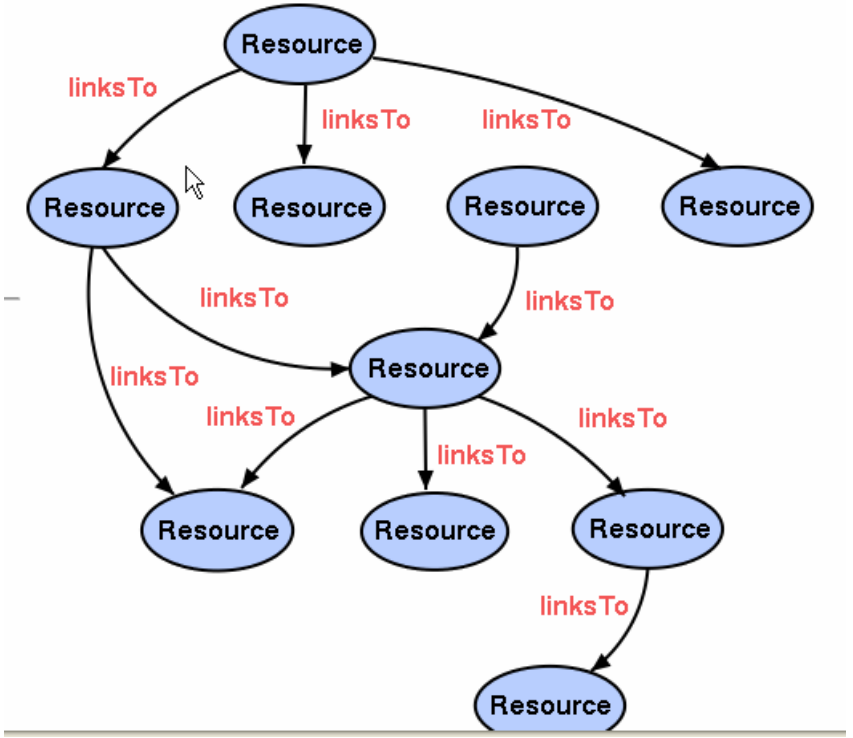
- Ian Horrocks (Manchester U.K.)
- Eric Miller (W3C)
- Dieter Fensel (Berlin)
- Volker Haarslev (Montreal)

Original Web Vision
Web of relationships amongst named objects



Current Web ("Syntactic" Web)

- Untyped resources named by URLs
- Untyped relationships (href with anchor text)
- User oriented – document rendering
- Machines must infer information

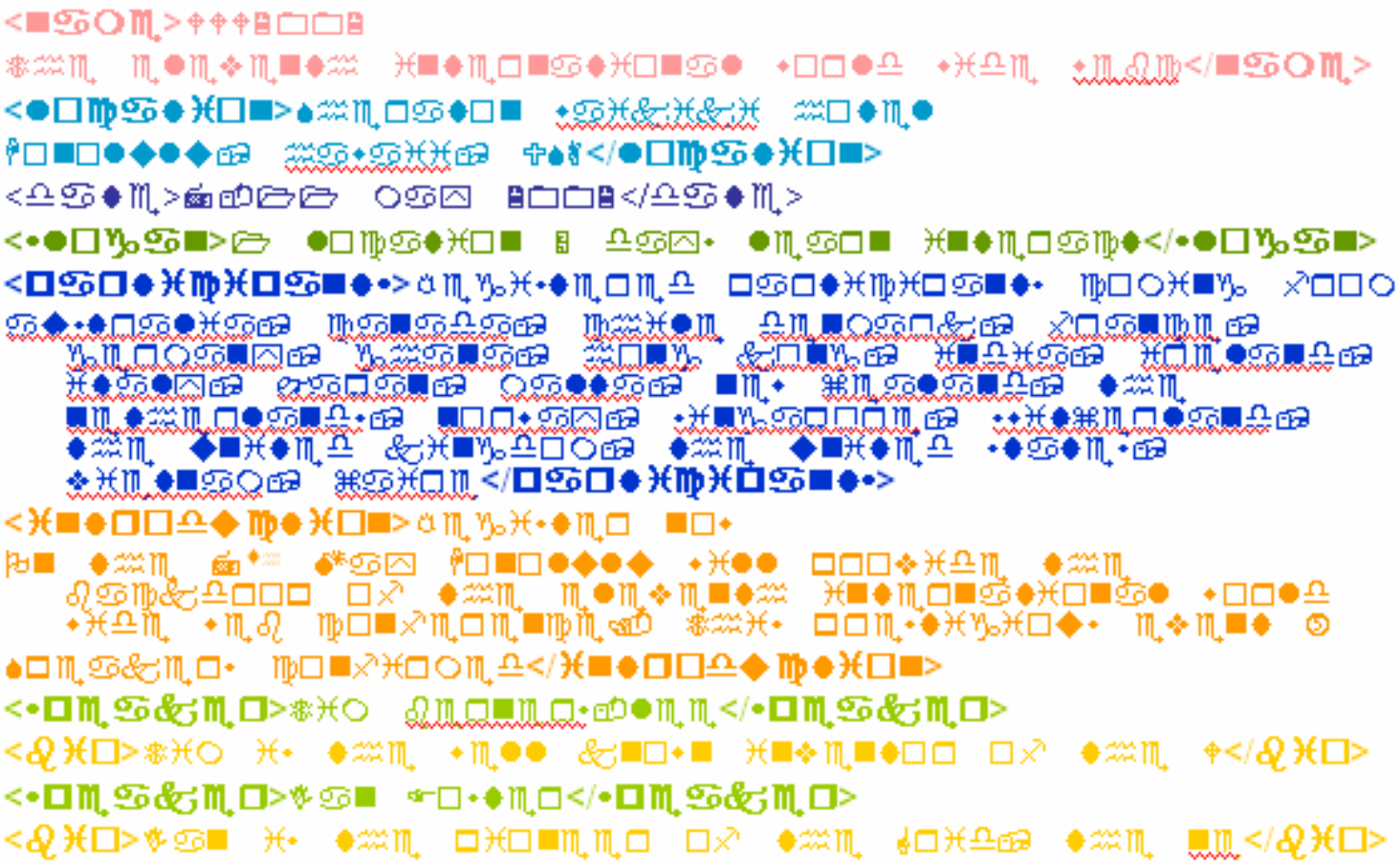


The Information in a Web Page

- Markup connotes semantics (bold, colors, font…)
- Humans interpret semantics
- Rendering semantics is not clear or available to machines

Why is XML not enough?

