General idea of Semantic Web

Make current web more machine accessible
(currently all the intelligence is in the user)

Do this by:

1. Making **data and meta-data** available on the Web in machine-understandable form (**formalised**)
2. Structure the data and meta-data in **ontologies**

These are non-trivial design decisions.
Alternative would be:

Which Semantic Web?

- Version 1:
  "Enrichment of the current Web"

- **recipe**:
  Annotate, classify, index
- **meta-data from**:
  - automatically producing markup:
    named-entity recognition,
    concept extraction, tagging, etc.
- **enable** personalisation, search, browse,...

Which Semantic Web?

- Version 2:
  "Semantic Web as Web of Data" (TBL)

- **recipe**:
  expose databases on the web,
  use RDF, integrate
- **meta-data** from:
  - expressing DB schema semantics
    in machine interpretable ways
- **enable** integration and unexpected re-use

Which Semantic Web?

- Version 1:
  "Enrichment of the current Web"

- Version 2:
  "Semantic Web as Web of Data"

- Different use-cases
- Different techniques
- Different users

Which Semantic Web?

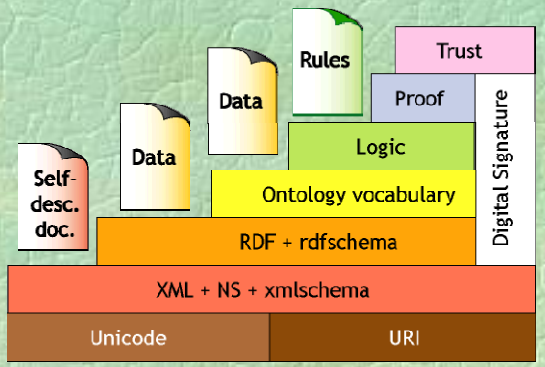
- Version 1:
  "Enrichment of the current Web"

- Version 2:
  "Semantic Web as Web of Data"

- V1 is often publicised, popularised
- V2 has most current business potential
- V2 has made most progress

RDF & OWL Briefing

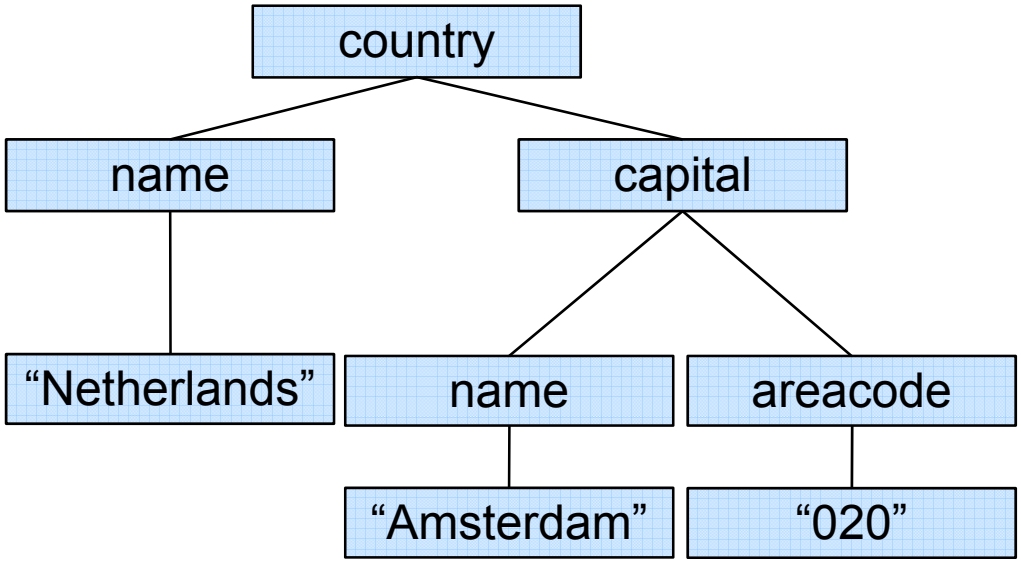
W3C Stack



- **XML**:
  - Surface syntax, no semantics
- **XML Schema**:
  - Describes structure of XML documents
- **RDF**:
  - Datamodel for “relations” between “things”
- **RDF Schema**:
  - RDF Vocabulary Definition Language
- **OWL**:
  - A more expressive Vocabulary Definition Language

What was XML again?

```
<country name=”Netherlands”>
  <capital name=”Amsterdam”>
    <areacode>020</areacode>
  </capital>
</country>
```



So why not just use XML?

- No **agreement** on:
  - structure
    - is country a:
      - object?
      - class?
      - attribute?
      - relation?
      - something else?
    - what does nesting mean?
  - vocabulary
    - is country the same as nation?

