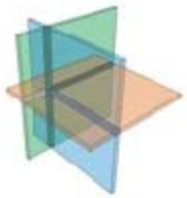
Semantic Web and Multimedia

- Motivation & Agenda –

Steffen Staab
http://isweb.uni-koblenz.de



UNIVERSITÄT KOBLENZ · LANDAU

X-MEDIA

Semantic Aspects

Web Aspects

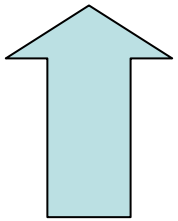
Semantics in Multimedia

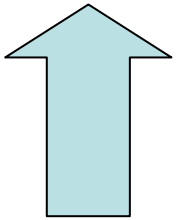
Multimedia on the Semantic Web

Semiotics

Formal semiotics – logics and probabilities as core tools

Social dynamics – the Web as an IT enabler





What is the meaning of some multimedia data?

How are things interwoven, i.e. networked?

Tasks & Purpose

Find me pictures about a boy on the beach

Find me cute pictures about Nico to show to my Mum

Find me pictures from Greece

Find me pictures from our common holiday

- with a resolution of ….
- with no other person in the foreground…

What does this mean?

Content semantics (general knowledge):
boy; beach; jet ski; sea; sky

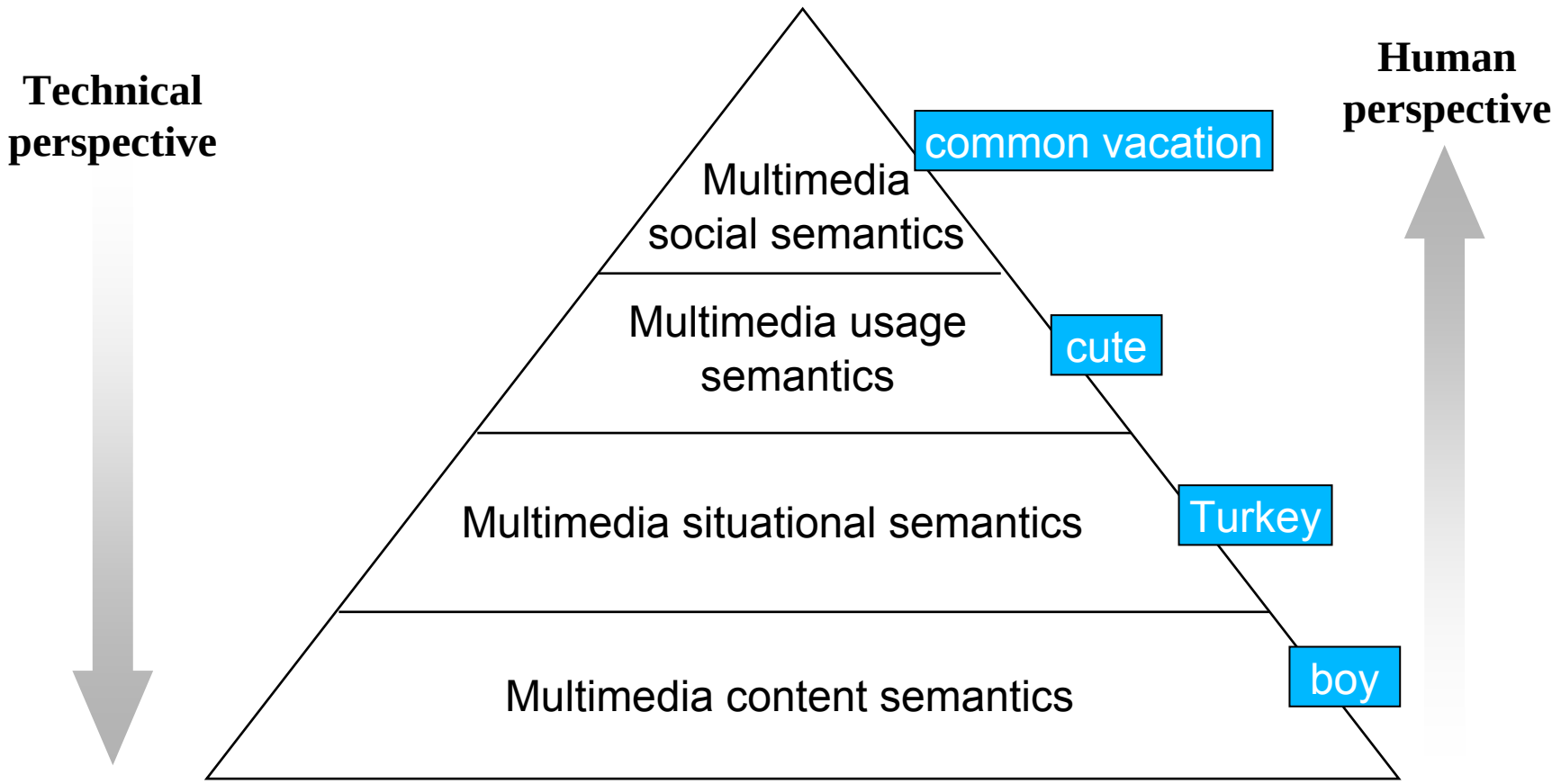
Content semantics („private knowledge“):
Nico digging

Retrieval semantics:
cute picture to show

Situational semantics:
Chalkidiki, Greece, 2006

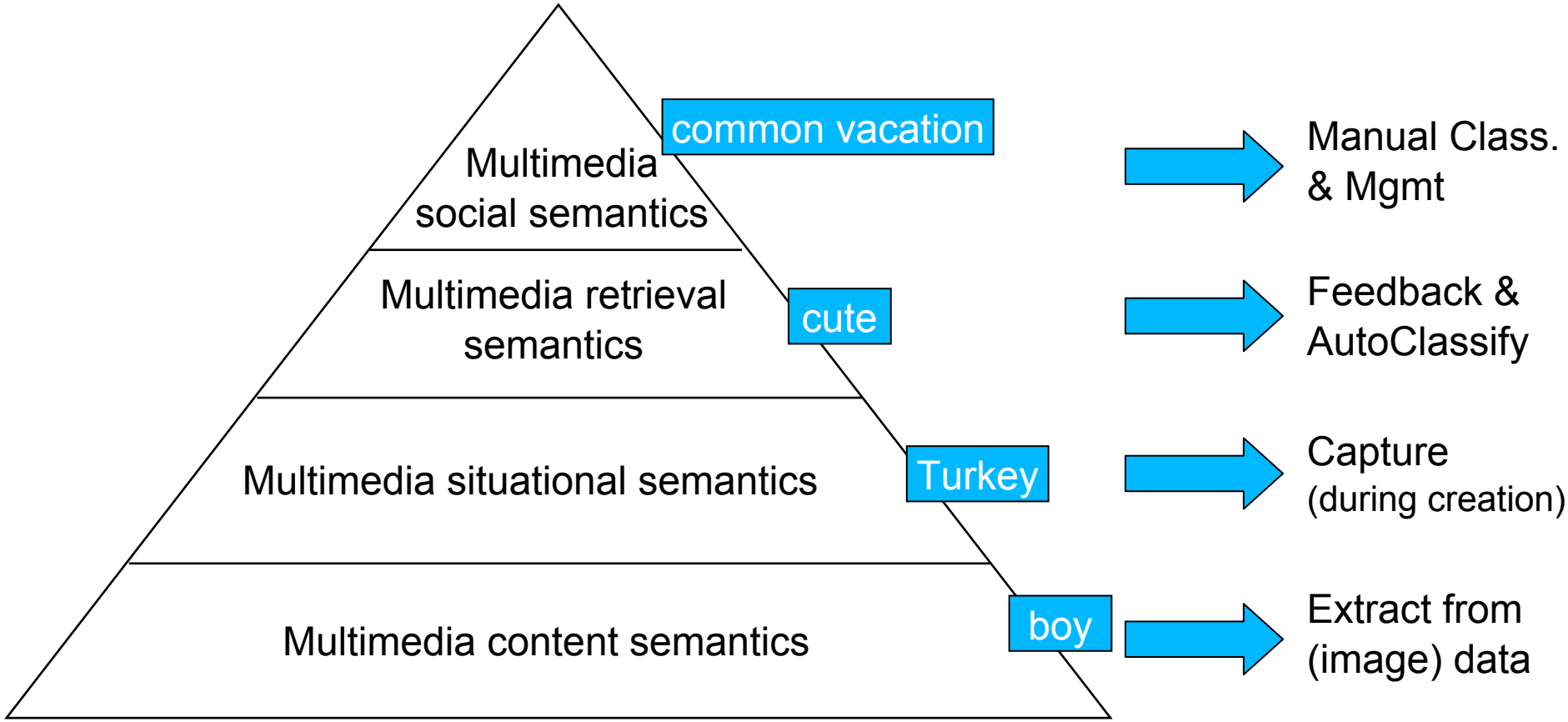
Social semantics:
SSMS 2006

Multimedia Semantics



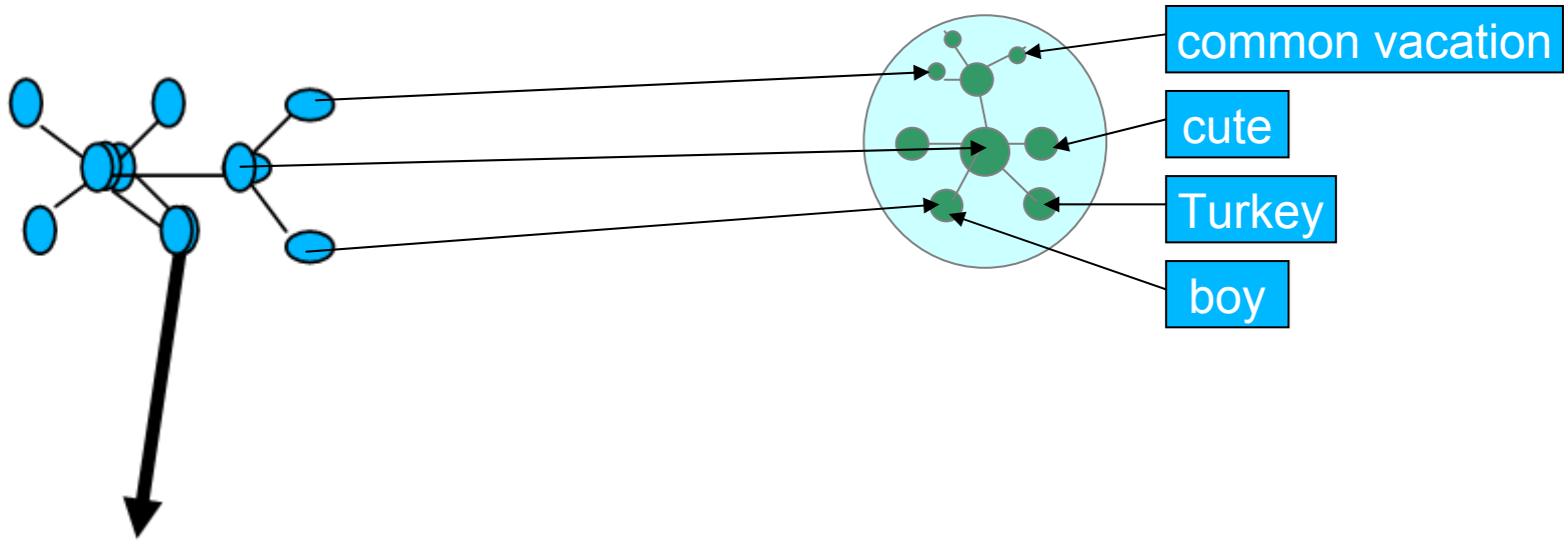
And semiotics will tell you that this list is incomplete…

Multimedia Semantics



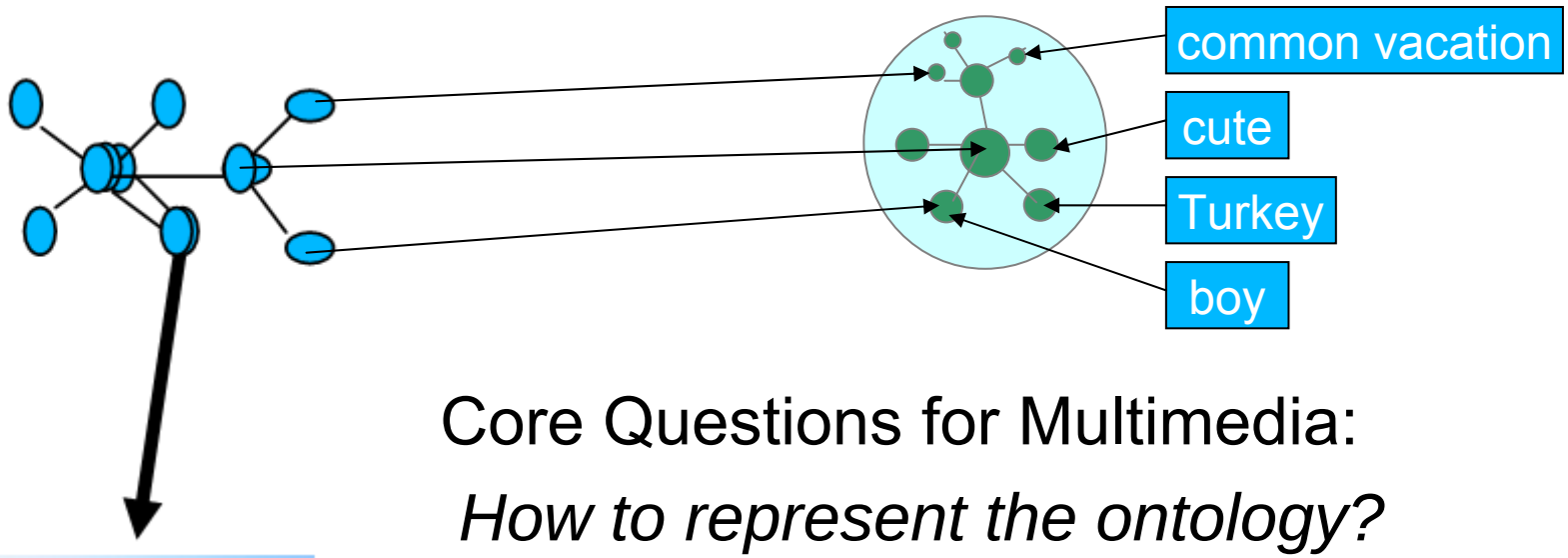
Find me a cute picture of Nico digging during our common vacation

- Manual Class. & Mgmt
- Feedback & AutoClassify
- Capture (during creation)
- Extract from (image) data



"One Ring to rule them all,
One Ring to find them,
One Ring to bring them all and
in the darkness bind them."

Inscribed on the One Ring



Core Questions for Multimedia:

How to represent the ontology?

How to query the ontology?

Which ontology to construct how?

How to populate the ontology?

Task: Depict ISWeb People

Precision

20/20

Recall

20/23 (FTE) 😐

20/43 (all) ☹

Some are missing…you may find on Flickr

Some are missing..

Students

- You may find on our group web page

Steffen Staab
staab@uni-koblenz.de

Some are missing..