but from the machines point of view…

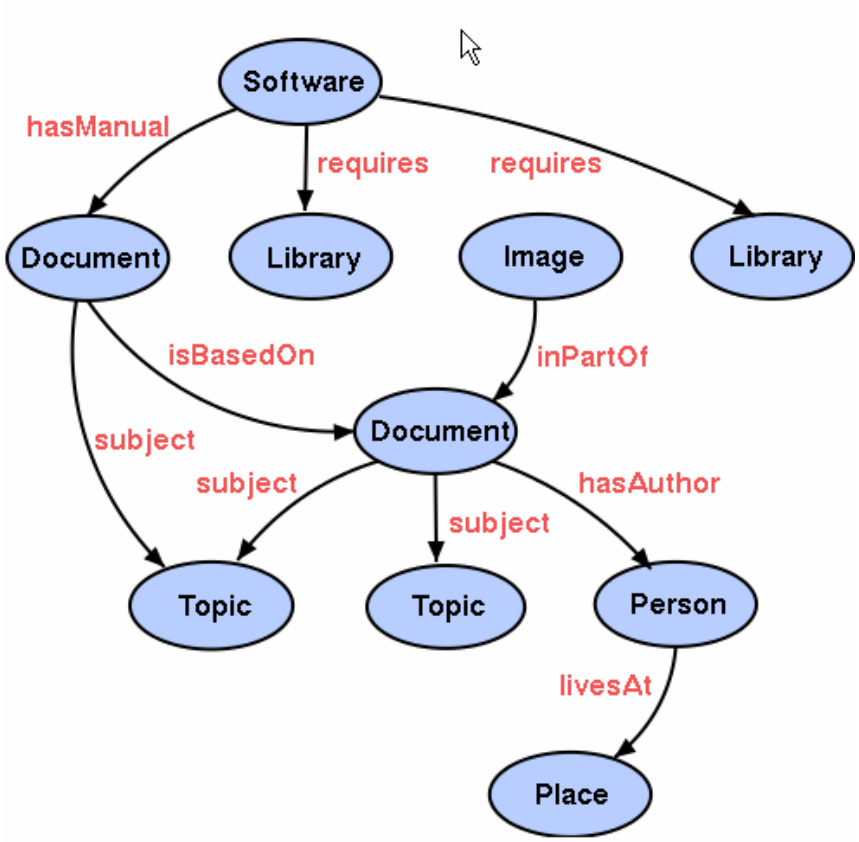


XMLs helps but not much…

- Descendent of DTD's
  - Mostly a structuring language
  - Doesn't express "meaning" of structure
- Problems with knowledge representation
  - Poor expression of concepts, relationships, and subsumption
  - No basis in formal logic, limited if any basis for reasoning
  - So, can't do:
    - Ford is a "kind of" car
    - VW is a "kind of" car
    - Joe "has a" Ford
    - Sue "has a" VW
    - Joe and Sue both have cars

Semantic Web

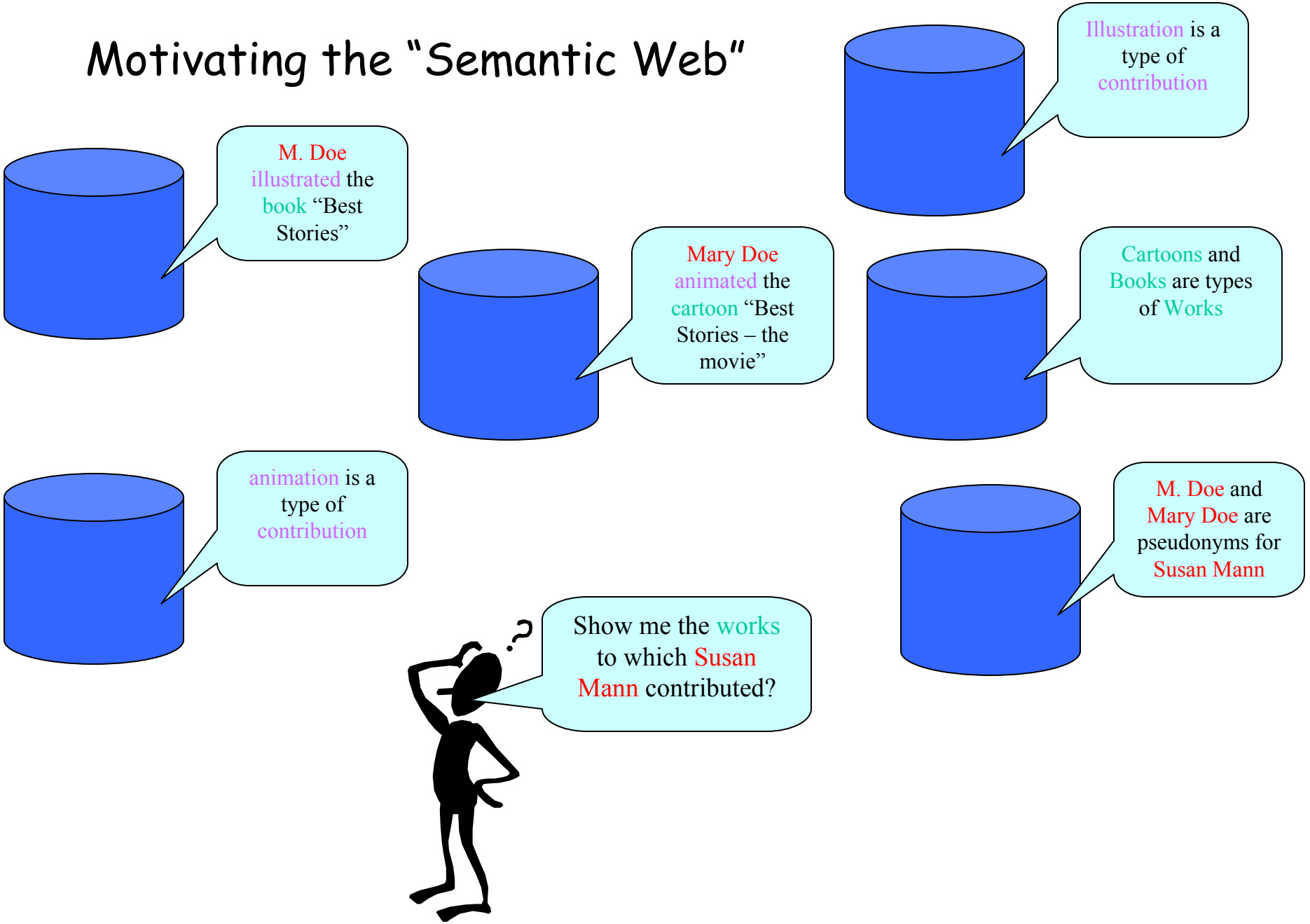
- Resources typed, types defined by URIs
- Relationships typed, types defined by URIs
- Types are structured and are first-class
- Machines can inference