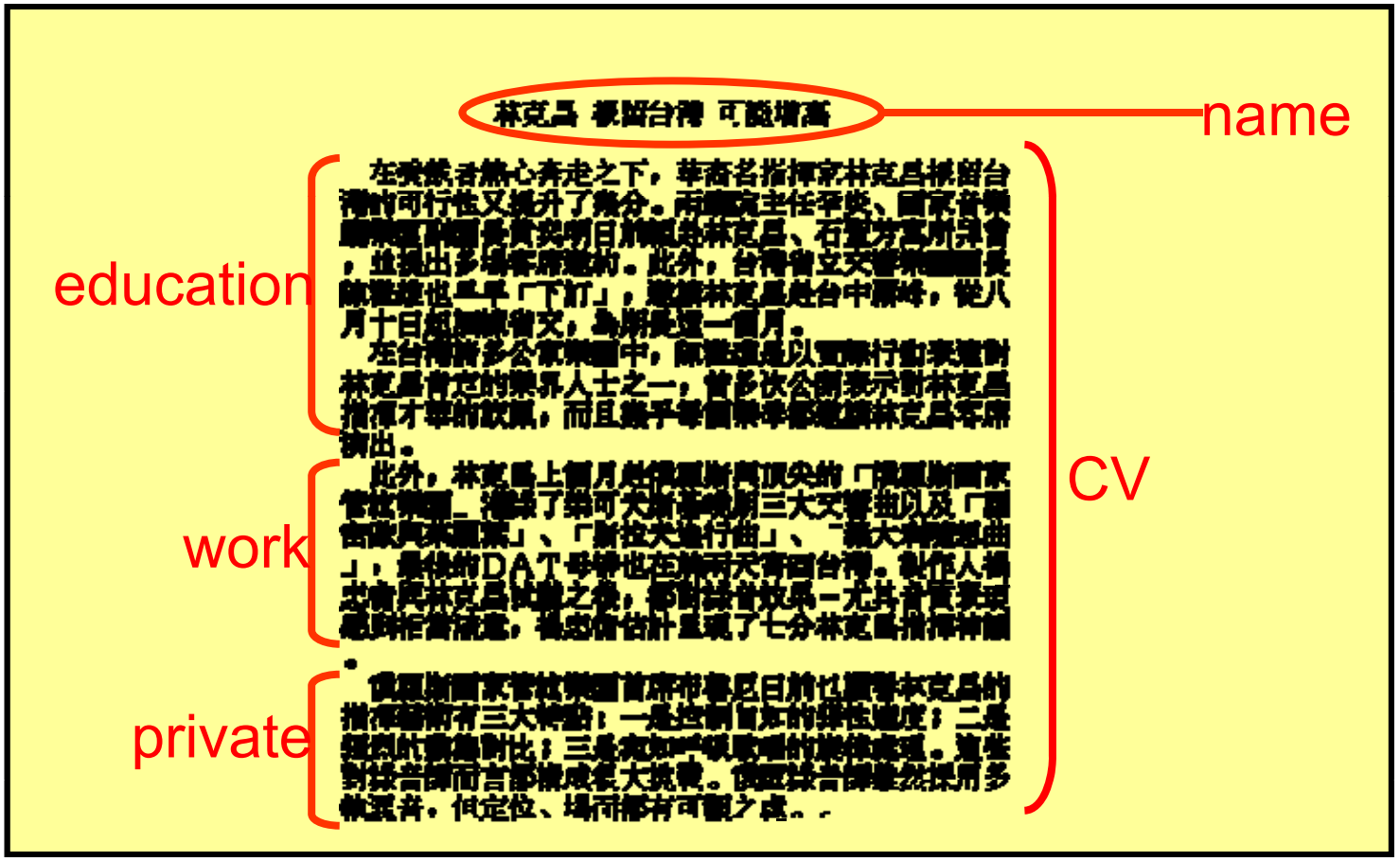
```
<country name=”Netherlands”>
  <capital name=”Amsterdam”>
    <areacode>020</areacode>
  </capital>
</country>
```

```
<nation>
  <name>Netherlands</name>
  <capital>Amsterdam</capital>
  <capital_areacode>
    020
  </capital_areacode>
</nation>
```