Students you may find by Wiki + social search:

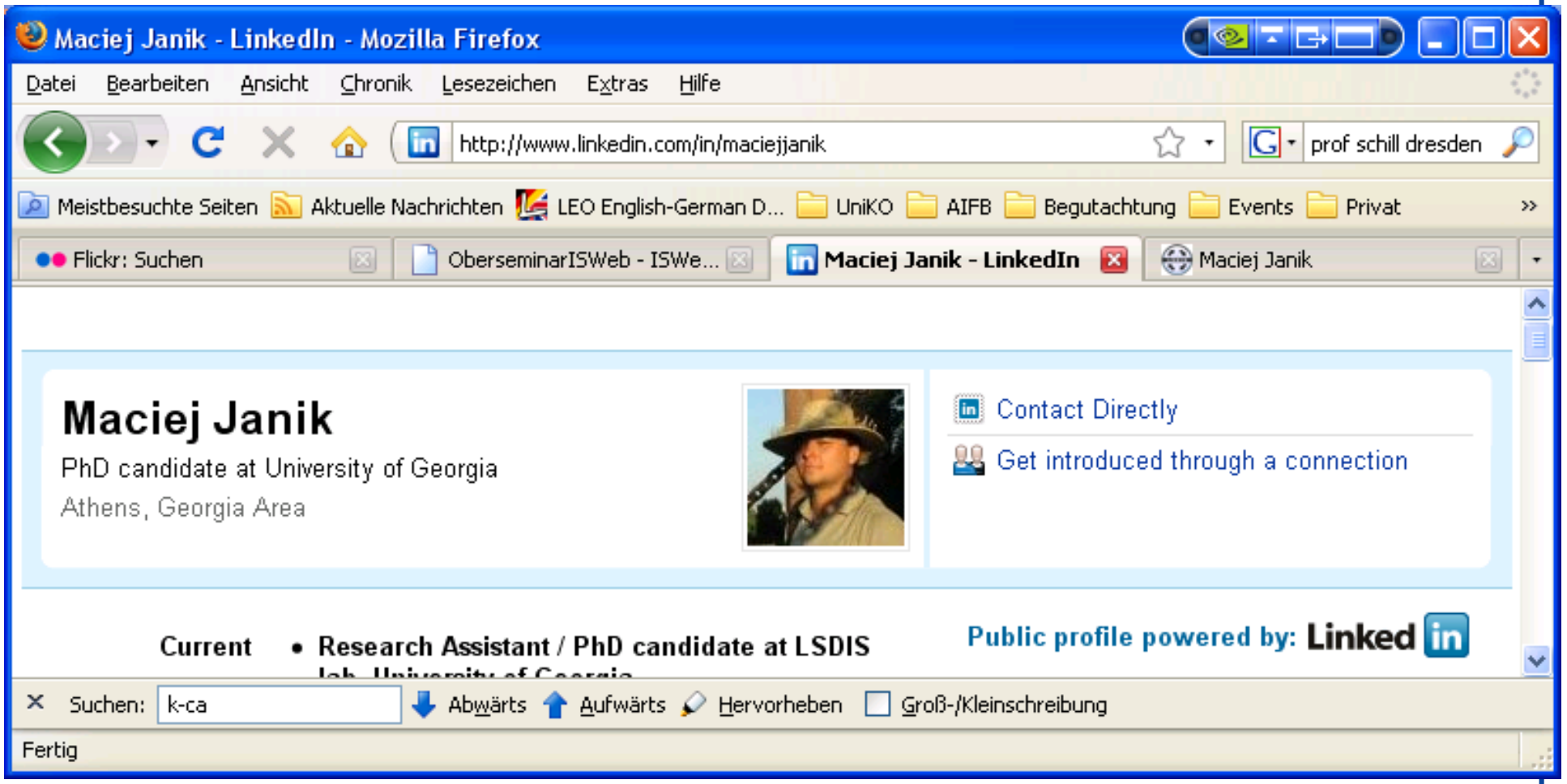
UNIVERSITÄT KOBLENZ · LANDAU

Some are missing

New members I may find in my mail box:

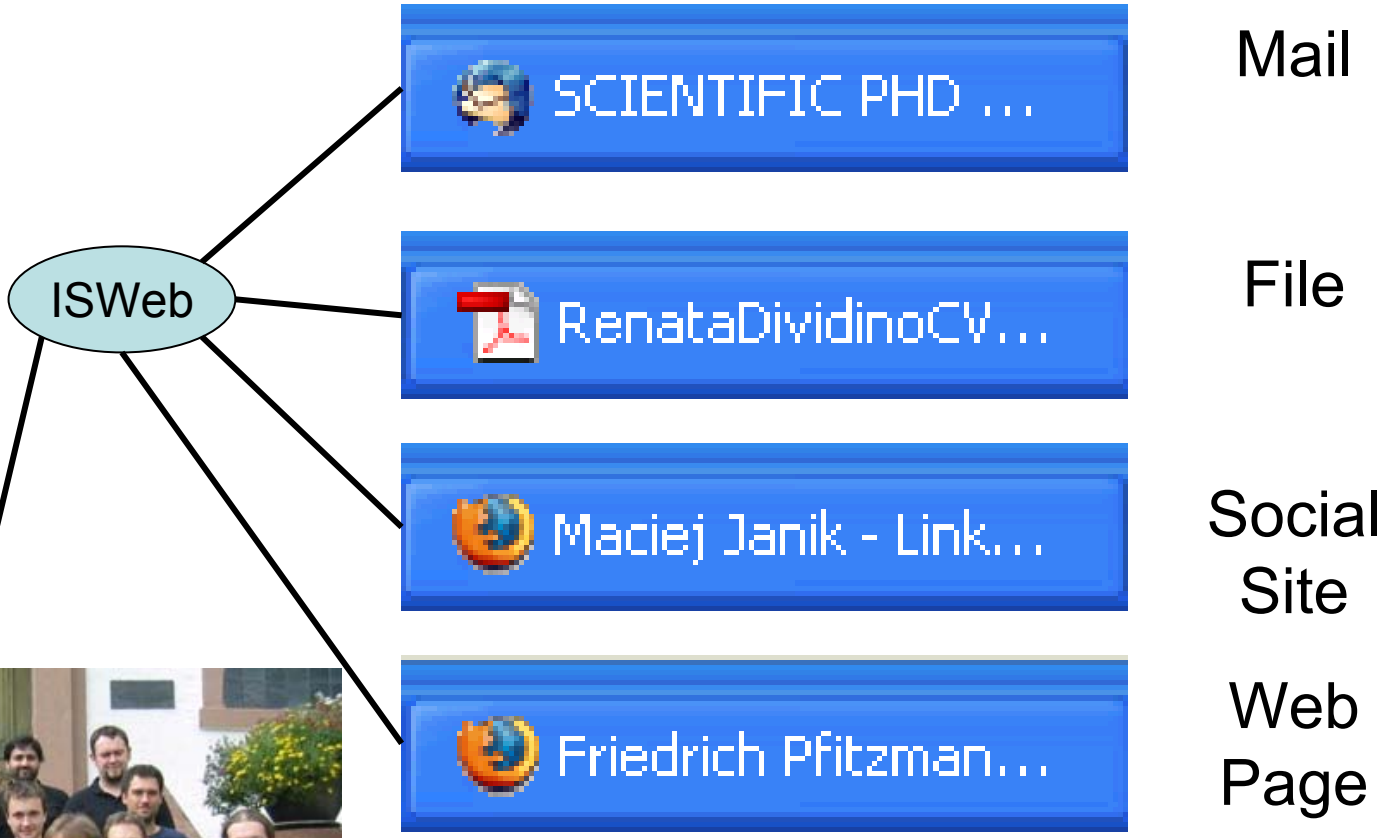
Some are missing…

New members you may find on LinkedIn:



The Semantic Web: Tying it together

<is web>



Demo Parallax

<is web>

http://www.vimeo.com/1513562?pg=embed&sec=1513562

How to represent Multimedia Data?

- Language
- Schema/Ontology/Standard

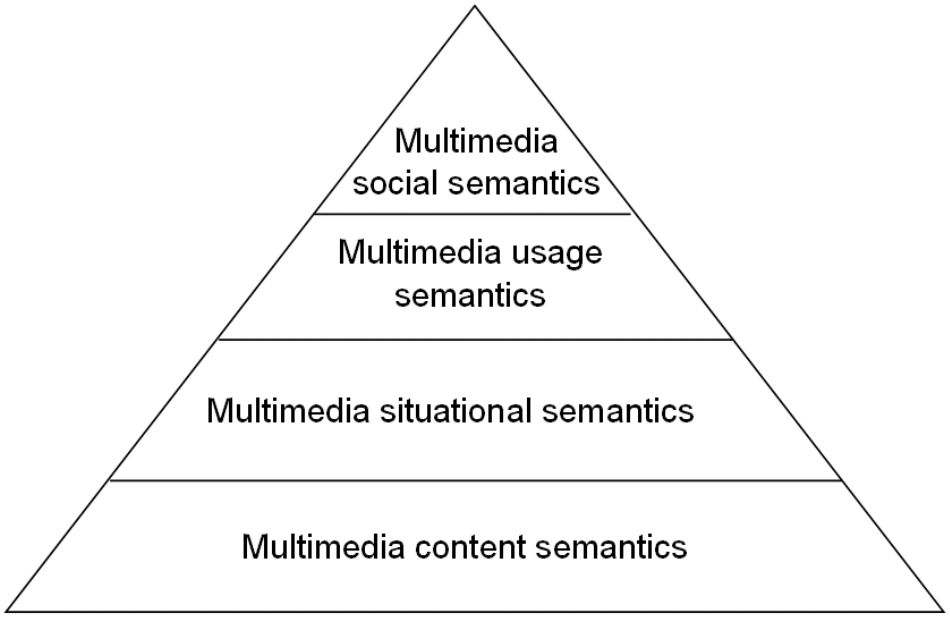
How to represent and access Multimedia Containers?

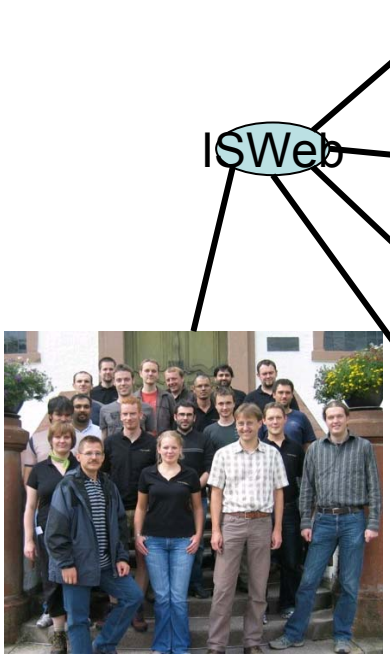
- E-mail
- Wiki
- Web pages (i.e. annotation)
- Documents
- Relational databases

How to link between different (Multimedia) Containers?

- E.g. staab@uni-koblenz.de and  refer to same person

How to query/search (Semantic) Data?





Not considered in this tutorial!

Agenda Items

Ontology Representation Languages

Ontology Query Languages

Multimedia Ontologies

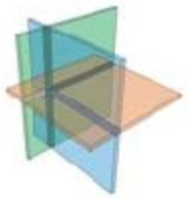
Ontology-based

- annotation
- content recognition
- content creation
- retrieval and feedback
- metadata management
- …

Semantic Web and Multimedia

- Ontologies & Their Languages –

Steffen Staab
http://isweb.uni-koblenz.de



UNIVERSITÄT KOBLENZ · LANDAU

X-MEDIA

Ontologies

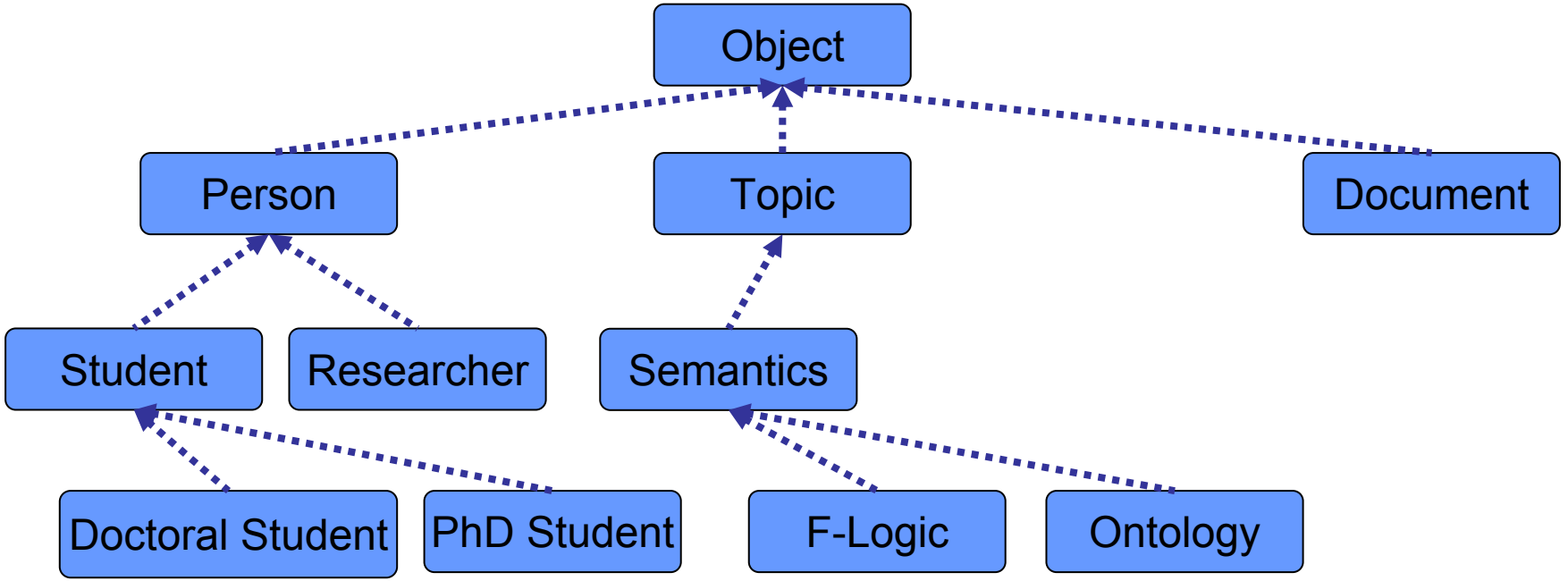
„People can‘t **share knowledge** if they do not speak a **common language**.“ [Davenport & Prusak, 1998]

Gruber 93:

An Ontology is a

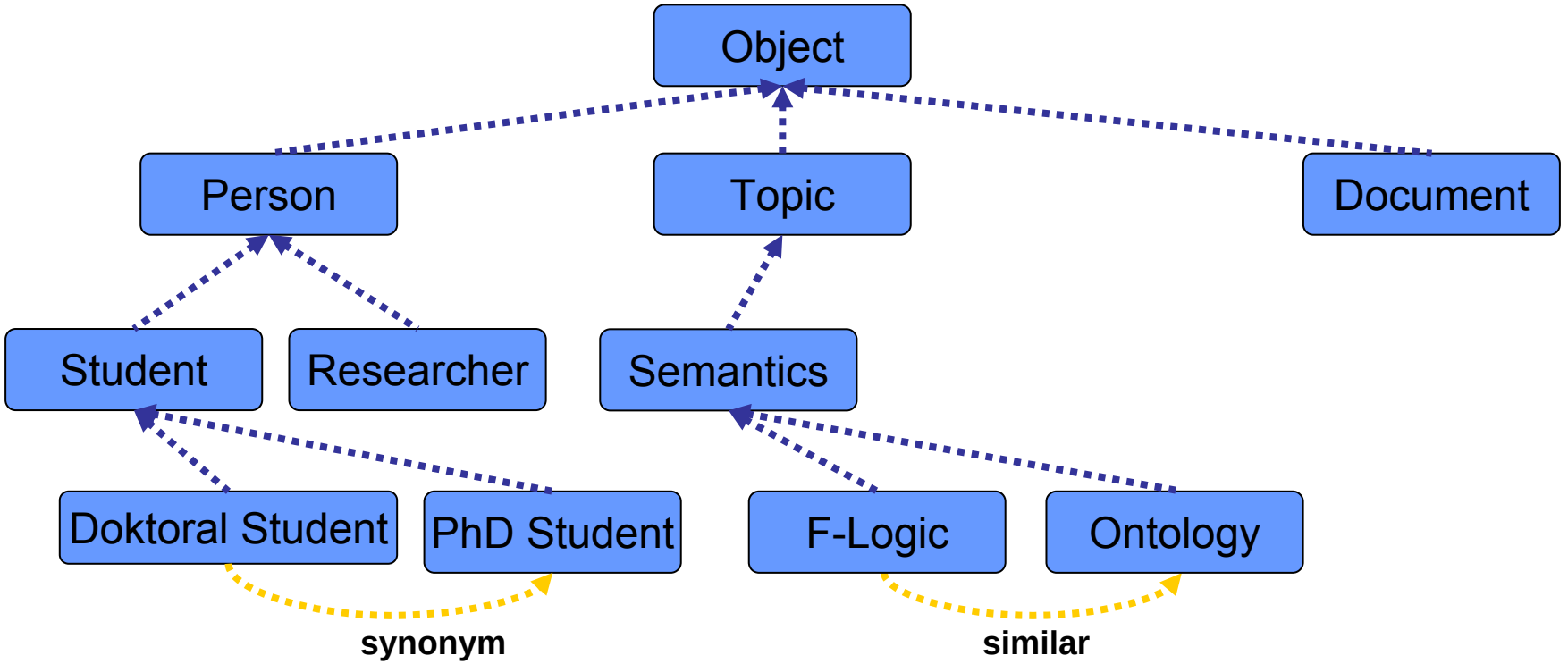
formal specification	⇒ Executable, to discuss
of a shared	⇒ group of stakeholders
conceptualization	⇒ about concepts
of a domain of interest	⇒ between application and single truth

Taxonomy



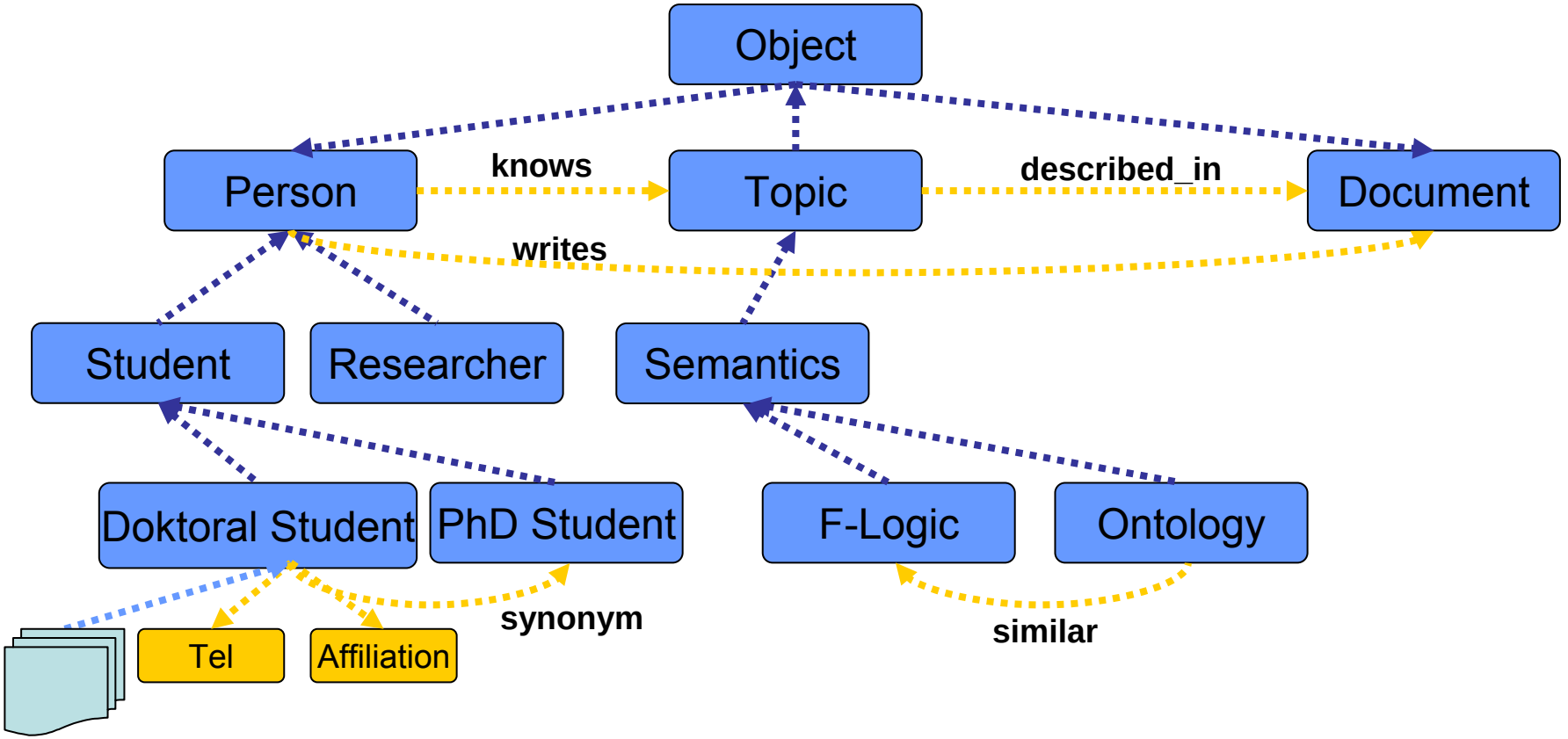
Taxonomy := Segmentation, classification and ordering of elements into a classification system according to their relationships between each other

Thesaurus



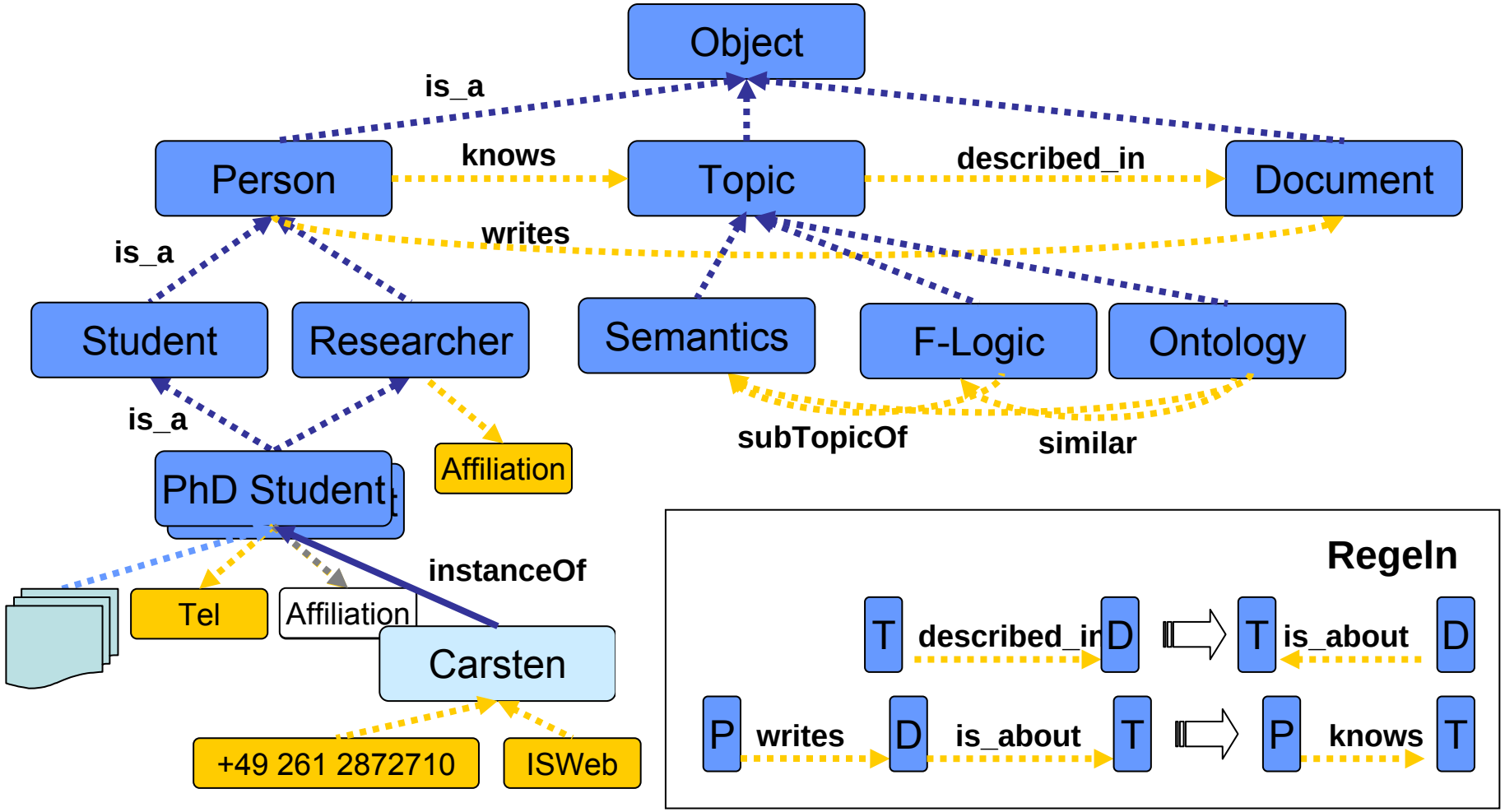
Terminology for specific domain
Taxonomy plus fixed relationships (similar, synonym, related to)
originate from library science

Topic Maps



- **Topics (nodes), relationships and *occurences* (to documents)**
- **ISO-Standard**
- **typically for navigation- and visualisation**
- **From publishing practice (back of the book index)**

Ontology



Representation Languages: Predicate Logic, Datalog, F-Logic
Standards: RDF(S); OWL

Ontologies - Some Examples

General purpose ontologies:

- DOLCE, http://www.loa-cnr.it/DOLCE.html
- The Upper Cyc Ontology, http://www.cyc.com/cyc-2-1/index.html
- IEEE Standard Upper Ontology, http://suo.ieee.org/

Multimedia Ontologies

- Acemedia harmonization effort: http://www.acemedia.org/aceMedia/reference/multimedia_ontology/

Domain and application-specific ontologies:

- GALEN, http://www.openclinical.org/prj_galen.html
- Foundational Model of Anatomy, http://sig.biostr.washington.edu/projects/fm/AboutFM.h
- Dublin Core, http://dublincore.org/

Semantic Desktop Ontologies

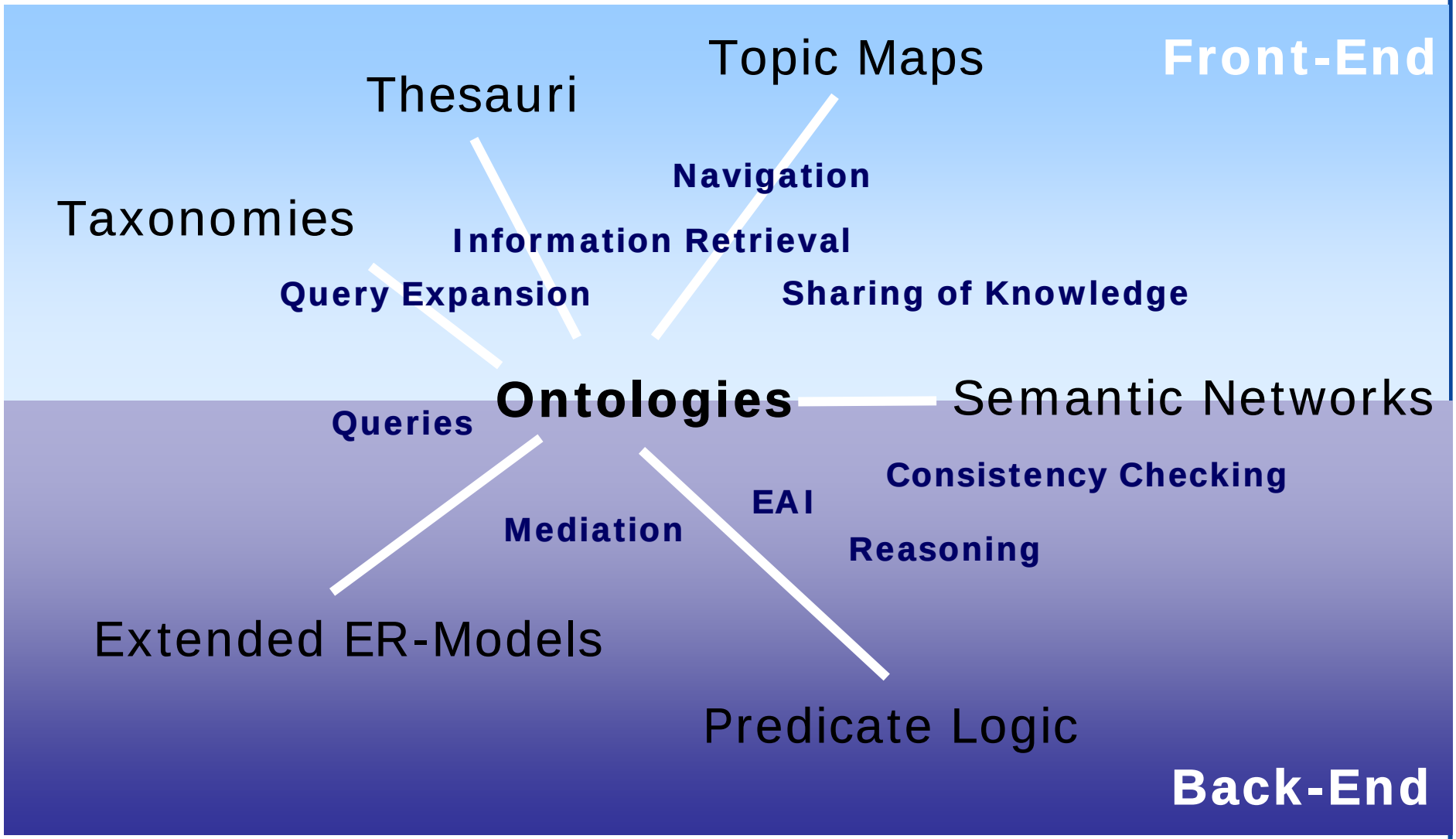
- X-COSIM Ontology, http://isweb.uni-koblenz.de/Research/X-cosim
- Haystack, http://haystack.lcs.mit.edu/, Gnowsis, http://www.gnowsis.org/, Piggybank, http://simile.mit.edu/piggy-bank/

Web Services Ontologies

- Core ontology of services http://cos.ontoware.org
- OWL-S, http://www.daml.org/services/owl-s/1.0/

Ontologies in a wider sense

- GO - Gene Ontology, http://www.geneontology.org/
- UMLS, http://www.nlm.nih.gov/research/umls/
- Agrovoc, http://www.fao.org/agrovoc/
- WordNet / EuroWordNet, http://www.cogsci.princeton.edu/~wn


Front-End
Topic Maps
Thesauri
Navigation
Taxonomies
Information Retrieval
Query Expansion
Sharing of Knowledge
Ontologies
Semantic Networks
Queries
Consistency Checking
EAI
Mediation
Reasoning
Extended ER-Models
Predicate Logic
Back-End