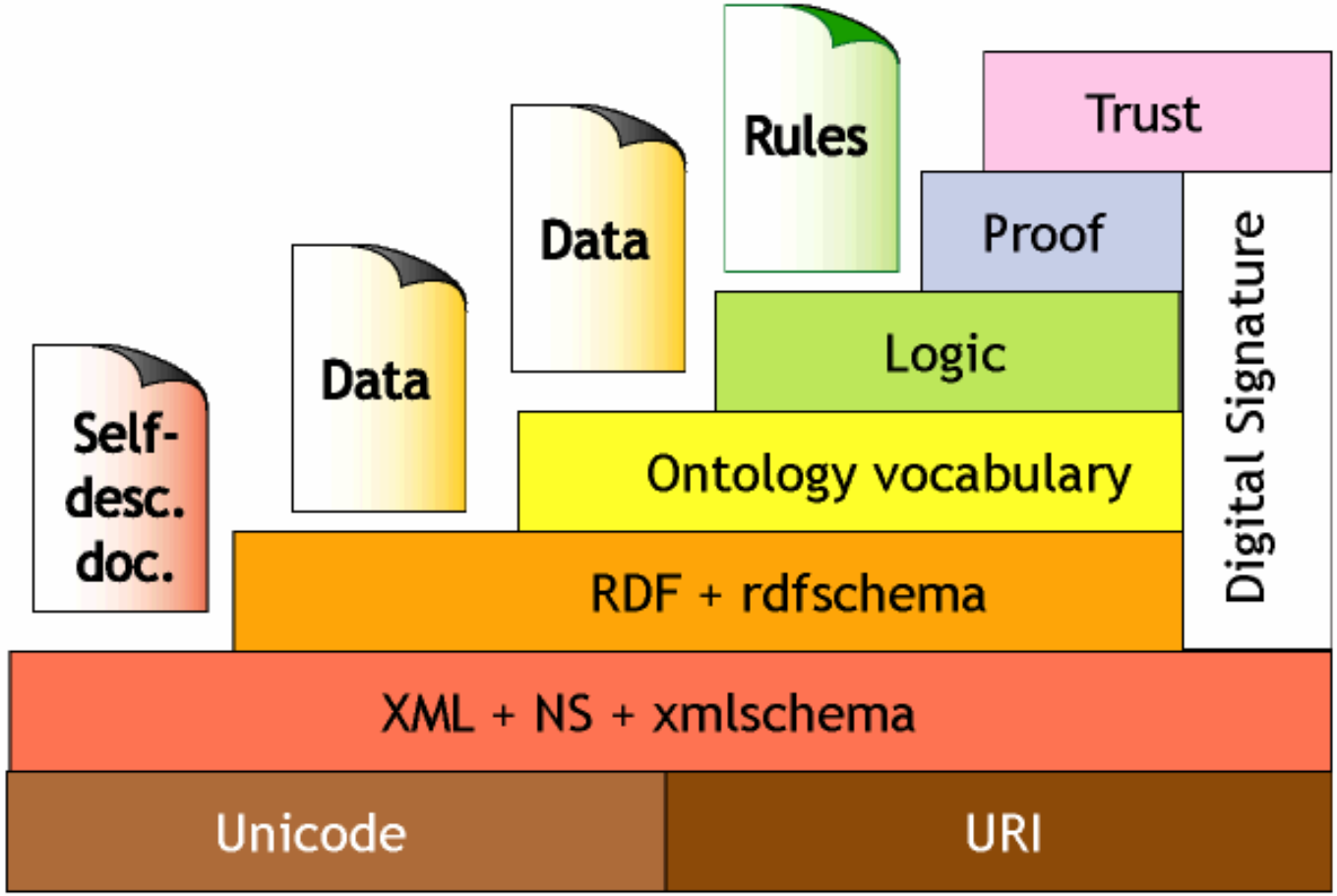
Why do we "need" a semantic web?

- Keyword search engines will continue to improve but…
  - High recall, low precision
  - Results sensitive to vocabulary
  - Result granularity is single web page
  - They don't capture "meaning"
- In constrained domains (i.e., not across the entire web) better information management, knowledge representation makes sense
- *"Maybe"* a common infrastructure for KR will lead to web-wide "*semantics*"
- Some of these tools are useful right now (e.g., Project 2)

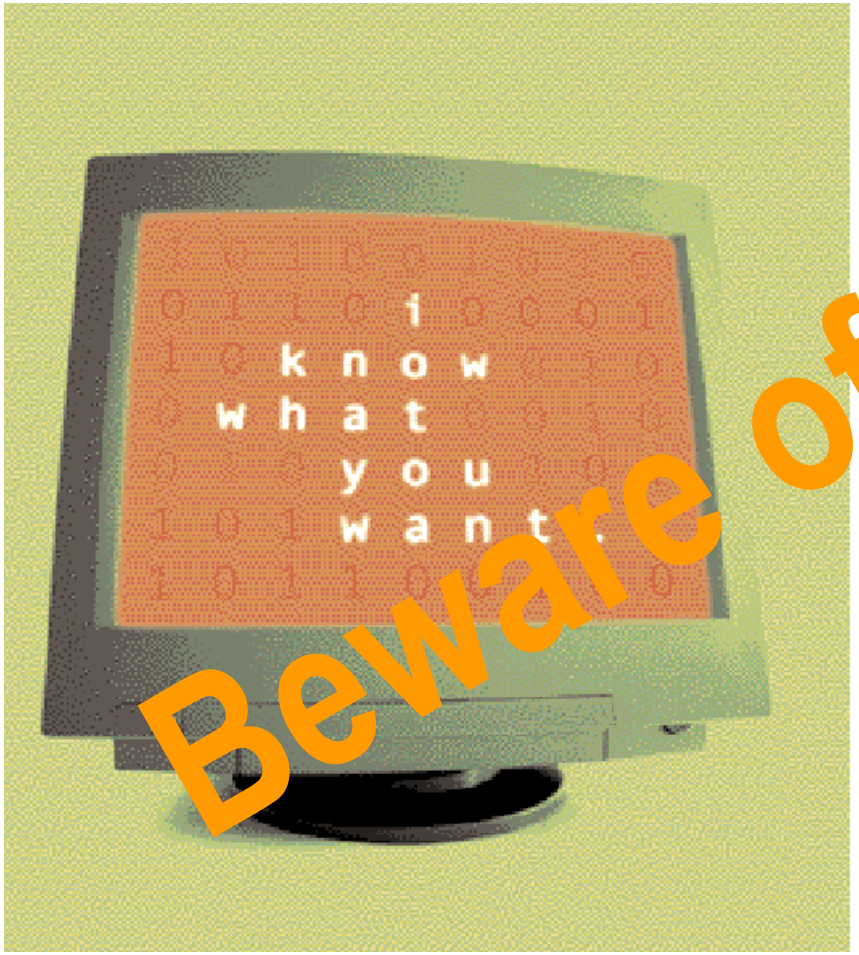
Motivating the "Semantic Web"



Components of the Semantic Web



Scientific American, May 2001:



Beware of the Hype

THE SEMANTIC WEB

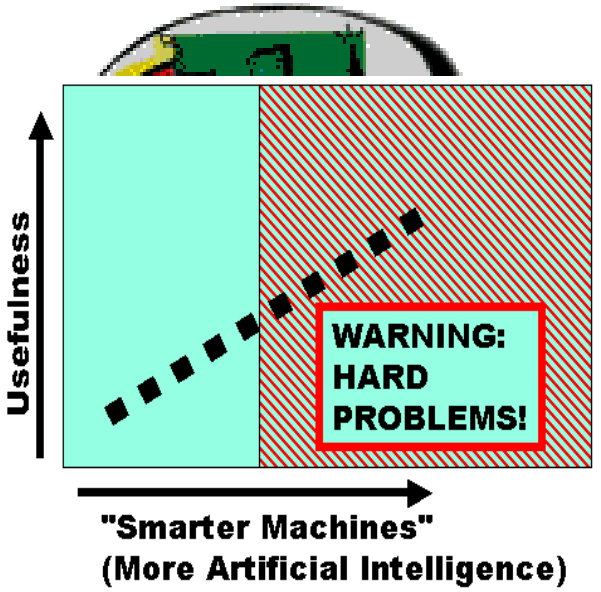
A new form of Web content
that is meaningful to computers
will unleash a revolution of new abilities

by
TIM BERNERS-LEE,
JAMES HENDLER and
ORA LASSILA

PHOTO CREDIT HERE

Beware of the Hype

- Hype seems to suggest that Semantic Web means: "semantics + web = AI"
  - "A new form of Web content that is meaningful to computers will unleash a revolution of new abilities"
- More realistic to think of it as meaning: "semantics + web + AI = more useful web"
  - Realising the complete "vision" is too hard for now (probably)
  - But we can make a start by adding semantic annotation to web resources



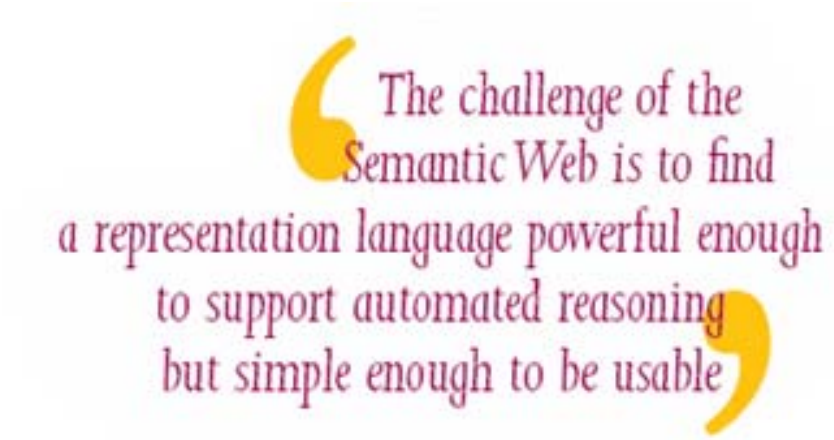
Images from Christine Thompson and David Booth

Knowledge Representation

- Objects/Instances/Individuals
  - Elements of the domain of discourse
  - Equivalent to constants in FOL
- Types/Classes/Concepts
  - Sets of objects sharing certain characteristics
  - Equivalent to unary predicates in FOL
- Relations/Properties/Roles
  - Sets of pairs (tuples) of objects
  - Equivalent to binary predicates in FOL

- Such languages are/can be:
  - Well understood
  - Formally specified
  - (Relatively) easy to use
  - Amenable to machine processing

There has been lots of work on Knowledge Representation but…



"All tractable languages are useless;
all useful languages are intractable"

Challenges of Web to KR

- Scale
- Distributed
- Dynamic
- Paradoxes
- Incomplete language
  - Closed world vs. open world assumptions

Modeling & Encoding Knowledge: RDF

- RDF (Resource Description Format)
- Provides enabling technology for richly-structured information
  - Support for and integration of multiple independent vocabularies
- Rich data model supporting notions of distinct entities and properties
  - Formal model with basis in logic
- Expressible in machine readable manner (e.g., XML)

RDF Components

- Formal data model
- Syntax for interchange of data
- Schema Type system (schema model)
- Syntax for machine-understandable schemas
- Query and profile protocols

- Ontologies layered on top via extensions to base RDF language (DAML, OIL, OWL)

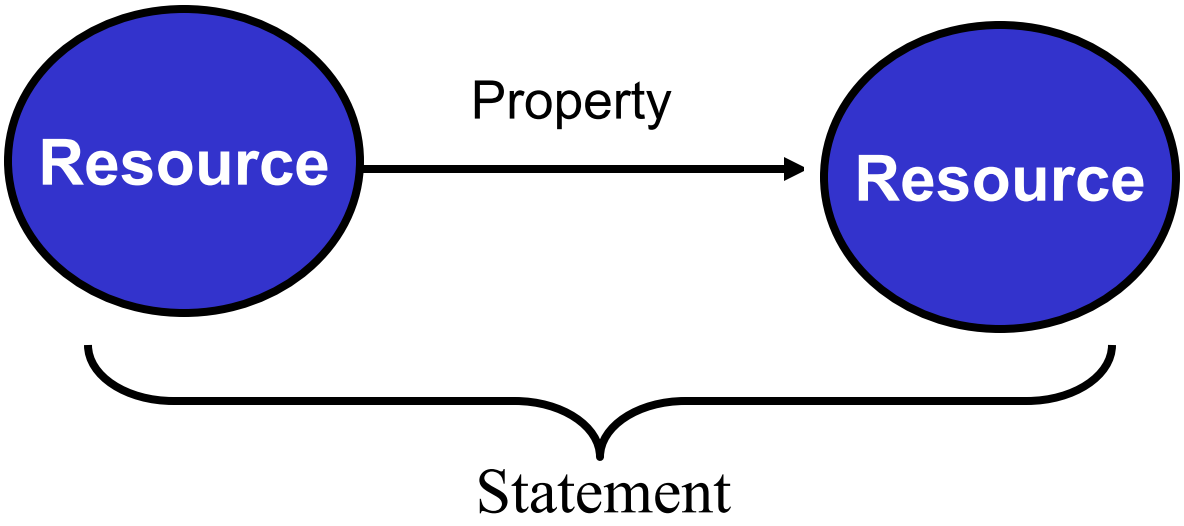
RDF Data Model

- Provides underlying structural foundation for the expression of application (instance) data models
  - for consistent encoding, exchange and processing of information
  - Provides for a basis for interoperability
- Individual communities can then define and express semantics on the basic model
- Model is distinct from the syntax for expressing it
  - XML
  - N3
  - triple notation
  - relational databases (triple-stores in tables)