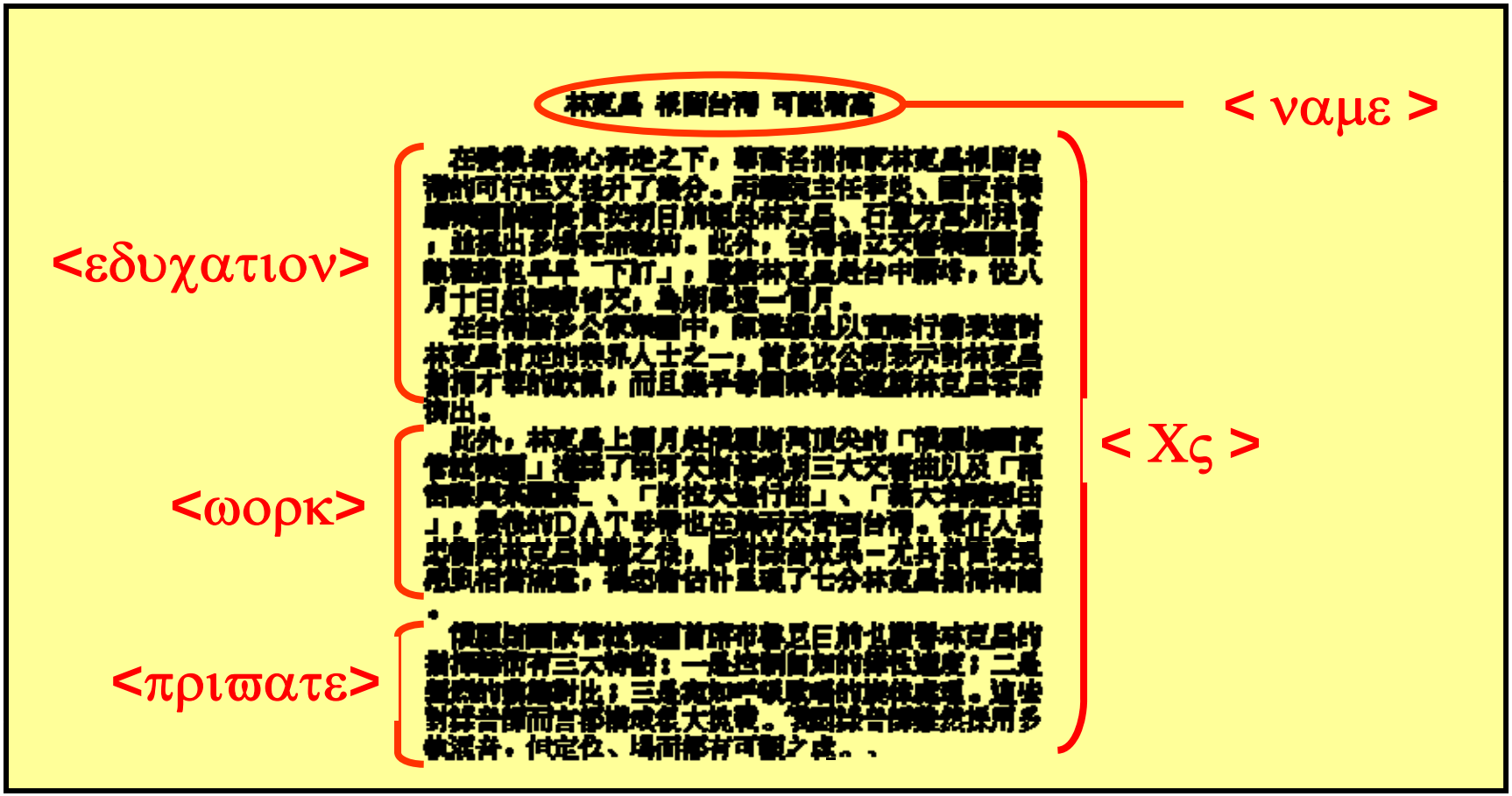
- Are the above XML documents the same?
- Do they convey the same information?
- Is that information machine-accessible?

machine accessible meaning

(What it's like to be a machine)