Ontology Trade-off

<is web>

Very formal (e.g. DOLCE)	Informal (e.g. Gene Ontology)
A lot of reasoning power	Little to no reasoning possible
Expensive to build	Comparatively inexpensive (total costs for Gene Ontology are not low!)
Misunderstandings can be corrected by expert developers (costs may be incurred)	Misunderstandings due to ambiguity are hard to correct (very high costs may be incurred!)
Ontology MUST be (at least partially) hidden from its users	Ontology may appeal to intuition of user

RDF

Resources

- A resource is a thing you talk about (can reference)
- Resources have URI's
- RDF definitions are itself Resources (linkage)

Properties

- slots, defines relationship to other resources or atomic values

Statements

- "Resource has Property with Value"
- (Values can be resources or atomic XML data)

Similar to Frame Systems

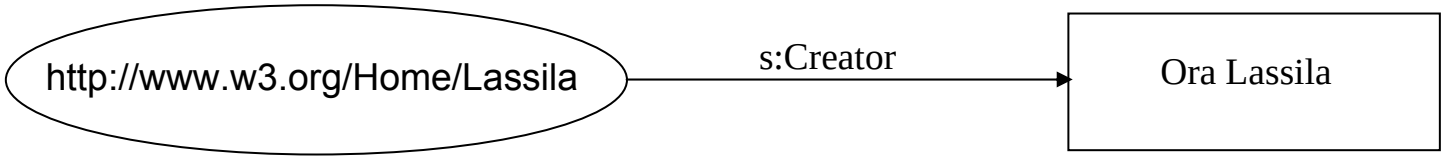
Statement

- “Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”

Structure

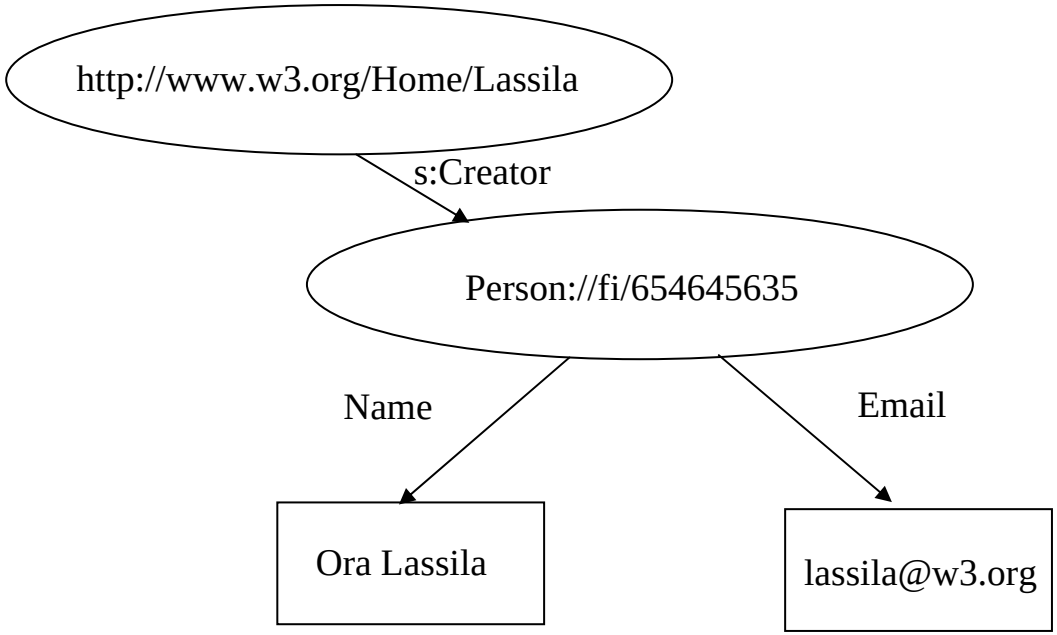
- Resource (subject) http://www.w3.org/Home/Lassila
- Property (predicate) http://www.schema.org/#Creator
- Value (object) "Ora Lassila”

Directed graph



Another Example

To add properties to Creator, point through an intermediate Resource.



Collection Containers

Multiple occurrences of the same PropertyType doesn't establish a relation between the values

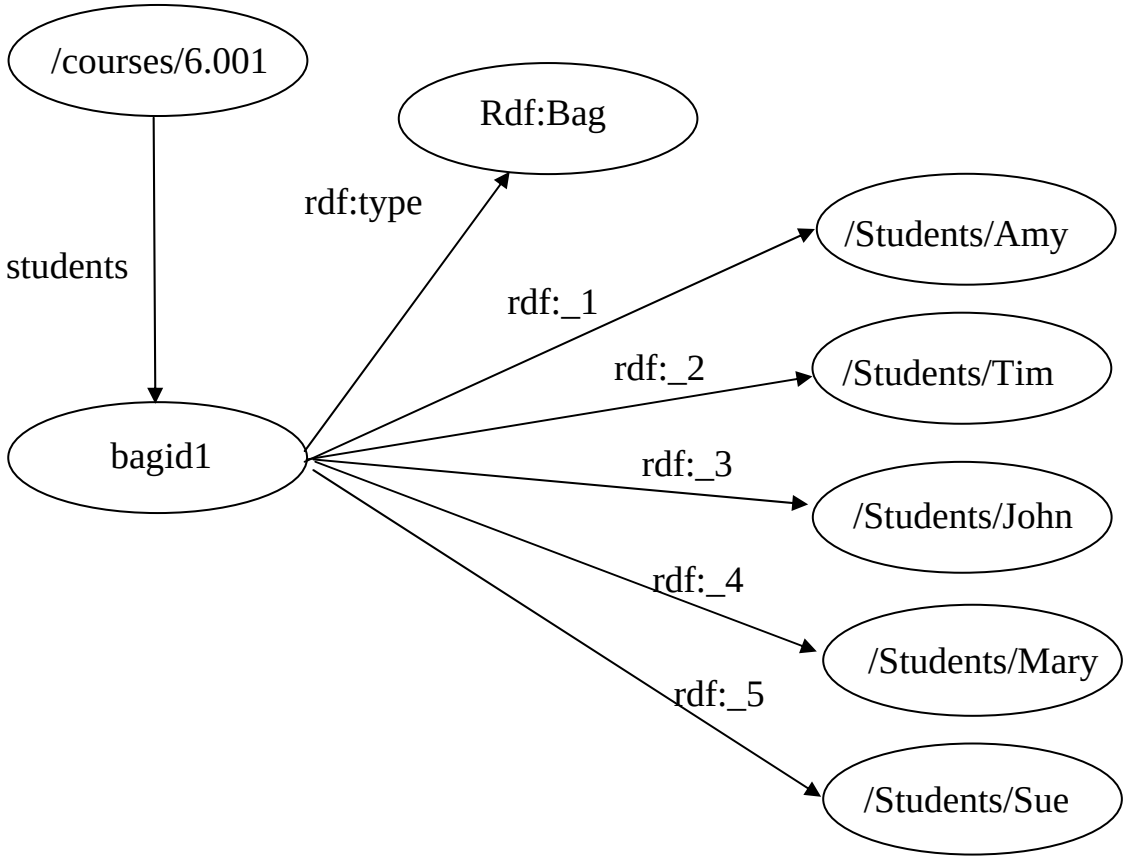
- The Millers own a boat, a bike, and a TV set
- The Millers need (a car or a truck)
- (Sarah and Bob) bought a new car

RDF defines three special Resources:

- **Bag** unordered values rdf:Bag
- **Sequence** ordered values rdf:Seq
- **Alternative** single value rdf:Alt
  - Core RDF does not enforce 'set' semantics amongst values

Example: **Bag**

The students in course 6.001 are Amy, Tim, John, Mary, and Sue

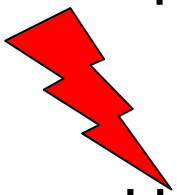


Making statements about *statements* requires a process for transforming them into Resources

- **subject** the original referent
- **predicate** the original property type
- **object** the original value
- **type** rdf:Statement

Distinguish:

• The image depicts Henry walking on water

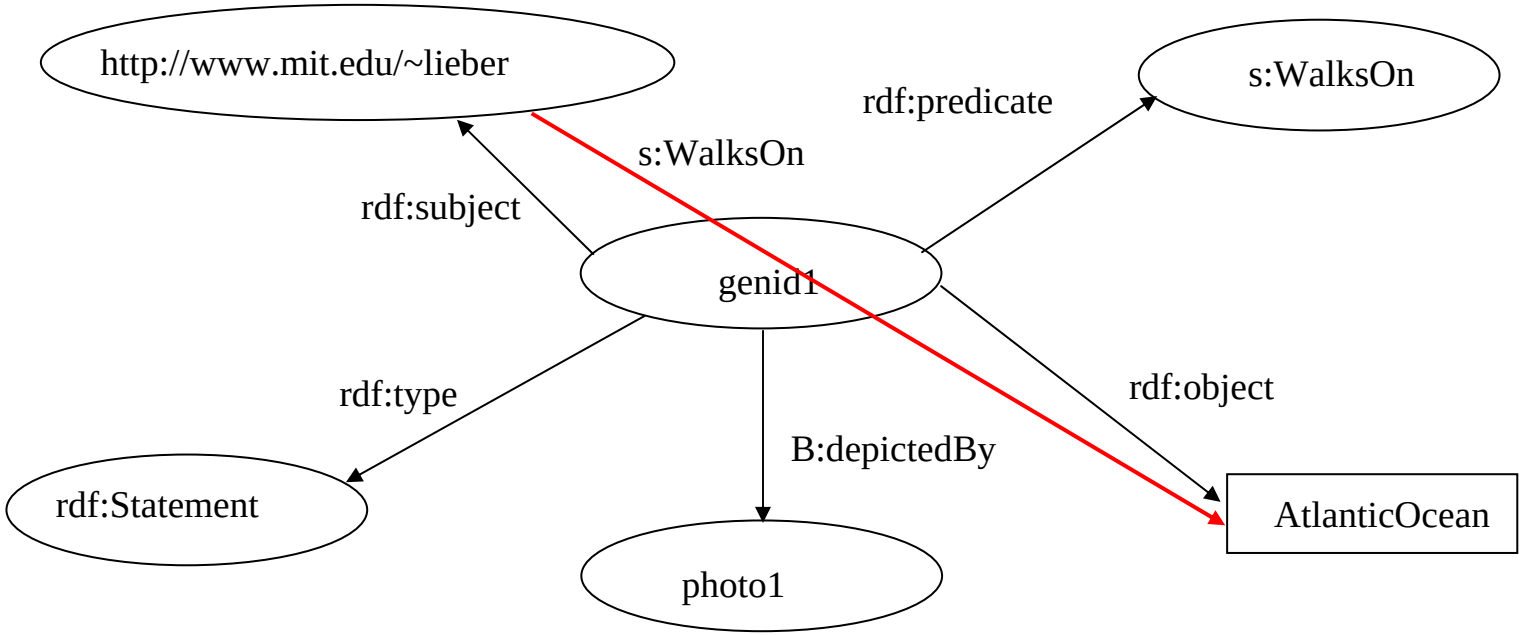


•Henry walks on water

Example: **Reification**

Photo1 depicts that

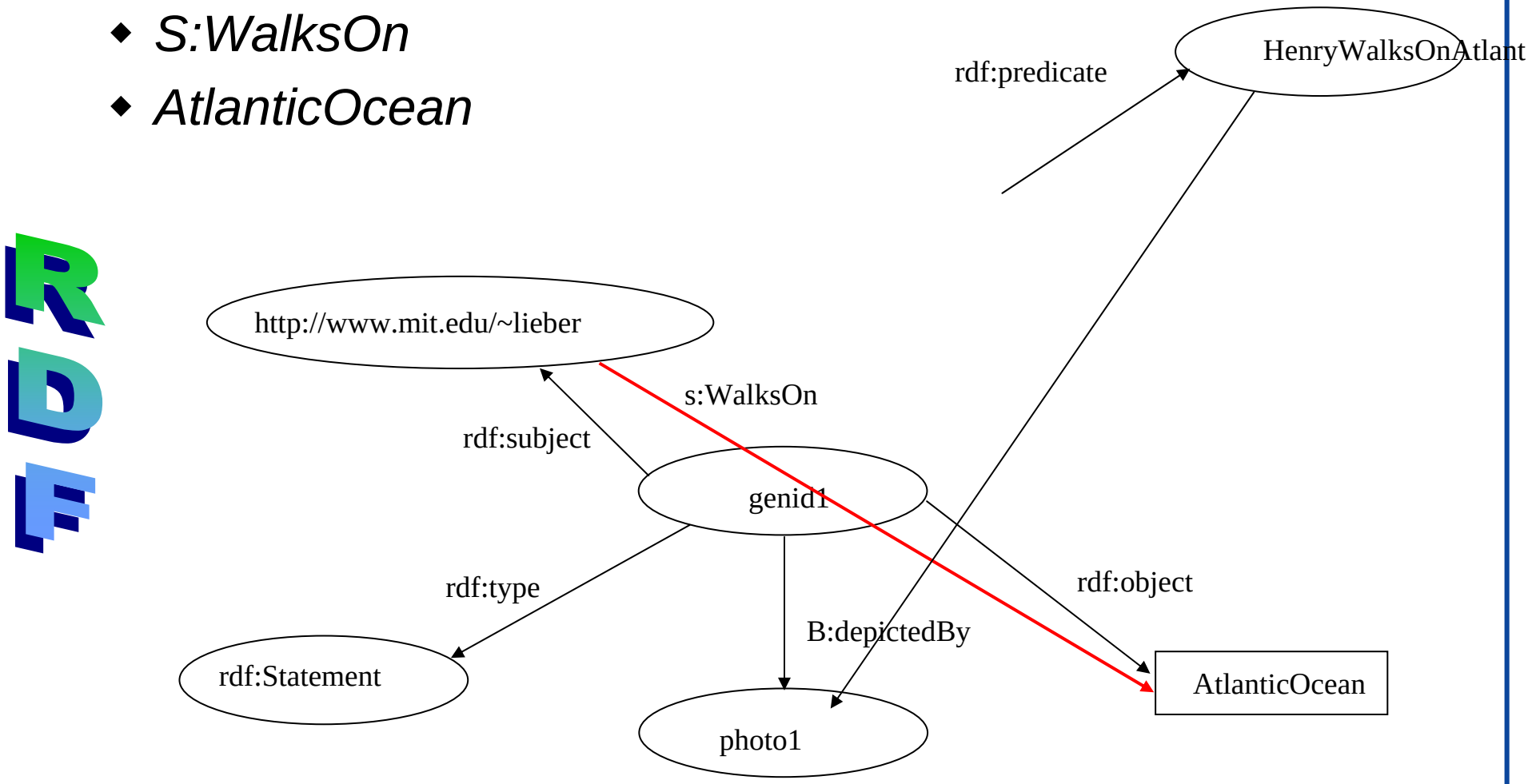
- *http://www.mit.edu/~lieber*
- *S:WalksOn*
- *AtlanticOcean*