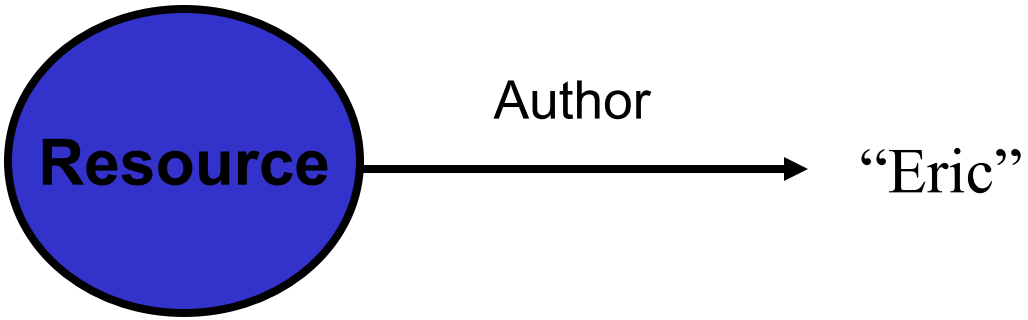
RDF Data Model

- Directed labeled graphs
- Model elements
  - Resource
  - Property
  - Value
  - Statement
  - Containers
- Nodes are all identified using URIs

RDF Model Primitives



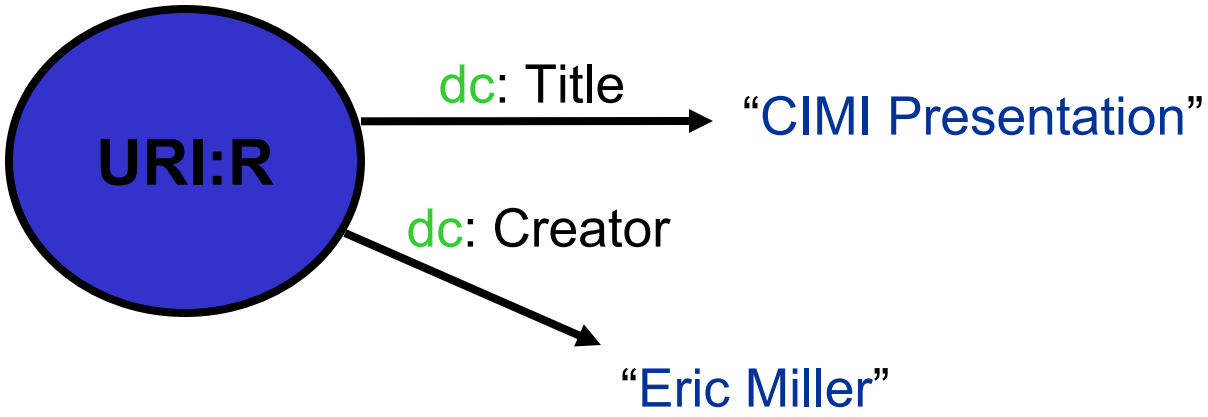
Simple Example



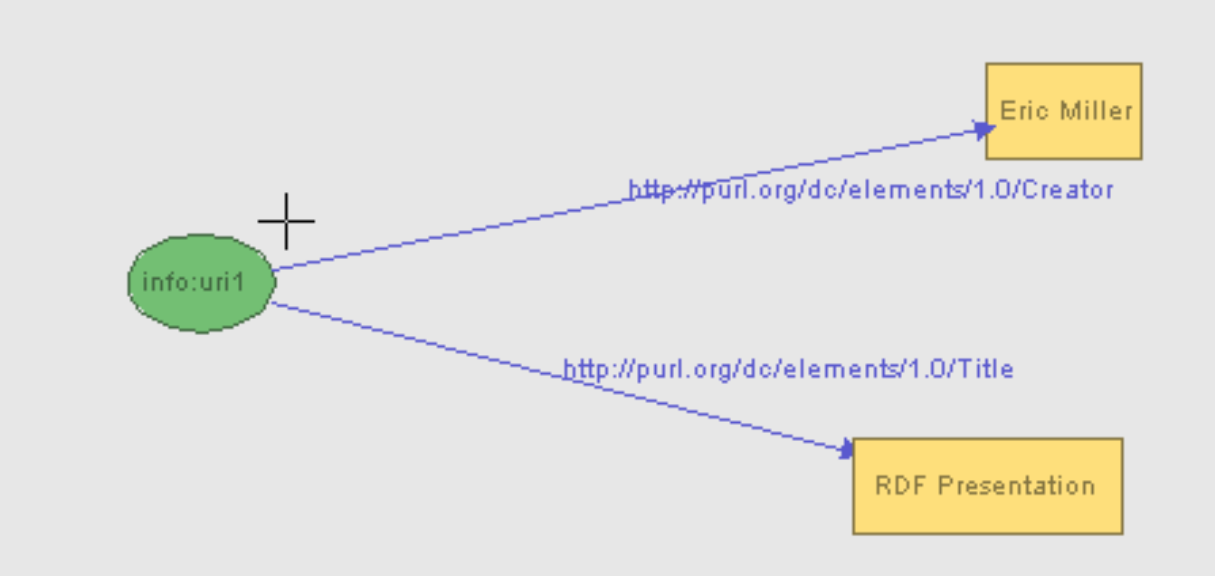
RDF Syntax

- RDF Model defines a formal relationships among resources, properties and values
- Syntax is required to...
  - Store instances of the model into files
  - Communicate files from one application to another
- XML is one well-supported syntax, N3 is another

RDF Model Example #1



RDF Syntax Example #1



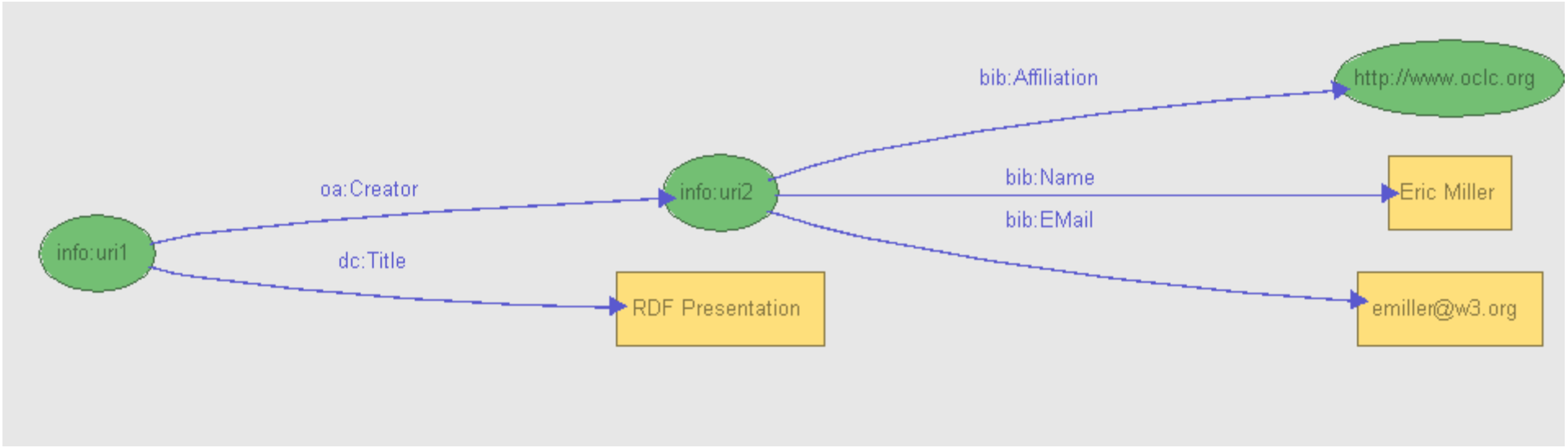

```
<?xml version="1.0"?>
<rdf:RDF xmlns:gss="http://www.w3.org/2001/11/IsaViz/graphstylesheets#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:dc="http://purl.org/dc/elements/1.0/">
 <rdf:Description rdf:about="info:uri1">
  <dc:Title>RDF Presentation</dc:Title>
  <dc:Creator>Eric Miller</dc:Creator>
 </rdf:Description>
</rdf:RDF>
```