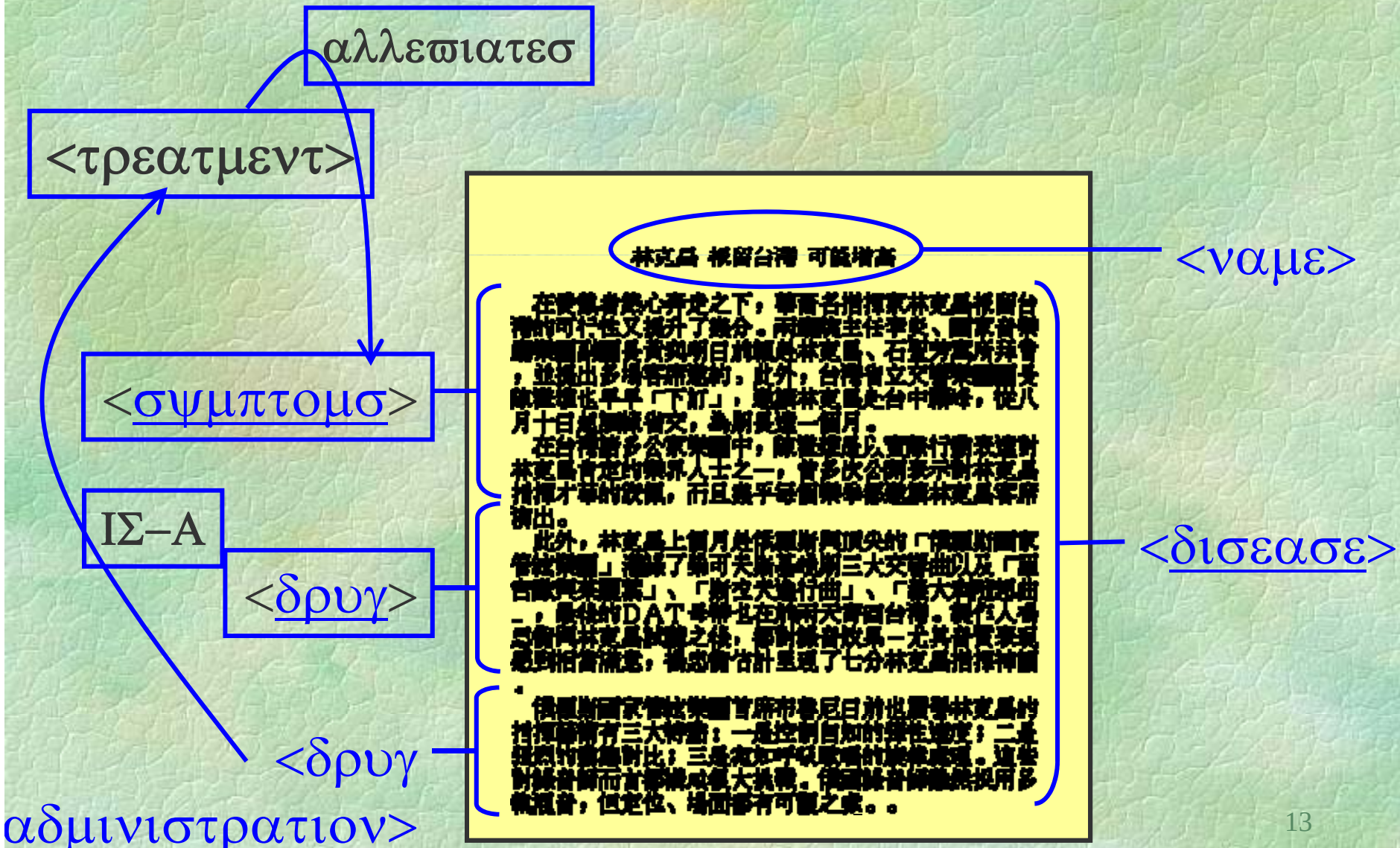
XML ≠
machine accessible meaning



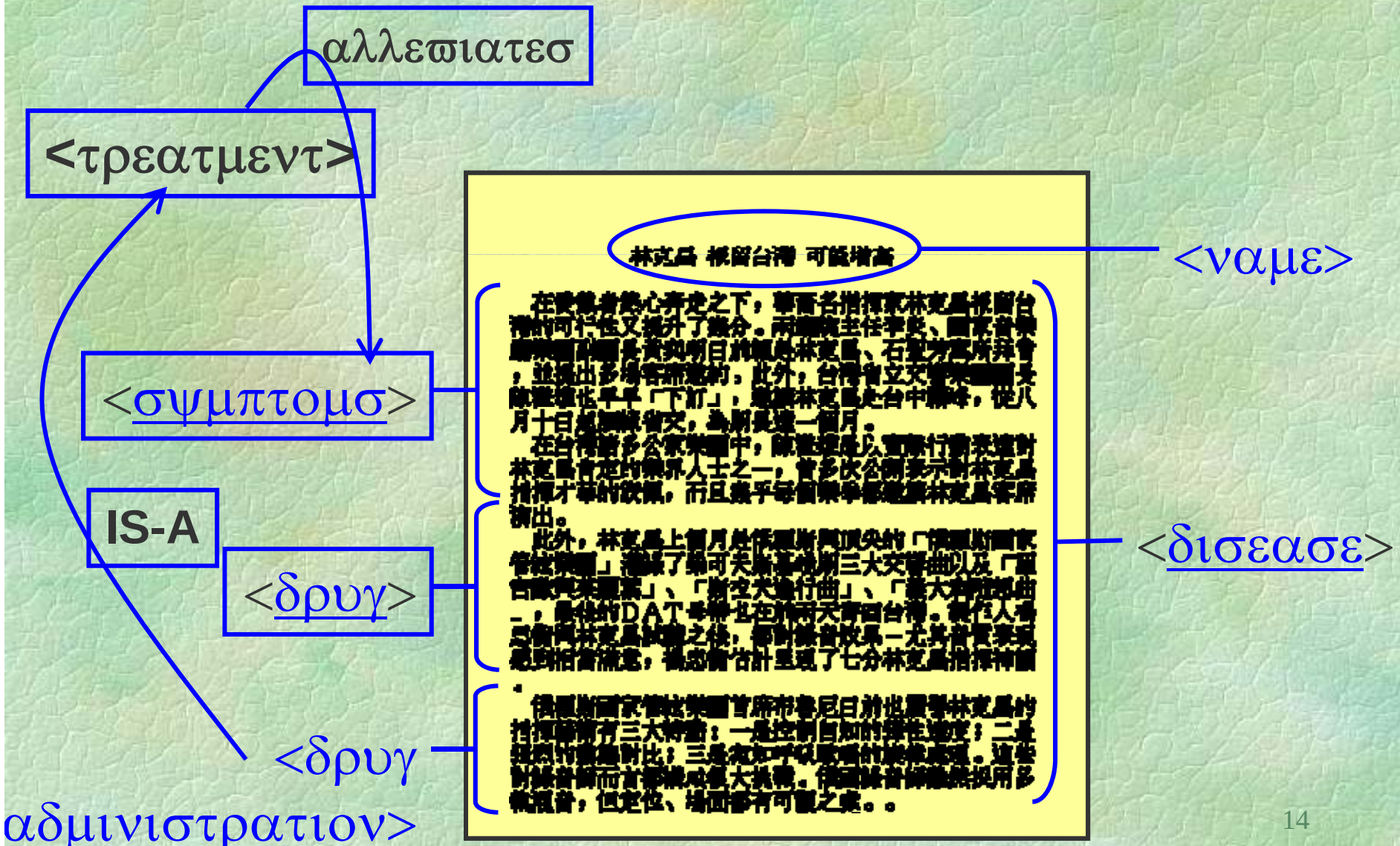
What is RDF?

- RDF
  - stands for Resource Description Framework
  - is a W3C Recommendation (http://www.w3.org/RDF)
- RDF is a data model
  - for representing **metadata (data about data)**
  - **for describing the semantics of information in a machine-accessible way**
- **What can you use it for?**
  - **intelligent information brokering**
  - **meaning-based computing**
  - **agent communication**