Example: **Reification**

Photo1 depicts that

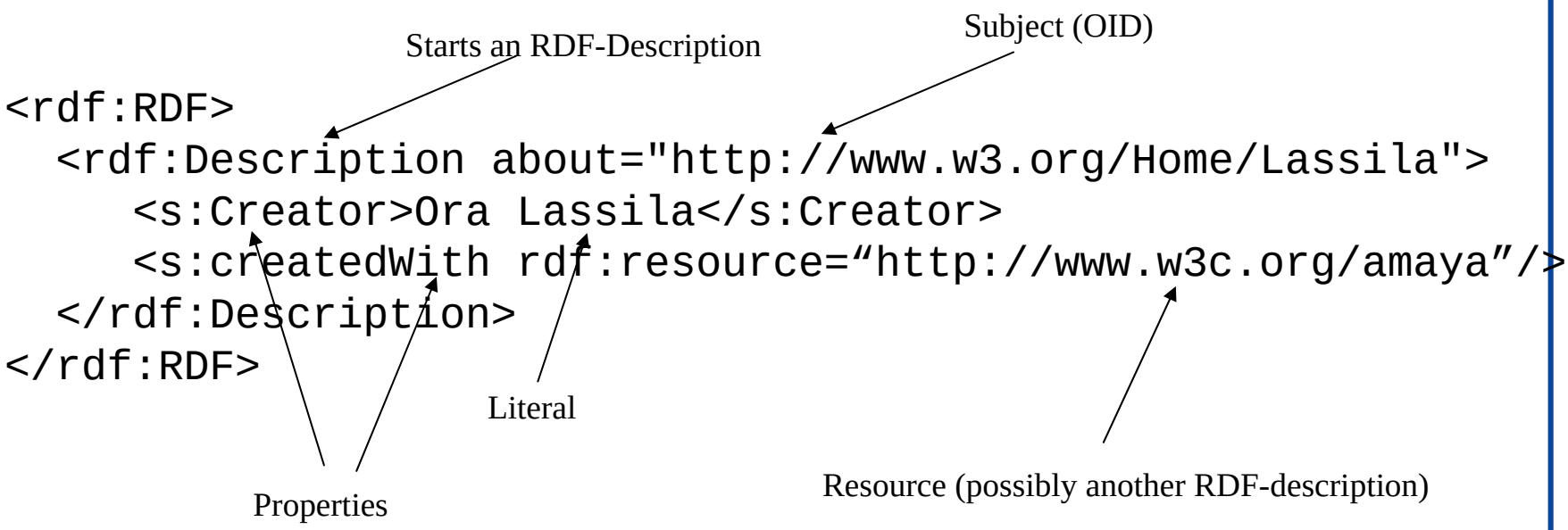
- *http://www.mit.edu/~lieber*
- *S:WalksOn*
- *AtlanticOcean*



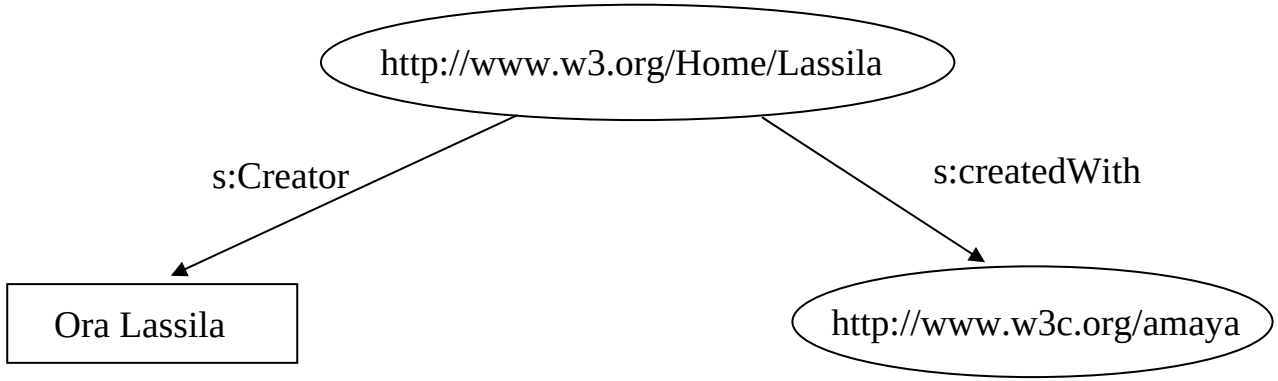
Datamodel does not enforce particular syntax

Specification suggests many different syntaxes based on XML

General form:

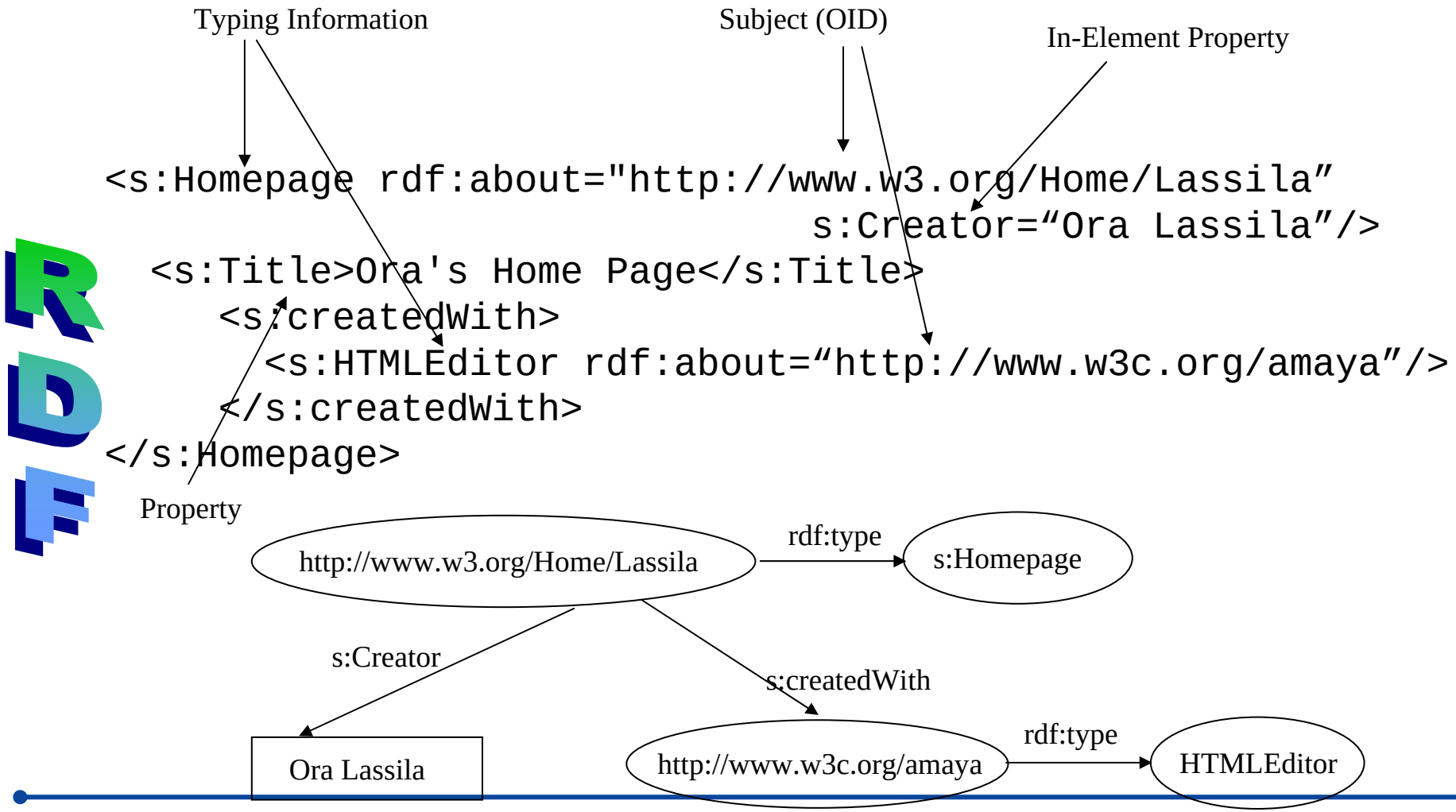


Resulting Graph




```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
     <s:Creator>Ora Lassila</s:Creator>
     <s:createdWith rdf:resource="http://www.w3c.org/amaya"/>
  </rdf:Description>
</rdf:RDF>
```

RDF Syntax II: Syntactic Varieties



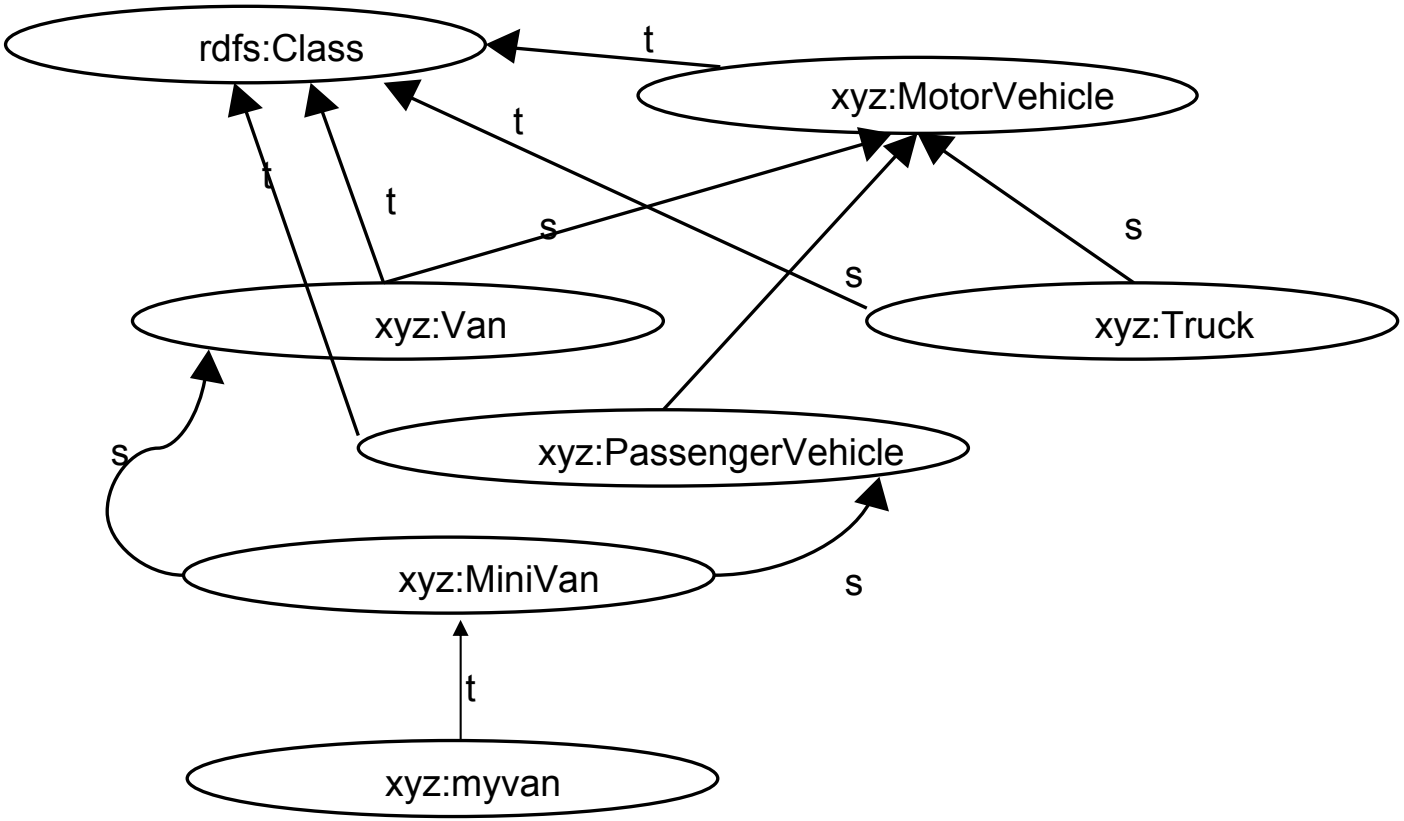
RDF just defines the datamodel

Need for definition of vocabularies for the datamodel - an Ontology Language!

RDF schemas are Web resources (and have URIs) and can be described using RDF

RDF-Schema: Example


```
s = rdfs:subClassOf
t = rdf:type
```



Rdfs:subclassOf

<is web>

```
<rdfs:description about=„Xyz:Minivan“>
      <rdfs:subclassOf about=„xyz:Van“/>
</rdfs:description>
<rdfs:description about=„myvan“>
      <rdf:type about=„xyz:MiniVan“/>
</rdfs:description>
```

Predicate Logic Consequences:

Forall X: type(X,MiniVan) -> type(X,Van).
Forall X: subclassOf(X,MiniVan) -> subclassOf(X,Van).

Rdf:property

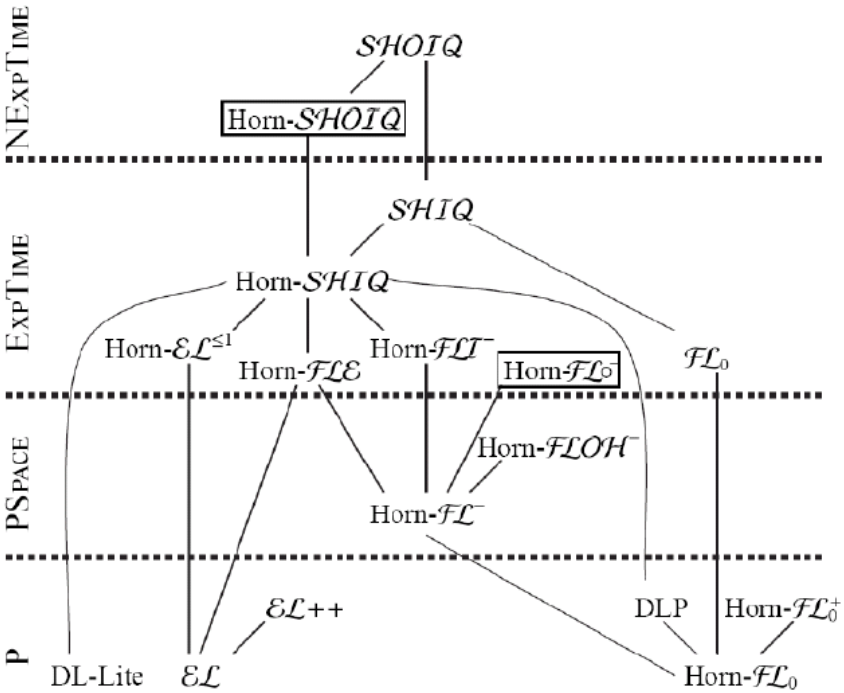
<is web>

```
<rdf:description about=„possesses“>
    <rdf:type about=„....property“/>
    <rdfs:domain about=„person“/>
    <rdfs:range about=„vehicle“/>
</rdf:description>
<rdf:description about=„peter“>
    <possesses>petersminivan</possesses>
</rdf:description>
```

Predicate Logic Consequences:

Forall X,Y: possesses (X,Y) -> (type(X,person) & type(Y,vehicle)).