N3 Syntax – Example #1

```
@prefix :        <#> .

<info:uri1>
      <http://purl.org/dc/elements/1.0/Creator>
               "Eric Miller" ;
      <http://purl.org/dc/elements/1.0/Title>
               "RDF Presentation" .
```

RDF Model Example #2



RDF/XML Syntax Example #2

```
<?xml version="1.0"?>
<rdf:RDF xmlns:gss="http://www.w3.org/2001/11/IsaViz/graphstylesheets#"
 xmlns:oa="http://agents.org/elements#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:bib="http://www.bib.org/persons#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:dc="http://purl.org/dc/elements/1.0/"
 xml:base="file:/C:/IsaViz/tmp/tmp41406.rdf">
 <rdf:Description rdf:about="info:uri2">
  <bib:Affiliation rdf:resource="http://www.oclc.org"/>
  <bib:EMail>emiller@w3.org</bib:EMail>
  <bib:Name>Eric Miller</bib:Name>
 </rdf:Description>
 <rdf:Description rdf:about="info:uri1">
  <oa:Creator rdf:resource="info:uri2"/>
  <dc:Title>RDF Presentation</dc:Title>
 </rdf:Description>
</rdf:RDF>
```

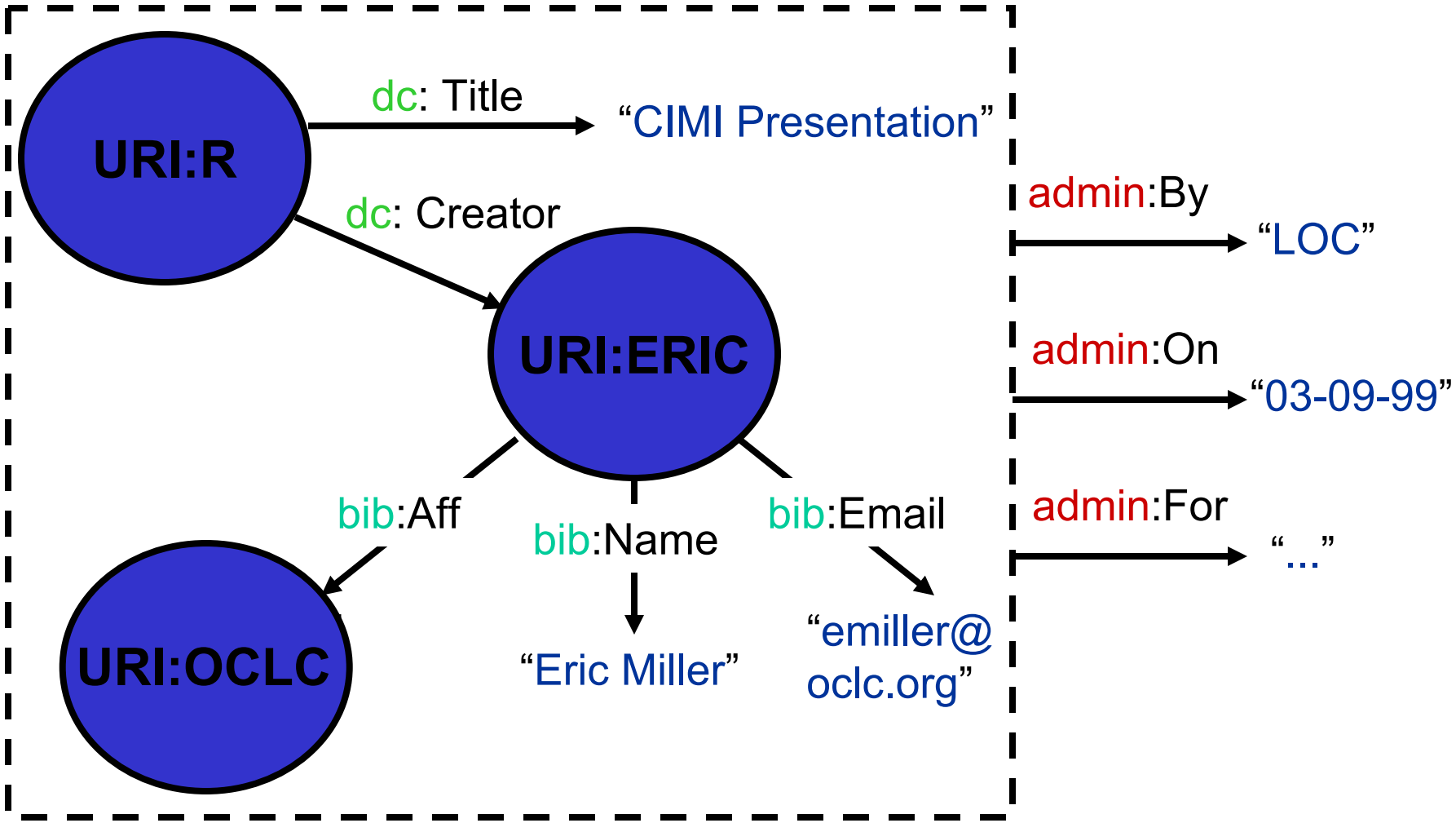
N3 Syntax Example #2

```
@prefix oa: <http://agents.org/elements#> .
@prefix bib: <http://www.bib.org/persons#> .
@prefix dc: <http://purl.org/dc/elements/1.0/> .
@prefix :         <#> .

<info:uri2>
      <bib:Affiliation>
            <http://www.oclc.org> ;
      <bib:EMail>
            "emiller@w3.org" ;
      <bib:Name>
            "Eric Miller" .

<info:uri1>
      <oa:Creator>
            <info:uri2> ;
      <dc:Title>
            "RDF Presentation" .
```

RDF Model Example #3
Reification



RDF Containers

- Permit the aggregation of several values for a property
- Express multiple aggregation semantics
  - unordered
  - sequential or priority order
  - alternative

RDF Containers

- Bag
  - unordered grouping
- Sequence
  - ordered grouping
- Alternatives
  - alternate values
    - need to choose
  - at least one value
  - first value is default or preferred value

RDF - Bag

- Unordered group
- "Carl Lagoze and Stuart Weibel are co-authors"

```
<BIB:Author>
  <Bag>
    <li> Carl Lagoze </li>
    <li> Stuart Weibel </li>
</Bag>
</BIB:Author>
```