Meta-data in XML

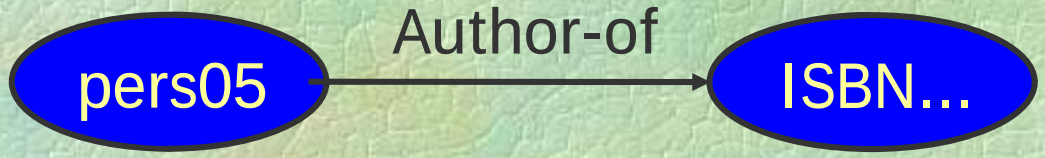


Meta-data in RDF

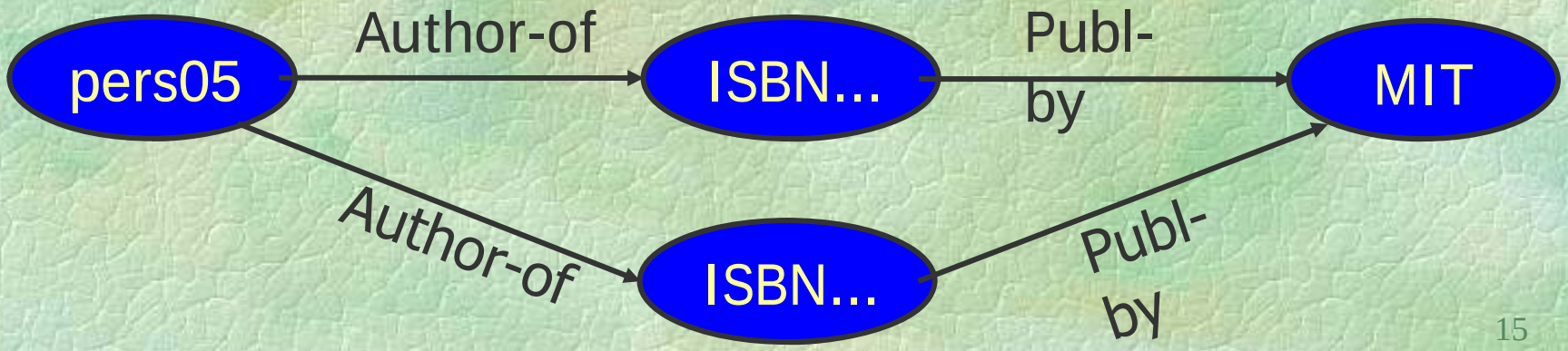


Bluffer's guide to RDF (1)

- **Object ->Attribute-> Value** triples



- objects are **web-resources**
- Value is again an Object:
  - triples can be **linked**
  - data-model = graph

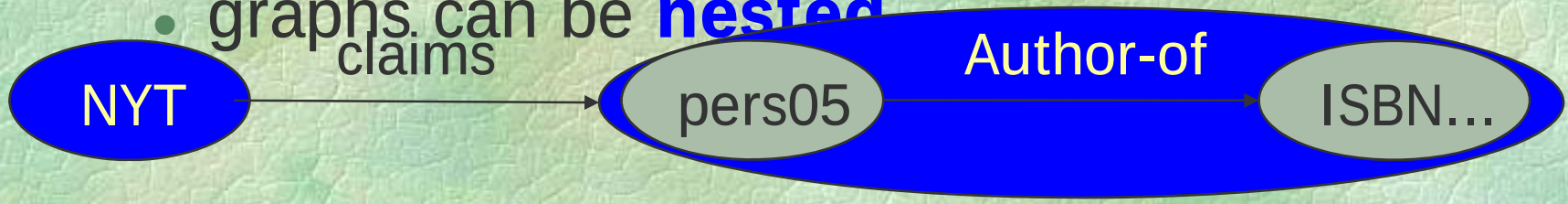


Bluffer's guide to RDF (2)

- Every identifier is a URL
  = world-wide unique naming
- Has XML syntax

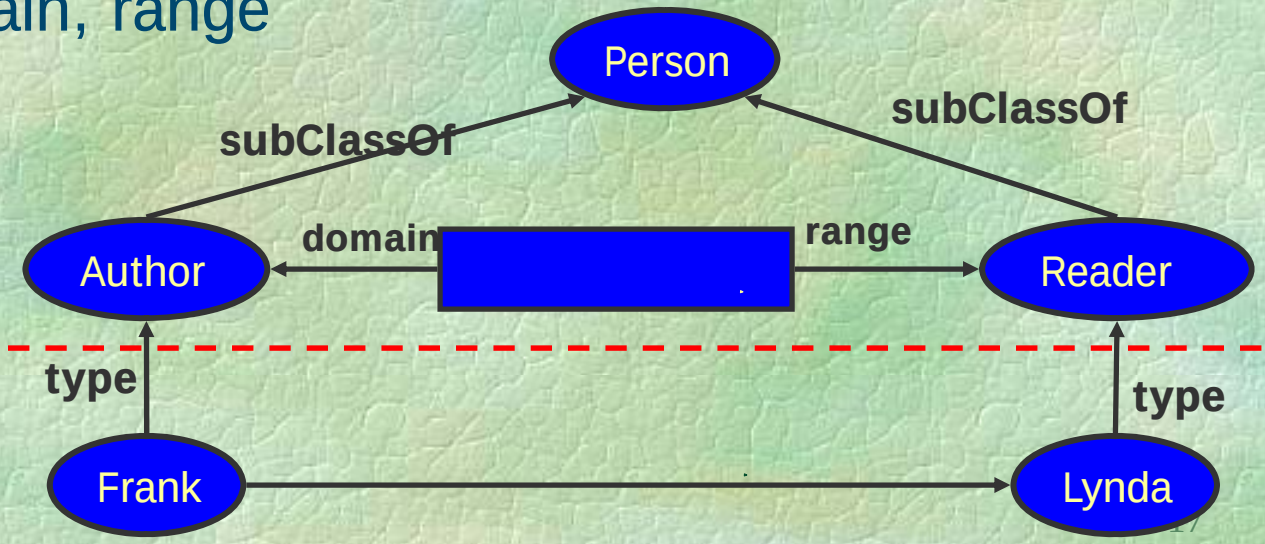
```
<rdf:Description rdf:about="#pers05">
  <authorOf>ISBN...</authorOf>
</rdf:Description>
```

- Any statement can be an object
  - graphs can be **nested**



What does RDF Schema add?