OWL



W3C Recommendation since 2004

More work on OWL2 to come

Semantic fragment of FOL

Four variants:
OWL Lite ⊆ OWL DL ⊆ OWL2
OWL Lite ⊆ OWL DL ⊆ OWL Full

RDFS is fragment of OWL Full

OWL DL is decidable

OWL DL = SHOIN(D) (description logics)

W3C-Documents contain many more details that we cannot talk about here

OWL – **Syntax and semantics**

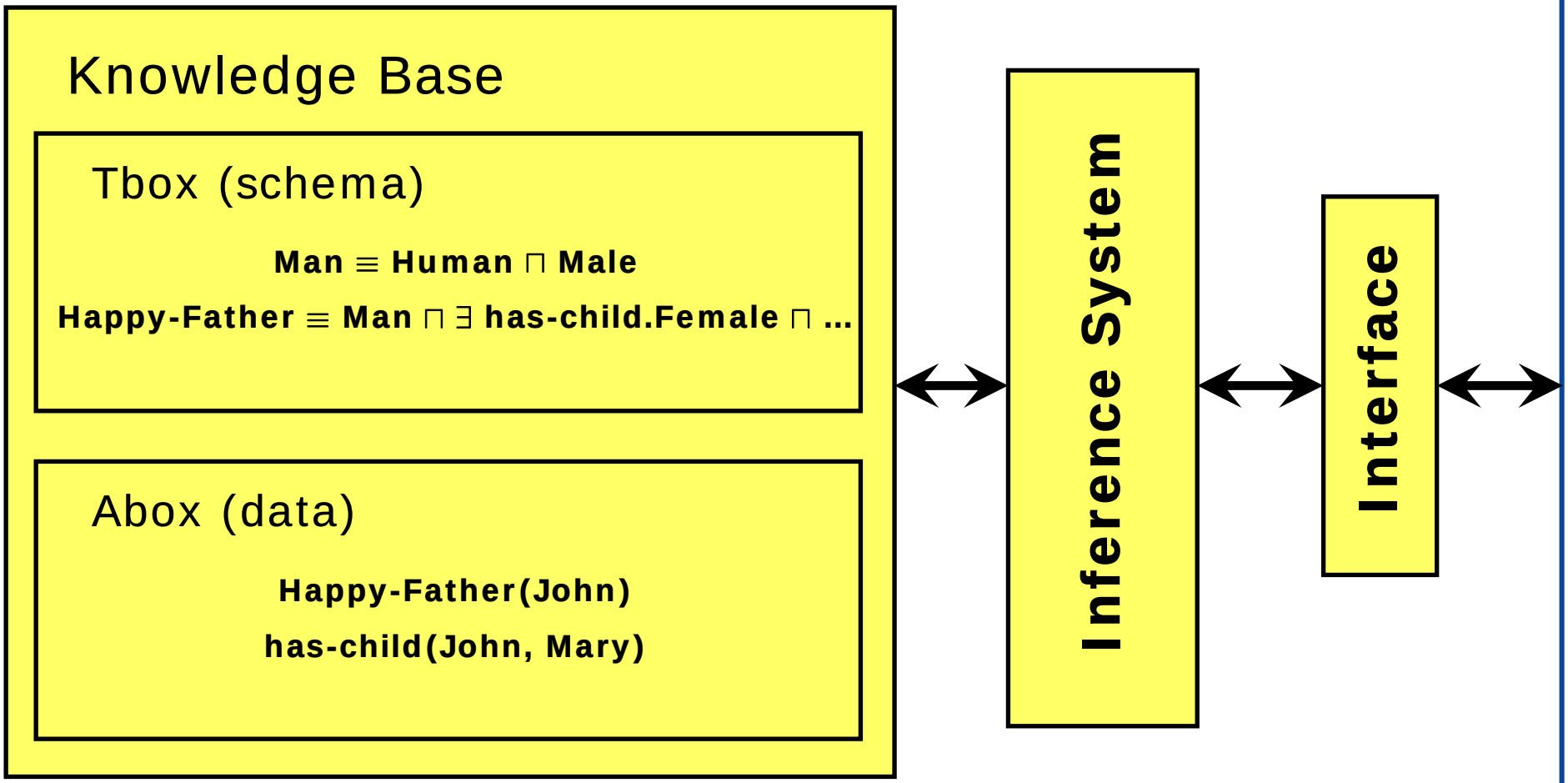
a. Description logics: SHOIN(D)

b. OWL as SHOIN(D)

c. Serializations

d. Knowledge modelling in OWL

General DL Architecture



- Sometimes: „TBox“ is equated with „Ontology
- Sometimes: „Knowledge Base“ is equated with Ontology
- My preference: „Ontology“ is everything in KB that is constant in all worlds possible in the given domain → Find out what the other person wants to say

DLs – general structure

DLs are a Family of logic-based formalism for knowledge representation

Special language characterized by:

- Constructors to define complex concepts and roles based on simpler ones.
- Set of axiom to express facts using concepts, roles and individuals.

ALC is the smallest DL, which is propositionally closed:

- constructors are noted by $\sqcap$, $\sqcup$, $\neg$ (intersection, union, negation)
- Quantors define how roles are to be interpreted:

$$\text{Man} \sqcap \exists \text{hasChild.Female} \sqcap \exists \text{hasChild.Male} \sqcap \forall \text{hasChild.}(\text{Rich} \sqcup \text{Happy})$$

Further DL concepts and role constructors

Number restrictions (cardinality constraints) for roles:

$\geq$3 hasChild, $\leq$1hasMother

Qualified number restrictions:

$\geq$2 hasChild.Female, $\leq$1 hasParent.Male

Nominals (definition by extension):

{Italy, France, Spain}

Concrete domains (datatypes): hasAge.($\geq$21)

Inverse roles: $\text{hasChild}^{-} \equiv \text{hasParent}$

Transitive roles: hasAncestor* (descendant)

Role composition: hasParent.hasBrother (uncle)

DL Knowledge Bases

DL Knowledge Bases consist of two parts (in general):

- TBox: Axioms, describing the structure of a modelled domain (conceptual schema):
  - HappyFather ≡ Man ⊓ ∃hasChild.Female ⊓ …
  - Elephant ⊑ Animal ⊓ Large ⊓ Grey
  - transitive(hasAncestor)

- Abox: Axioms describing concrete situations (data, facts):
  - HappyFather(John)
  - hasChild(John, Mary)

The distinction between TBox/ABox does not have a deep logical distinction
… but it is common useful modelling practice.

OWL – **Syntax and semantics**

a. **Description logics: SHOIN(D)**
b. **OWL as SHOIN(D)**
c. **Serializations**
d. **Knowledge modelling in OWL**

OWL DL as DL: Class constructors

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	$\neg$Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	$\leqslant$1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	$\geqslant$2hasChild	$\exists^{\geqslant n} y.P(x,y)$

Nesting of expression is allowed at arbitrary depth:

Person $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

OWL DL as DL: 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	$\top \sqsubseteq \leqslant 1P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$

General Class Inclusion ($\sqsubseteq$):

C $\equiv$ D IFF (C $\sqsubseteq$ D und D $\sqsubseteq$ C)

Obvious equivalances with FOL:

C $\equiv$ D $\Leftrightarrow$ ($\forall$x) (C(x) $\leftrightarrow$ D(x))

C $\sqsubseteq$ D $\Leftrightarrow$ ($\forall$x) (C(x) $\rightarrow$ D(x))

OWL/DL Example

Terminological Knowledge (*TBox*):

$\text{Human} \sqsubseteq \exists \text{hasParent.Human}$

$\text{Orphan} \equiv \text{Human} \sqcap \neg \exists \text{childOf.Alive}$

Knowledge about Individuals (*ABox*):

Orphan(harrypotter)

ParentOf(jamespotter,harrypotter)

Semantics and logical consequences may be derived by translation to FOL

Model theoretical Semantics – direct

Concept expressions	
A	Subset of Δ^I
$\neg C$	$\Delta^I \setminus C^I$
$C \sqcap D$	$\{x \mid x \in C^I \text{ and } x \in D^I\}$
$C \sqcup D$	$\{x \mid x \in C^I \text{ or } x \in D^I\}$
$\exists R.C$	$\{x \mid (x,y) \in R^I \text{ and } y \in C^I\}$
$\forall R.C$	$\{x \mid \text{if } (x,y) \in R^I \text{ then } y \in C^I\}$
$\geq n\ R.C$	$\{x \mid \#\{(x,y) \in R^I \text{ and } y \in C^I\} \geq n\}$
$\leq n\ R.C$	$\{x \mid \#\{(x,y) \in R^I \text{ and } y \in C^I\} \leq n\}$
$\{i_1,\ldots,i_n\}$	$\{i_1^I,\ldots,i_n^I\}$

Role expressions	
R	Subset of $\Delta \times \Delta$
R^-	$\{(y,x) \mid (x,y) \in R^I\}$

Ontology (=Knowledge Base)

Concept Axioms (TBox)	
$C \sqsubseteq D$	$C^I \subseteq D^I$
$C \equiv D$	$C^I \equiv D^I$

Role Axioms (rarely: RBox)	
$R \sqsubseteq S$	$R^I \subseteq S^I$

Assertional Axioms (ABox)	
C(a)	$a^I \in C^I$
R(a,b)	$(a^I,b^I) \in R^I$
a = b	$a^I = b^I$
$a \neq b$	$a^I \neq b^I$

Concrete Domains

Strings and Integers (required by W3C OWL rec)

Further datatypes may be supported.

Restricted to decidable predicates over the concrete domain

Each concrete domain must be implemented separately and then included into the reasoner (weak analogy: built-ins – but no procedural semantics!)

OWL – **Syntax and model theoretic semantics**

a. **Description logics: SHOIN(D)**
b. **OWL as SHOIN(D)**
c. **Serializations**
d. **Knowledge modelling in OWL**

OWL RDF Syntax	W3C recommendation
OWL Abstract Syntax	W3C recommendation See next section
OWL XML Syntax	W3C document
DL Notation	widely used in scientific contexts
FOL Notation	uncommon

Example: RDF Syntax

Person ⊓ ∀hasChild.(Doctor ⊔ ∃hasChild.Doctor):

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType="collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:someValuesFrom rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Take home message: avoid RDF serializations – use existing APIs (where possible)

OWL – **Syntax and model theoretic semantics**

a. **Description logics: SHOIN(D)**
b. **OWL as SHOIN(D)**
c. **Serializations**
d. **Knowledge modelling in OWL**

Example ontology and conclusion from
http://owl.man.ac.uk/2003/why/latest/#2

Also an example for OWL Abstract Syntax.

```
Namespace(a = <http://cohse.semanticweb.org/ontologies/people#>)
Ontology(
    ObjectProperty(a:drives)
    ObjectProperty(a:eaten_by)
    ObjectProperty(a:eats inverseOf(a:eaten_by) domain(a:animal))
    …
    Class(a:adult partial annotation(rdfs:comment "Things that are adult.")
    Class(a:animal partial restriction(a:eats someValuesFrom (owl:Thing)))
    Class(a:animal_lover complete intersectionOf(restriction(a:has_pet
          minCardinality(3)) a:person))
…)
```

Class(a:bus_driver complete intersectionOf(a:person restriction(a:drives someValuesFrom (a:bus))))

$$\mathbf{bus\_driver \equiv person \sqcap \exists drives.bus}$$

Class(a:driver complete intersectionOf(a:person restriction(a:drives someValuesFrom (a:vehicle))))

$$\mathbf{driver \equiv person \sqcap \exists drives.vehicle}$$

Class(a:bus partial a:vehicle)

$$\mathbf{bus \sqsubset vehicle}$$

A bus driver is a person that drives a bus.

A bus is a vehicle.

A bus driver drives a vehicle, so must be a driver.

The subclass is inferred due to subclasses being used in existential quantification.

Knowledge modelling: examples

Class(a:driver complete intersectionOf(a:person restriction(a:drives someValuesFrom (a:vehicle))))

$$\mathbf{driver \equiv person \sqcap \exists drives.vehicle}$$

Class(a:driver partial a:adult)

$$\mathbf{driver \sqsubseteq adult}$$

Class(a:grownup complete intersectionOf(a:adult a:person))

$$\mathbf{grownup \equiv adult \sqcap person}$$

Drivers are defined as persons that drive cars (complete definition)

We also know that drivers are adults (partial definition)

So all drivers must be adult persons (e.g. grownups)

An example of axioms being used to assert additional necessary information about a class. We do not need to know that a driver is an adult in order to recognize one, but once we have recognized a driver, we know that they must be adult.

Knowledge modelling: Example

Individual(a:Walt type(a:person) value(a:has_pet a:Huey) value(a:has_pet a:Louie) value(a:has_pet a:Dewey))

Individual(a:Huey type(a:duck))

Individual(a:Dewey type(a:duck))

Individual(a:Louie type(a:duck))

DifferentIndividuals(a:Huey a:Dewey a:Louie)

Class(a:animal_lover complete intersectionOf(a:person restriction(a:has_pet minCardinality(3))))

ObjectProperty(a:has_pet domain(a:person) range(a:animal))

Walt has pets Huey, Dewey and Louie.

Huey, Dewey and Louie are all distinct individuals.

Walt has at least three pets and is thus an animal lover.

Note that in this case, we don't actually need to include person in the definition of animal lover (as the domain restriction will allow us to draw this inference).

OWA: Open World Assumption

The existence of further individuals is possible if it is not explicitly excluded.

OWL uses OWA!

CWA: Closed World Assumption

One assumes that the knowledge base contains all known individuals and all known facts.

Are all children of Bill male?

No idea, since we do not know all children of Bill.

If we assume that we know everything about Bill, then all of his children are male.

		DL answers	Prolog
child(Bill,Bob) **Man(Bob)**	**? ⊨ ∀child.Man(Bill)**	**don't know**	**yes**
≤ 1 child.T(Bill)	**? ⊨ ∀child.Man(Bill)**	**yes**	

Now we know everything about Bill's children.

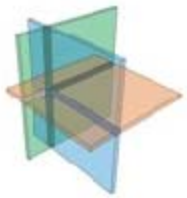
Thank You

Acknowledgements to Pascal Hitzler, York Sure@Karlsruhe for some slides on OWL

Semantic Web and Multimedia

- Query Languages –

Steffen Staab
http://isweb.uni-koblenz.de



UNIVERSITÄT KOBLENZ · LANDAU

X-MEDIA

Query Language

Requirements

Digest RDF

Digest OWL

Precise queries

Conjunctive queries

Similarity querying

…

„Standards“:

Sparql for RDF

Approaches:

OWL QL for OWL [Fikes]

Conjunctive queries for DL

More recent work by [Parsia et al07], [Kubias et al, 07]

Recommendation (since early 2008)

SPARQL is a query language for getting information from such RDF graphs. It provides facilities to:

- extract information in the form of URIs, blank nodes, plain and typed literals.
- extract RDF subgraphs.
- construct new RDF graphs based on information in the queried graphs.