RDF - Sequence

- Ordered or priority group
- "Carl Lagoze is primary author and Stuart Weibel is second author"

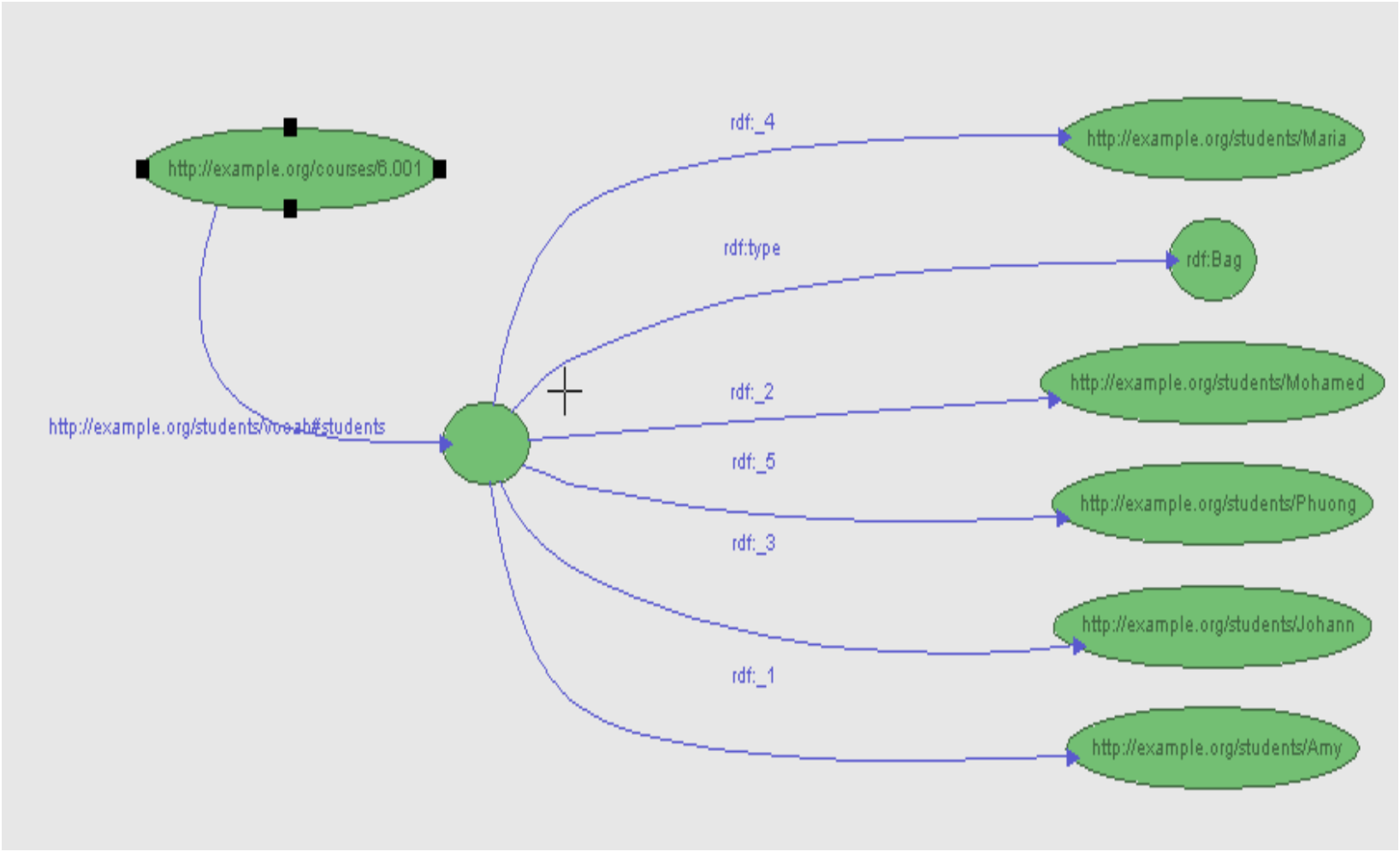
```
<BIB:Author>
  <Seq>
    <li> Carl Lagoze </li>
    <li> Stuart Weibel </li>
</Seq>
</BIB:Author>
```

RDF - Alt

- Client chooses one of several values
- First value is default
- "The distance is 15 kilometers or 9.3 miles"

```
<DC:Coverage>
  <Alt>
    <li> 15KM </li>
    <li> 9.3M </li>
</Alt>
</DC:Coverage>
```

Container Example



RDF/XML for Container

```
<?xml version="1.0"?>
<rdf:RDF xmlns:gss="http://www.w3.org/2001/11/IsaViz/graphstylesheets#"
 xmlns:j.0="http://example.org/students/vocab#" xmlns:oa="http://agents.org/elements#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:bib="http://www.bib.org/persons#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:s="http://example.org/packages/vocab#"
 xmlns:dc="http://purl.org/dc/elements/1.0/"
 xml:base="file:/C:/Documents and Settings/Carl Lagoze/Desktop/ex2rdf.n3">
  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <j.0:students>
      <rdf:Bag>
        <rdf:li rdf:resource="http://example.org/students/Amy"/>
        <rdf:li rdf:resource="http://example.org/students/Mohamed"/>
        <rdf:li rdf:resource="http://example.org/students/Johann"/>
        <rdf:li rdf:resource="http://example.org/students/Maria"/>
        <rdf:li rdf:resource="http://example.org/students/Phuong"/>
      </rdf:Bag>
    </j.0:students>
  </rdf:Description>
</rdf:RDF>
```

N3 for Container

```
# Base: file:/C:/Documents and Settings/Carl Lagoze/Desktop/ex2rdf.n3
@prefix :        <#> .

<http://example.org/courses/6.001>
      <http://example.org/students/vocab#students>
              [ a       <http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag> ;
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1>
                        <http://example.org/students/Amy> ;
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#_2>
                        <http://example.org/students/Mohamed> ;
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#_3>
                        <http://example.org/students/Johann> ;
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#_4>
                        <http://example.org/students/Maria> ;
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#_5>
                        <http://example.org/students/Phuong>
              ] .
```