- Defines **vocabulary** for RDF
- Organizes this vocabulary in a **typed hierarchy**
  - Class, subClassOf, type
  - Property, subPropertyOf
  - domain, range



RDF(S) have a (very small) formal semantics

- Defines what other statements are **implied** by a given set of RDF(S) statements
- Ensures mutual **agreement on minimal content** between parties without further contact
- In the form of “entailment rules”
- Very **simple to compute** (and not explosive in practice)

RDF(S) semantics: examples

- Aspirin **isOfType** Painkiller
  Painkiller **subClassOf** Drug
  ➔ Aspirin **isOfType** Drug

- aspirin alleviates headache
  alleviates **range** symptom
  ➔ headache **isOfType** symptom

RDF(S) semantics: examples

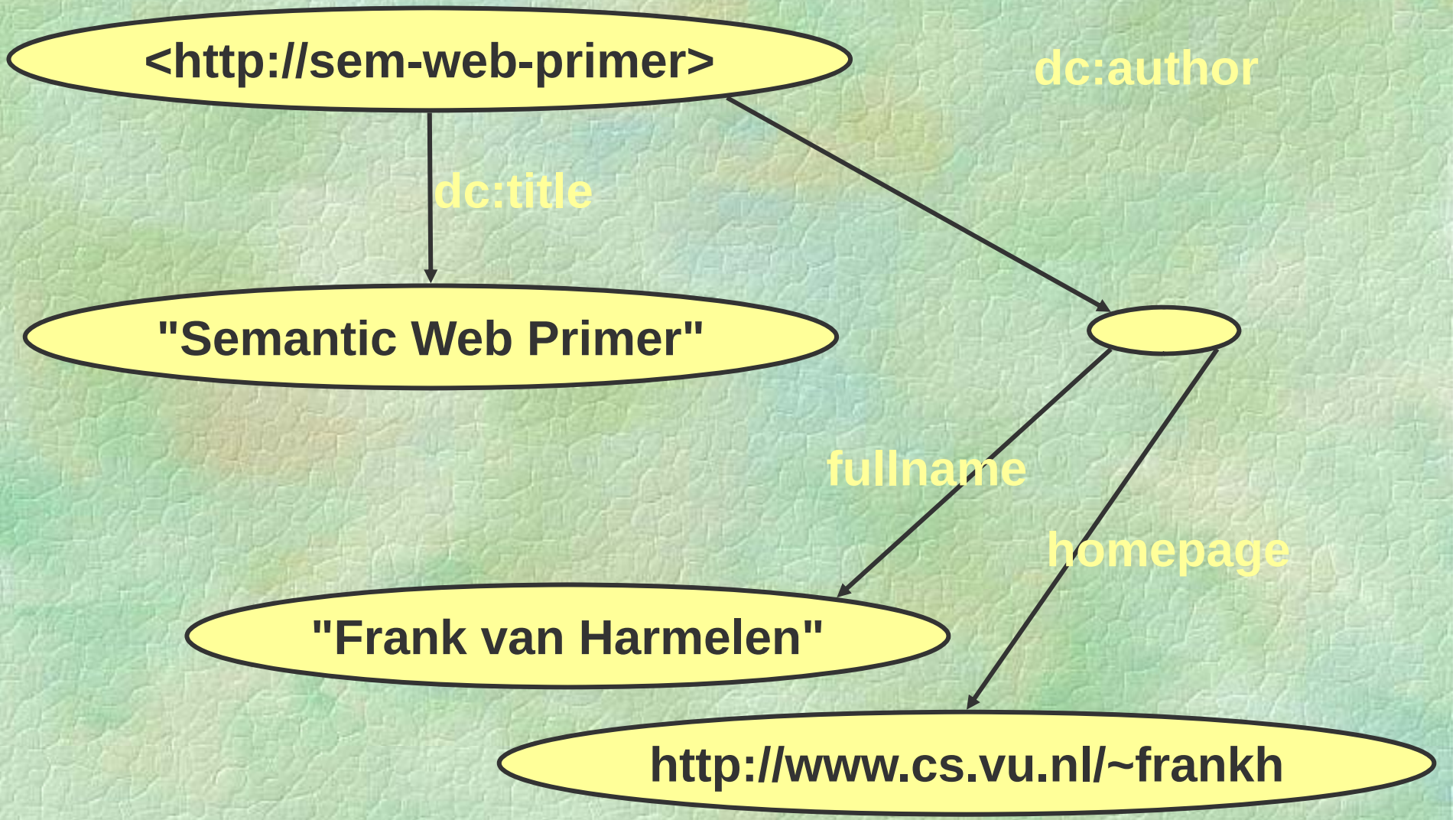
- Ασπιριν **isOfType** Παινκιλλερ
  Παινκιλλερ **subClassOf** Δρυγ
  ➔ Ασπιριν **isOfType** Δρυγ

- ασπιριν αλλεωιατεσ ηεαδαχηε
  τρεατσ **range** σψμπτομ
  ➔ ηεαδαχηε **isOfType** σψμπτομ