There is also remote access protocol: SPROT

Example queries

<is web>

Data:
<http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title>
"SPARQL Tutorial"

Query:
SELECT ?title
WHERE
{
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}

Query Result:

Triple pattern

title
"SPARQL Tutorial"

Prefixes


```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE
{
<http://example.org/book/book1> dc:title ?title
}
```

```
PREFIX dc: http://purl.org/dc/elements/1.1/
PREFIX : <http://example.org/book/>
SELECT $title
WHERE
{
:book1 dc:title $title
}
```

```
BASE <http://example.org/book/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT $title
WHERE
{
<book1> dc:title ?title
}
```

```
BASE <http://example.org/book/>
PREFIX dcore: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE
{
<book1> dcore:title ?title
}
```

Basic Graph Pattern Matching

<is web>

Data:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
 _:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox
<mailto:outlaw@example.com> .
_:b foaf:name "A. N. Other" .
_:b foaf:mbox
<mailto:other@example.com> .
```

Query:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?mbox
WHERE
{
?x foaf:name "Johnny Lee Outlaw" .
?x foaf:mbox ?mbox
}
```

Query Result:

mbox
<mailto:outlaw@example.com>

Optional Pattern Matching

Data:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
_:a rdf:type foaf:Person .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.com> .
_:a foaf:mbox <mailto:alice@work.example> .
_:b rdf:type foaf:Person .
_:b foaf:name "Bob" .
```

Query:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{
        ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox }
}
```

Query Result:

name	mbox
„Alice“	<mailto:alice@example.com>
„Alice“	<mailto:alice@work.example>
„Bob“	

Constructing an Output Graph

Data:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:givenname "Alice" .
_:a foaf:family_name "Hacker" .
_:b foaf:firstname "Bob" .
_:b foaf:surname "Hacker" .
```

Query:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard:
<http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT
{
?x vcard:N _:v .
_:v vcard:givenName ?gname .
 _:v vcard:familyName ?fname
}
WHERE
{
        { ?x foaf:firstname ?gname }
UNION
        { ?x foaf:givenname ?gname } .
        { ?x foaf:surname ?fname }
UNION
        { ?x foaf:family_name ?fname } .
}
```

Result:

```
@prefix vcard:
        <http://www.w3.org/2001/vcard-rdf/3.0#> .
_:v1 vcard:N        _:x .
 _:x vcard:givenName  "Alice" .
 _:x vcard:familyName "Hacker" .
 _:v2 vcard:N        _:z .
 _:z vcard:givenName  "Bob" .
_:z vcard:familyName "Hacker" .
```

Filters

Boolean combinations

Testing for types (e.g. datatypes)

Regular expressions

Comparisons (corresponding to datatype, e.g. integer comparison)

External function texting (function named by IRI)

Literature

Kemafor Anyanwu, Angela Maduko, Amit P. Sheth: SemRank: ranking complex relationship search results on the semantic web. WWW 2005: 117-127

Nenad Stojanovic: On Analysing Query Ambiguity for Query Refinement: The Librarian Agent Approach. ER 2003: 490-505

Fikes, R.; Hayes, P.; & Horrocks, I. OWL-QL - A Language for Deductive Query Answering on the Semantic Web. Knowledge Systems Laboratory, Stanford University, Stanford, CA, 2003. ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-03-14.pdf.gz

Parsia et al. In: Proc. OWL-ED 2007 workshop

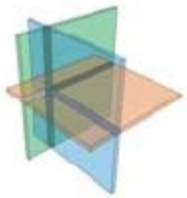
Kubias, Staab, Pan. In: Proc. OWL-ED 2007 workshop

Thank You

Semantic Web and Multimedia

- Multimedia Ontology-

Steffen Staab
http://isweb.uni-koblenz.de



UNIVERSITÄT KOBLENZ · LANDAU

X-MEDIA

"One Ring to rule them all,
One Ring to find them,
One Ring to bring them all and
in the darkness bind them."

Inscribed on the One Ring

For ~99% of multimedia people the answer for content annotation is …

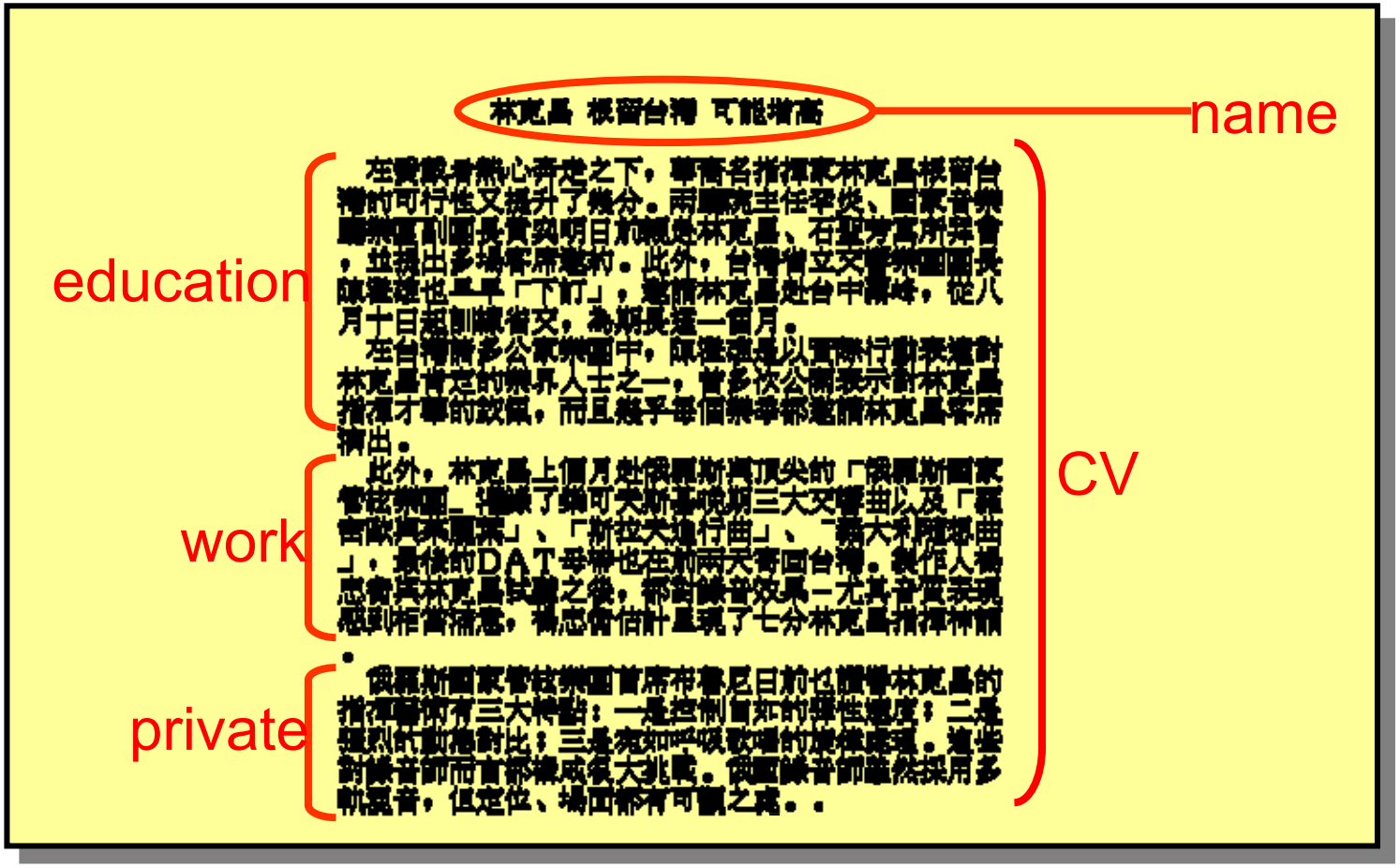
MPEG-7!

BUT: what did we learn from eCommerce?

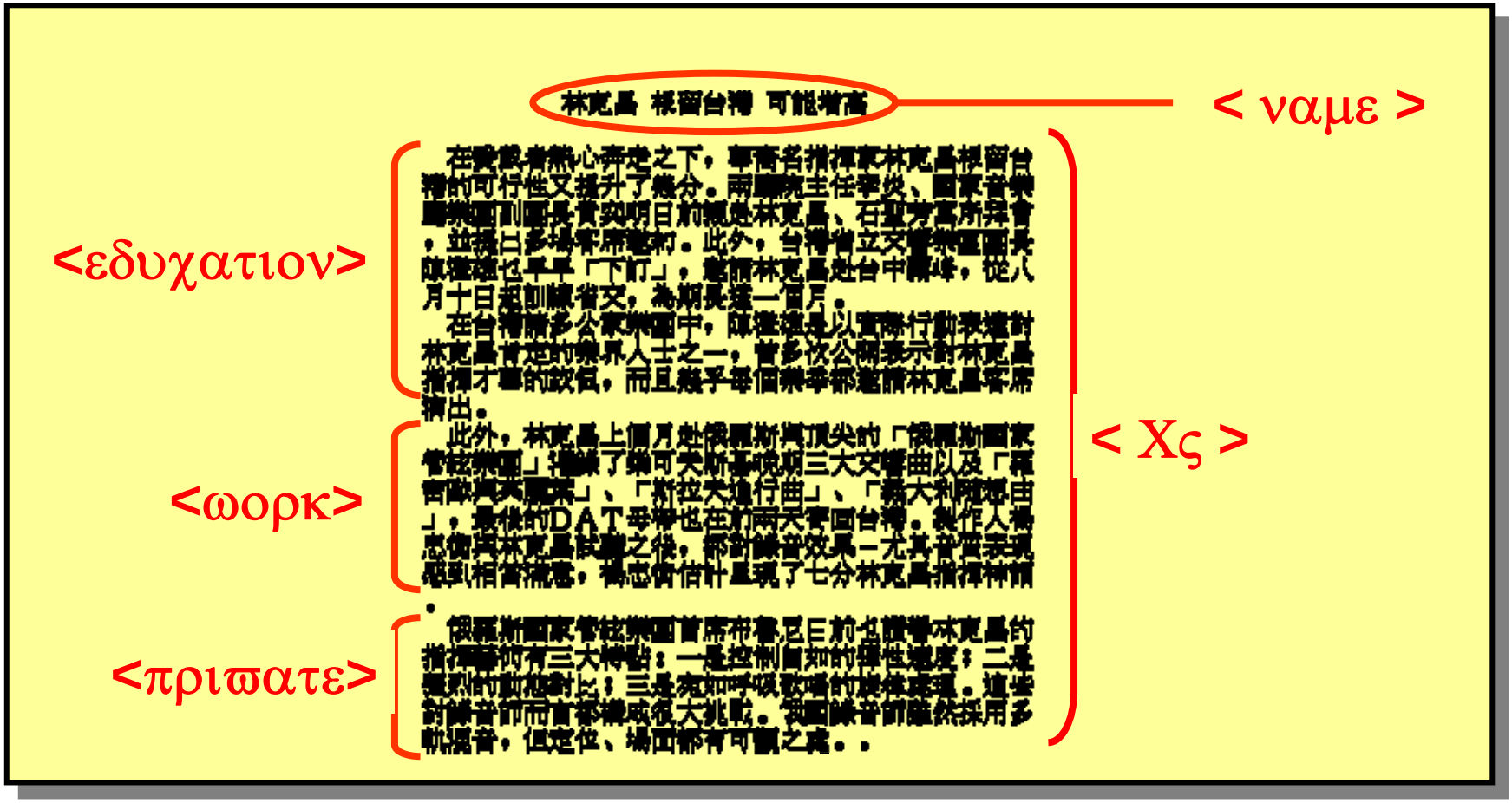
→ An XML standard is per se not the solution
for a general information integration problem!

Meaning of Informationen:

(or: what it means to be a computer)



XML ≠ Meaning, XML = Structure



What is the Problem with MPEG-7?

<is web>

Segment 1

Segment 2

Segment 3

```
<Mpeg7>
 <Description xsi:type="ContentEntityType">
  <MultimediaContent xsi:type=„ImageType">
   <Image>
    <SpatialDecomposition>
     <StillRegion id=„SR1“>
      <TextAnnotation>
       <KeywordAnnotation xml:lang="en">
        <Keyword>Sky</Keyword>
       </KeywordAnnotation>
      </TextAnnotation>
     </StillRegion>
     <StillRegion id=„SR2“>
      <Semantic>
       <Label>
        <Name>Sky</Name>
       <Label>
      </Semantic>
     </StillRegion>
     <StillRegion id=„SR3“>
      <Semantic>
       <Definition> <!-- Also TextAnnotation!! -->
        <StructuredAnnotation>
         <WhatObject>
          <Name xml:lang="en">Sky</Name>
         </WhatObject>
        </StructuredAnnotation>
       </Definition>
      </Semantic>
     </StillRegion>
     ...
```

How do you formulate a query to get all segments that show „Sky“?

First Shot:

XQL: //StillRegion[.//Keyword=“Sky”]

What is the Problem with MPEG-7?

Annotations are not interoperable:

- Ambiguities due to complementary description tools
  - ⇨***Multiple ways to model semantically identical descriptions!***
- Insufficient semantic annotations
- Several alternatives for placing description tools inside an annotation
  - ⇨***Complex queries needed to cover all alternatives!***
- MPEG-7 profiles can only partly solve interoperability problems [Bailer et al., 2005]
  - *Semantic conformance to a profile cannot be checked automatically*

Former Situation:

no standard, no vocabulary, no coupling of data and application unless by tiresome 1:1 agreement of involved parties

Current situation:

MPEG-7, tight coupling, standard vocabulary, agreement between involved parties on which vocabulary to use and its exact meaning

Future / desired situation:

loose coupling, standard vocabulary with pre-defined meaning, automatic ad-hoc coupling of data and integration

Replace MPEG-7 with a **high quality multimedia ontology** that fulfils the following requirements:

- **Reusability**
  Design a core ontology for any multimedia related application
- **MPEG-7-Compliance**
  Support most important description tools (decomposition, visual / audio descriptors, …)
- **Extensibility**
  Enable inclusion of further
  - *description tools (even those that are not part of MPEG-7!)*
  - *media types*
- **Modularity**
  Enable customization of multimedia ontology
- **High degree of axiomatization**
  Ensure interoperability through machine accessible semantics

Is MPEG-7 a good Basis for a high Quality Ontology?

Shortcomings of badly modelled ontologies [Oberle et al., 2006]:

1) **Conceptual ambiguity**
   - *Difficulties in understanding the meaning of concepts and their relations*
2) **Poor axiomatization**
   - *Axiomatization of well defined concepts is missing*
3) **Loose Design**
   - *Presence of modelling artefacts (concepts without ontological meaning)*

Shortcomings mainly hinder

- **Extensibility**
- **Interoperability**

Especially 1) and 2) are major shortcomings of MPEG-7

⇨ **1-to-1 translations from MPEG-7 to OWL/RDFS (e.g. [Hunter, 2003a]) will not result in high quality ontologies!**

Approach from [Oberle, 2005], [Oberle et al., 2006]:
Use a well designed foundational ontology as a modelling basis to avoid shortcomings

DOLCE is well suited because it provides 2 design patterns that are important for MPEG-7 (see Gangemi et al., 2005] for details)

- **Ontology of Information Objects (OIO)**:
  Formalization of information exchange
- **Descriptions & Situations (D&S)**:
  Formalization of context

Use D&S and OIO to translate MPEG-7 in the DOLCE vocabulary, **but:**

- Separate translation of each MPEG-7 description tool is not feasible!
- ⇨ **Define patterns that allow the translation of numerous description tools**

Methodology for Design Pattern Definition

Identification of most important MPEG-7 functionalities

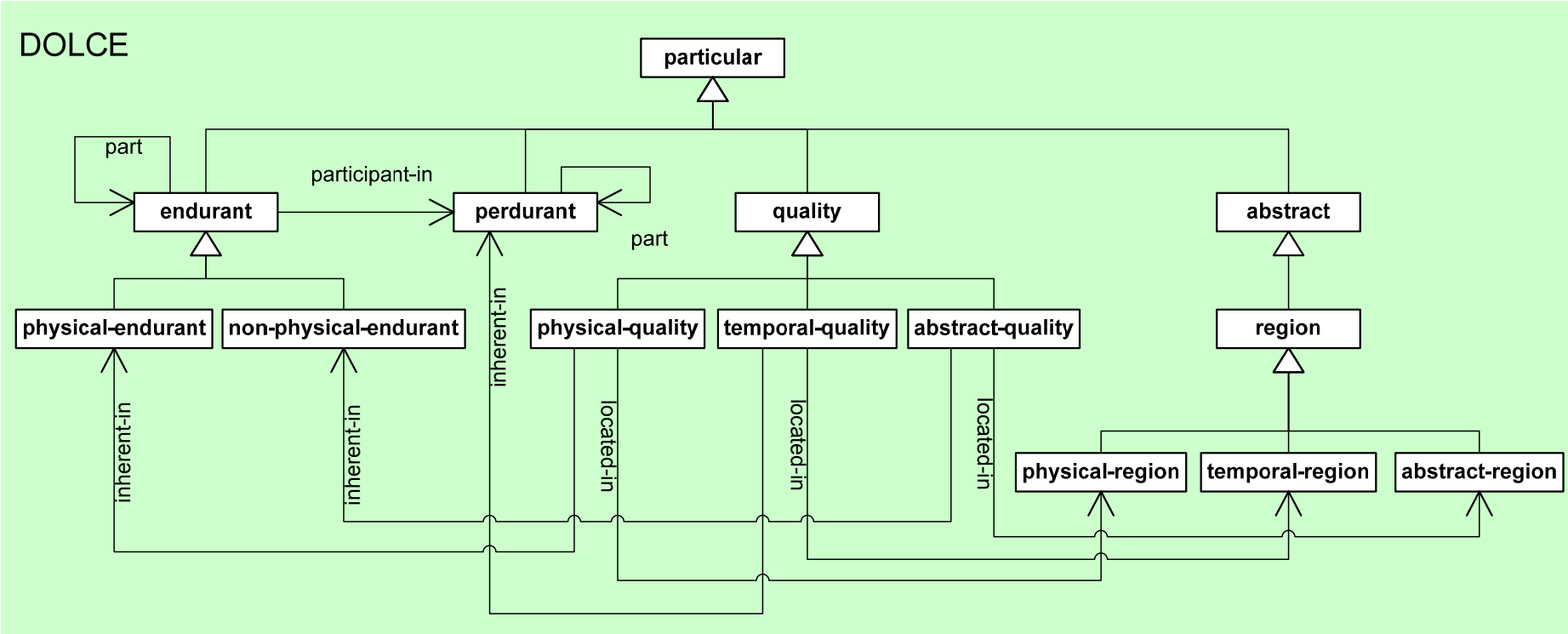
- **Decomposition** of multimedia content into segments
- **Annotation** of segments with meta data (e.g. visual descriptor, media information, creation & production, …)
- General:
  Describe digital data by digital data at an arbitrary level of granularity