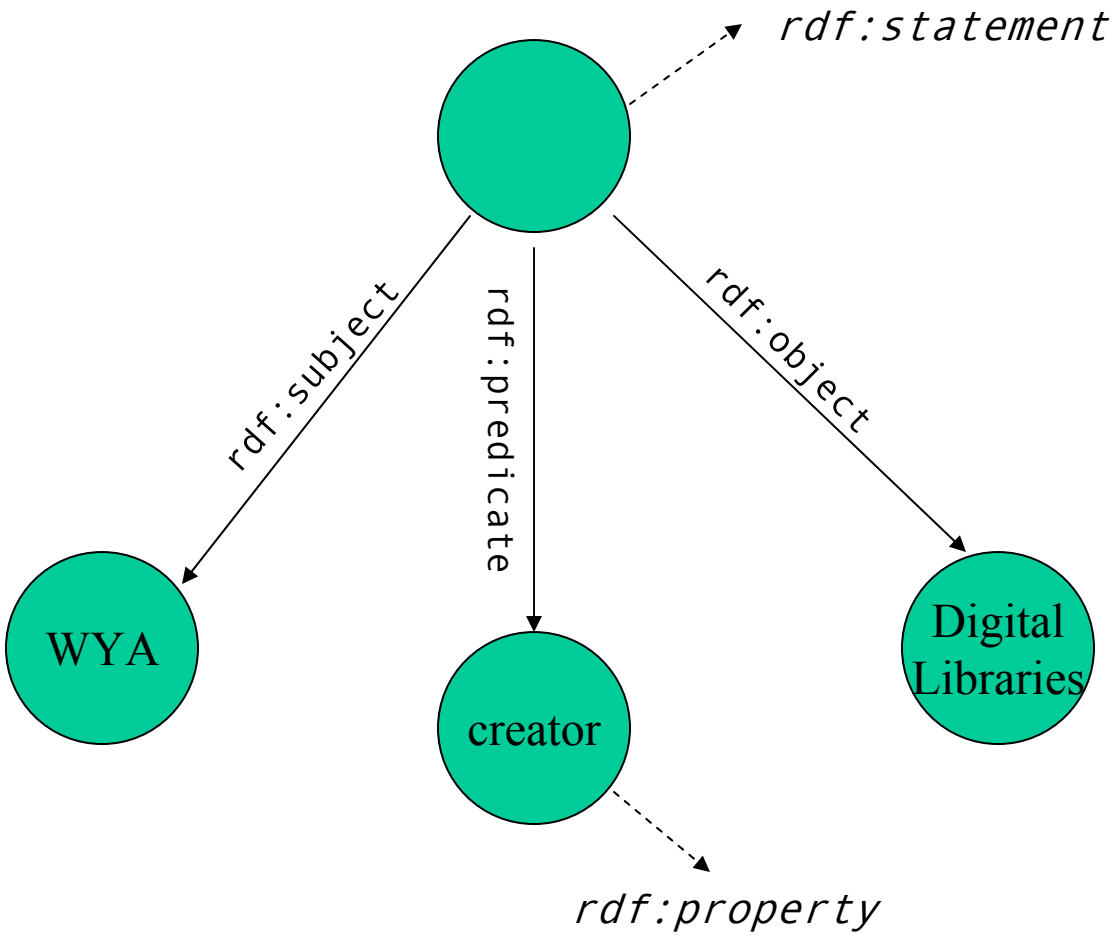
RDF meta-model

- RDF basic types
  - `rdf:Resource` – everything that can be identified (with a URI)
  - `rdf:Property` – specialization of a resource expressing a binary relation between two resources
  - `Rdf:type` – predefined property to express that subject of property is considered to be an *instance* of that category or class defined by the value of the property
  - `rdf:statement` – a triple with properties `rdf:subject`, `rdf:predicate`, `rdf:object`
- An RDF statement is a triple consisting of a resource (subject), a property and a second resource (object)
  - `(:s :p :o)`
- Expressible also as binary relations
  - P(S,O) – e.g., Title(R, "War & Peace")

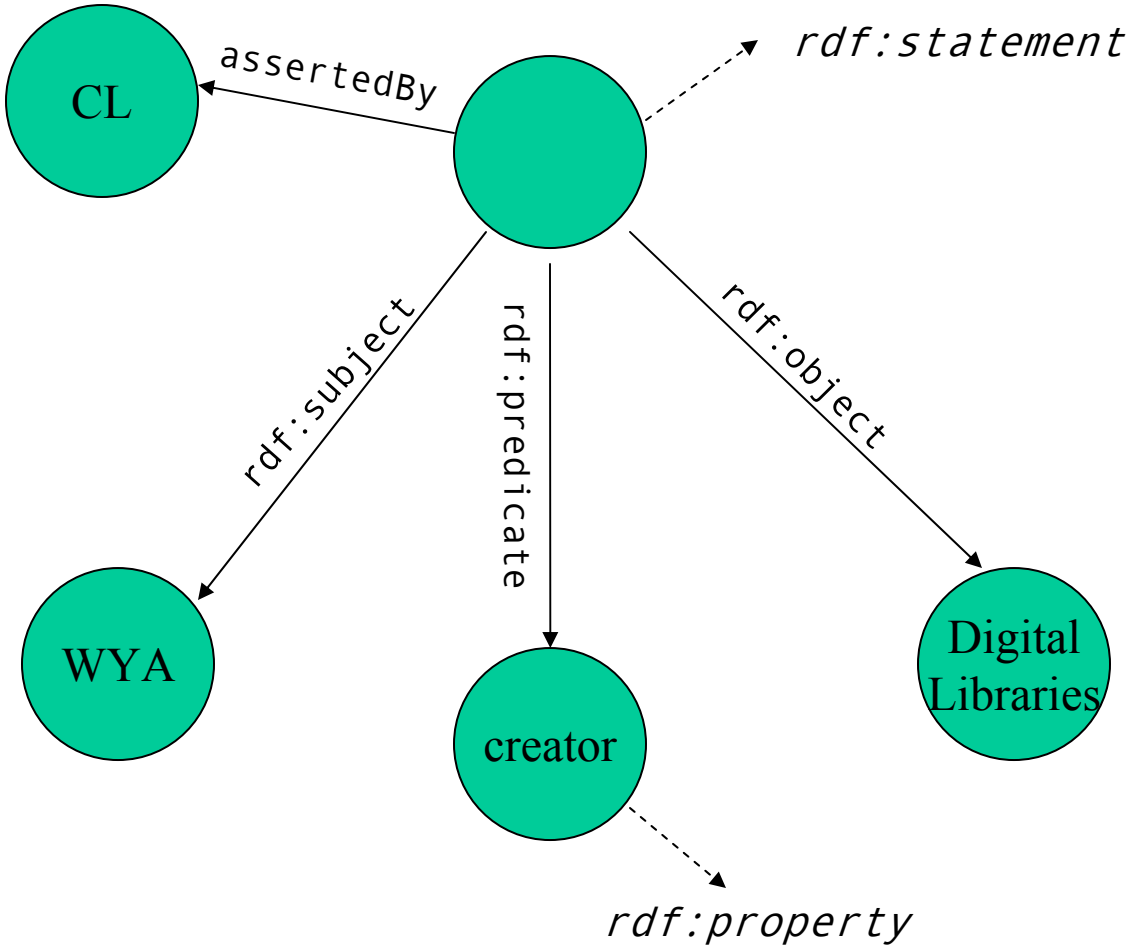
RDF triple model

explicit triple	Allows inference
(:s :p :o)	(:s rdf:type rdf:Resource) (:p rdf:type rdf:Property) (:o rdf:type rdf:Resource)

RDF statements and basic types



Reification – Statements about statements



"CL says 'WYA wrote Digital Libraries'"