RDF(S) semantics

- X R Y + R **domain** T → X **IsOfType** T
- X R Y + R **range** T → Y **IsOfType** T
- T1 **SubClassOf** T2 +
  T2 **SubClassOf** T3 → T1 **SubClassOf** T3
- X **IsOfType** T1 +
  T1 **SubClassOf** T2 → X **IsOfType** T1

RDF(S) syntax: graphics



RDF(S) syntax: XML

```
<rdf:RDF>
  <rdf:Description rdf:about="http://sem-web-primer"
   dc:title="Semantic Web Primer">
    <dc:author>
      <rdf:Description fullname="Frank van Harmelen">
        <homePage rdf:resource="http://www.cs.vu.nl/~frankh"/>
      </rdf:Description>
    </dc:author>
  </rdf:Description>
</rdf:RDF>
```

RDF(S) syntax: Turtle

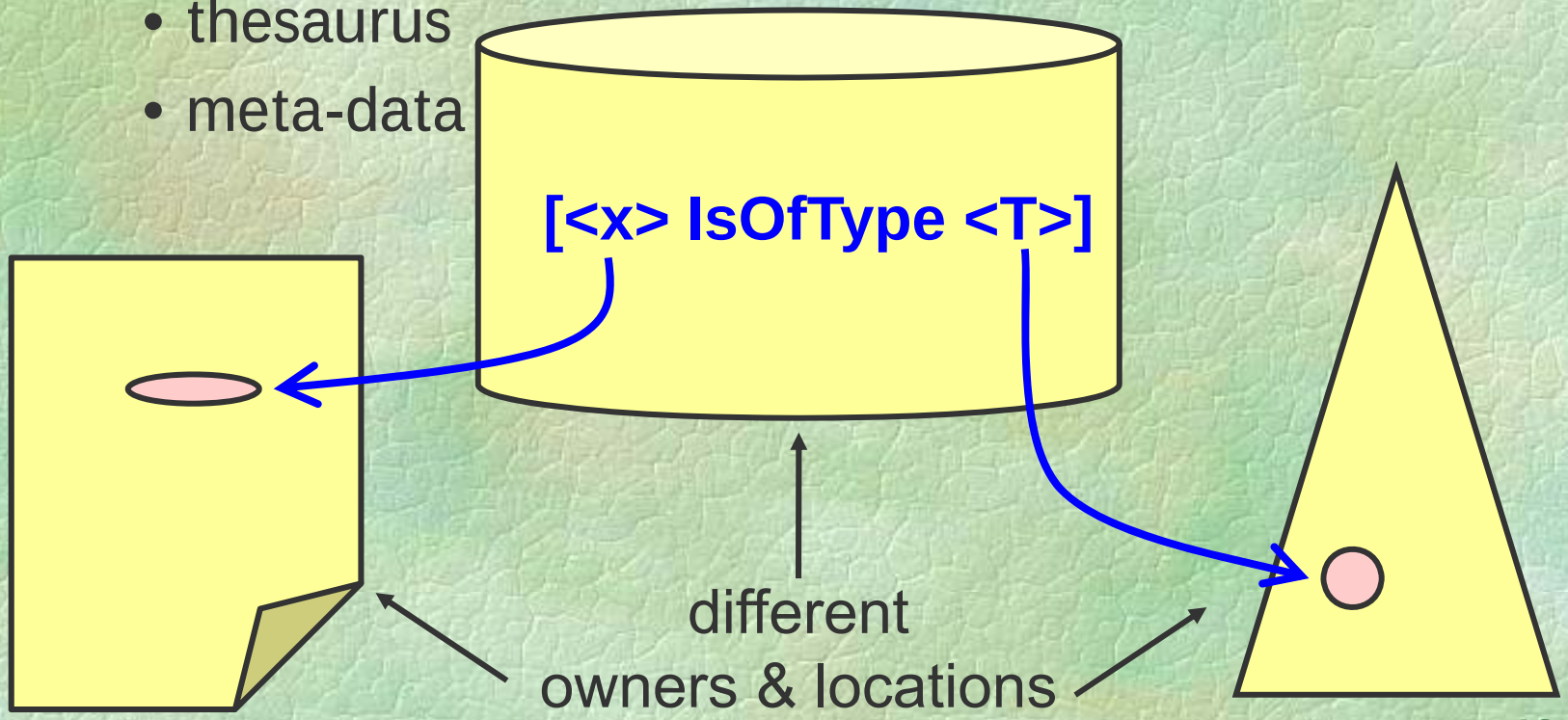
```
<http://sem-web-primer>
  dc:title "Semantic Web Primer" ;
  dc:author [
    fullname "Frank van Harmelen";
    homePage <http://www.cs.vu.nl/~frankh>
  ] .
```

RDF(S) / XML relationship

- XML is a just a syntax for RDF(S)
  - (one of many)
- RDF(S) assigns meaning to some terms
  - (XML doesn't)
- This allows greater interoperability:
  - tools/tools
  - thesaurus/thesaurus
  - tools/thesaurus

RDF(S) / XML relationship

- All identifiers are URL's
  - Allows total decoupling of
    - document
    - thesaurus
    - meta-data



RDF(S) interoperability: example: EMTREE → UMLS

Work by Heiner Stuckenschmidt@VU
and Maria Taboada@Santiago

1. Converted EMTREE to RDF(S)
2. Load into existing RDF(S) editor (Protégé)
3. Use existing RDF(S) wrapper for UMLS
4. Deploy existing linguistic term mapper

RDF(S) interoperability: example: EMTREE → UMLS

- 24305 EMTREE pref.names → unique UMLS concept
- 2051 EMTREE pref.names → multiple UMLS concepts
- 20071 EMTREE pref. name → no UMLS concepts
- 34332 EMTREE pref. names + synonyms → some UMLS concept(s): 74%

Effort = days

RDF(S) / XML conversion

- step-wise process description exists
- hardest part is:
  - mentally re-engineering the thesaurus model

➔ make this model as sharable as possible

➔ RDF does, XML doesn't

Summary in quotes

"RDF developers focus on its non-anglebracketty abstract information model rather than its representation in markup"

"the RDF information model is couched in terms of "resources" (aka things, objects, entities...) and their "properties" (aka relationships)"

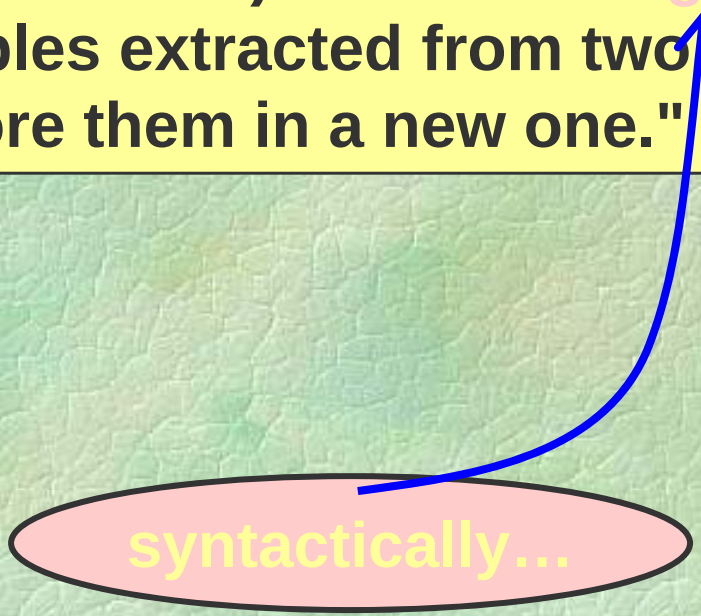
"RDF offers XML tools a way of being explicit about the content of (some subset of) XML documents"

"RDF can be used to represent the claims implicit in XML Linking elements […] we can think about the resulting RDF data as a characterisation of what the XML was telling us"

"RDF cares about the messages encoded in XML, not about the specific form of their encoding in elements and attributes"

Summary in quotes

"There is no algorithm for merging two XML Infosets, to enable us to pool knowledge acquired from diverse sources. The RDF information model, by constrast, was designed with data aggregation (rather than structured documents) in mind. Merging RDF data is trivial: add the triples extracted from two RDF/XML documents, and store them in a new one."



Things RDF(S) can't do

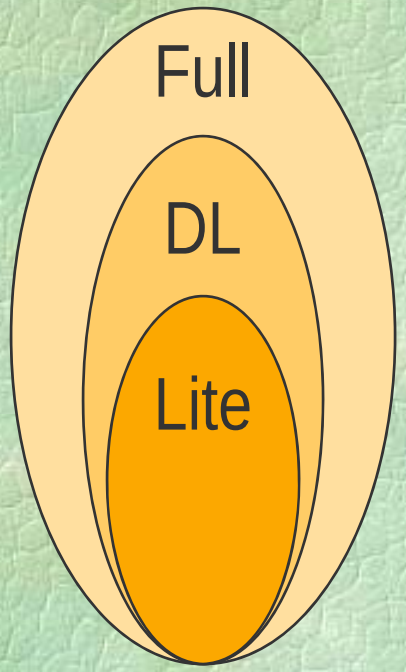
- equality
- enumeration
- number restrictions
  - Single-valued/multi-valued
  - Optional/required values
- inverse, symmetric, transitive
- boolean algebra
  - Union, complement
- ...

OWL: more expressivity

- **OWL Light**
  - (sub)classes, individuals
  - (sub)properties, domain, range
  - conjunction

  RDF Schema

  - (in)equality
  - cardinality 0/1
  - datatypes
  - inverse, transitive, symmetric
  - hasValue
  - someValuesFrom
  - allValuesFrom



- **OWL DL**
  - Negation
  - Disjunction
  - Full Cardinality
  - Enumerated types
- **OWL Full**
  - Allow meta-classes etc

OWL also has a formal semantics

- Defines what other statements are **implied** by a given set of statements
- Ensures **mutual agreement** on content (both **minimal and maximal**) between parties without further contact
- Can be used for integrity/**consistency checking**
- Hard to compute (and *rarely/sometime/always explosive in practice*)

OWL semantics: minimal

- vanGogh isOfType Impressionist
  Impressionist subClassOf Painter
  ➔ **vanGogh isOfType Painter**

- vanGogh painter-of sunflowers
  painter-of domain painter
  ➔ **vanGogh isOfType painter**

OWL semantics: maximal

- vanGogh isOfType Impressionist
  Impressionist disjointFrom Cubist
  ➔ **NOT: vanGogh isOfType Cubist**

- painted-by has-cardinality 1
  sun-flowers painted-by vanGogh
  Picasso different-individual-from vanGogh
  ➔ **NOT: sun-flowers painted-by Picasso**