Definition of **design patterns** for decomposition and annotation based on D&S and OIO

Additional patterns are needed for:

- **Complex data types** of MPEG-7
- **Semantic annotation** by using domain ontologies
  - ⇨ *Interface between reusable multimedia core and domain specific knowledge*

DOLCE Foundational Ontology



DOLCE Foundational Ontology

<is web>

4D world view centered around

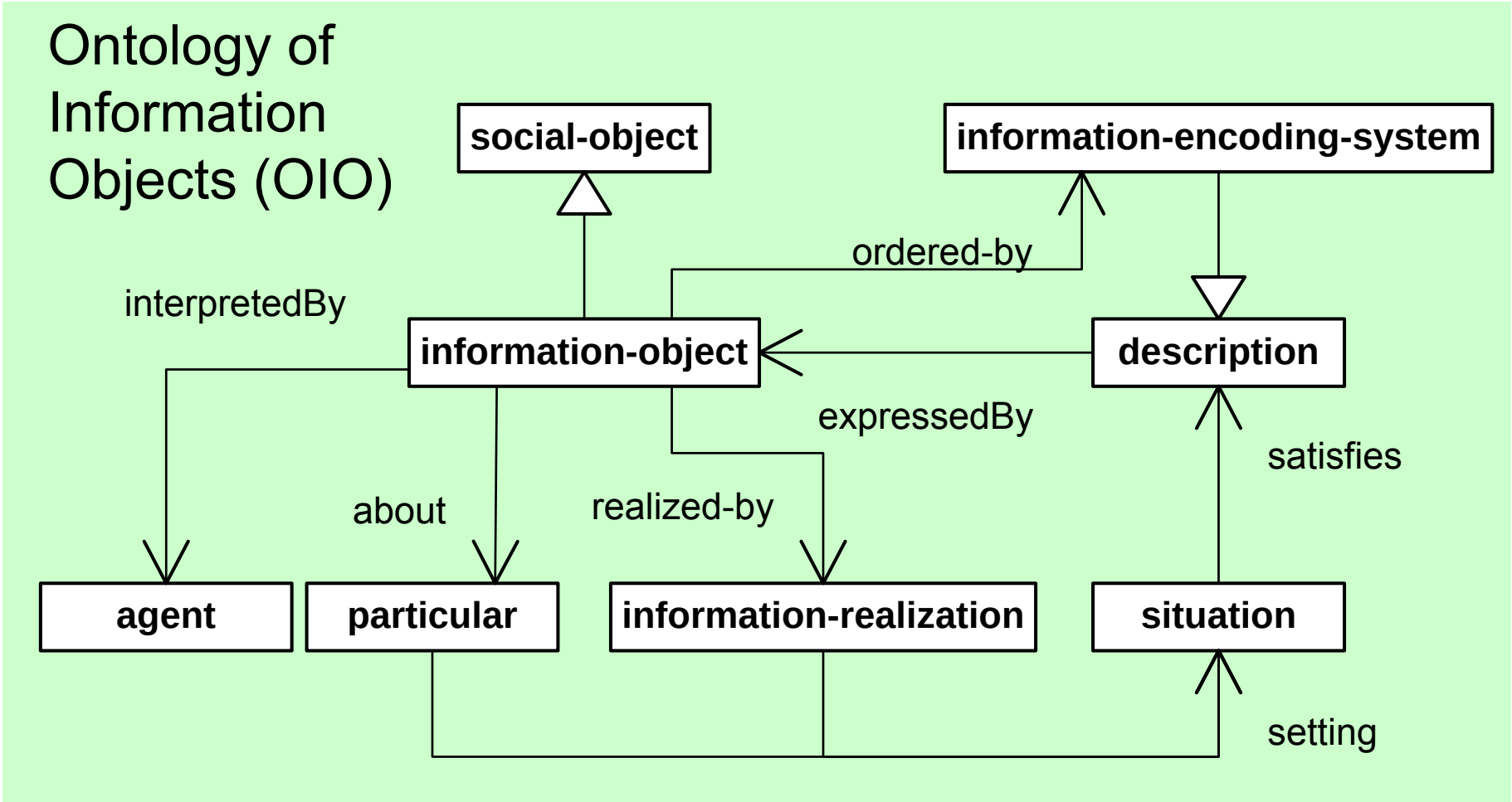
- **Endurants**: Independent wholes that exist in time and space
- **Perdurants**: Events, processes, phenomena, …

DOLCE is a library of foundational ontologies that provides 2 design patterns (extensions) that are especially important for MPEG-7:

- **Ontology of Information objects (OIO)**: Formalization of information exchange
- **Descriptions & Situations (D&S)**: Formalization of context

Use these extensions to translate the technical concepts of MPEG-7 in the DOLCE vocabulary

Ontology of Information Objects (OIO)



Example

Information Object „Secure the building“

Information Realization

83-101-99-117-114-101-32-116-104-101-32-98-117-105-108-100-105-110-103

Information Encoding: ASCII-Code decimal

About: the White House

Situation: Securing the president

Agent: 1. US Airforce / 2. US Army / 3. US Navy

Expresses:

1. Buy the building
2. Everyone out of the building, blinds shut down
3. Bomb the building

Ontology of Information Objects (OIO)

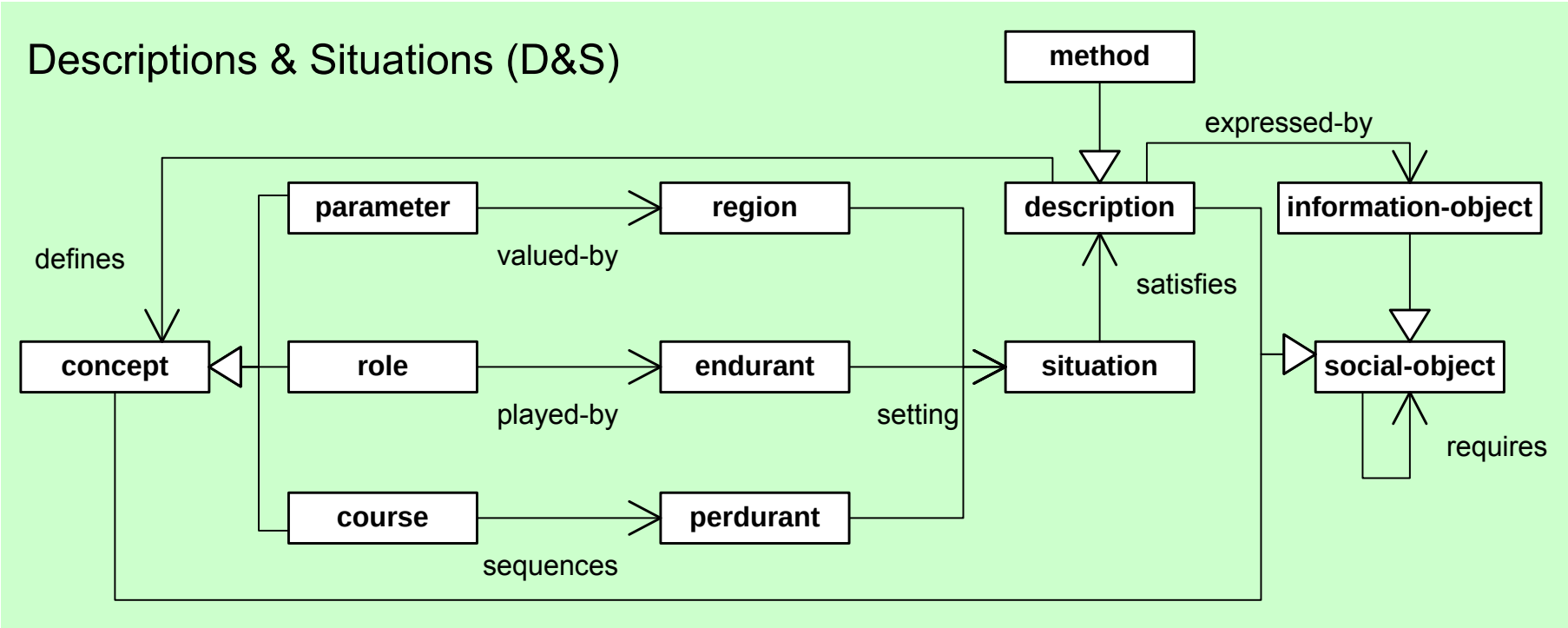
Formalization of information exchange

- Shannon's communication theory
- Communication elements by Jakobson

Information object represents pure abstract information (message)

Relevance for multimedia ontology:

- MPEG-7 describes digital data (multimedia information objects) with digital data (annotation)
- Digital data entities are information objects


Descriptions & Situations (D&S)
method
expressed-by
parameter
region
description
information-object
defines
valued-by
satisfies
concept
role
endurant
situation
social-object
played-by
setting
requires
course
perdurant
sequences

Descriptions & Situations (D&S)

Distinction between:

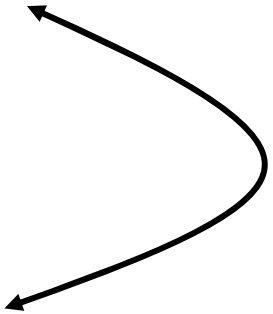
- DOLCE ground entities (regions, endurants, perdurants)
- Descriptive entities (parameters, roles, courses)

Descriptions

- Formalize context
- Define descriptive concepts

Situations

- Are explained by descriptions
- Are settings for ground entities



Don‘t confuse a situation and its description. The situation is unique, its descriptions may be conflicting!

Comparable to Reification - in a more systematic manner!

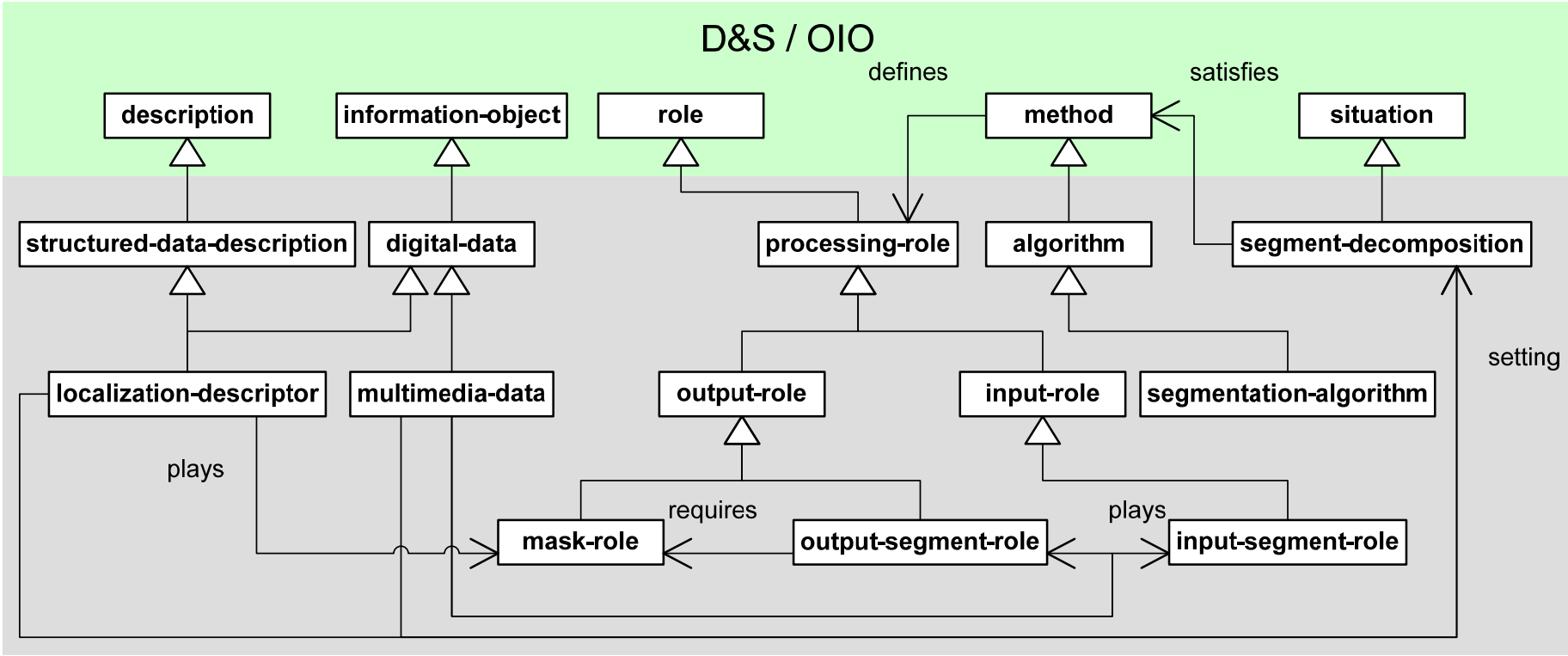
Relevance for multimedia ontology:

- Meaning of digital data depends on context
- Digital data entities are connected through computational situations (e.g. input and output data of an algorithm)
- Algorithms are descriptions
- Annotations and decompositions are situations that satisfy the rules of an algorithm / method

Benefit of DOLCE for Design of MM Ontology

Usage of DOLCE+D&S+OIO enforces clean design of the multimedia ontology

- Constraints, that are part of the axiomatization do not allow an arbitrary placement of MPEG-7 concepts into DOLCE
- Multimedia ontology will be more extensible due to the underlying general taxonomy of DOLCE (similar concepts will be placed on similar locations of the taxonomy)


D&S / OIO
defines
satisfies
description
information-object
role
method
situation
structured-data-description
digital-data
processing-role
algorithm
segment-decomposition
setting
localization-descriptor
multimedia-data
output-role
input-role
segmentation-algorithm
plays
requires
plays
mask-role
output-segment-role
input-segment-role

Example

Image1 playsRole
SegmentationInput

Segment1 playsRole SegmOutp
Segment3 playsRole SegmOutp
Segment2 playsRole SegmOutp
Segment2 playsRole SegmInp
Segment4 playsRole SegmOutp

Via its role in a computational task the different parts may be arbitrarily nested and related to different computing algorithms

Querying for all subparts takes place along a well-defined pattern

Usage of DOLCE enforces clean design

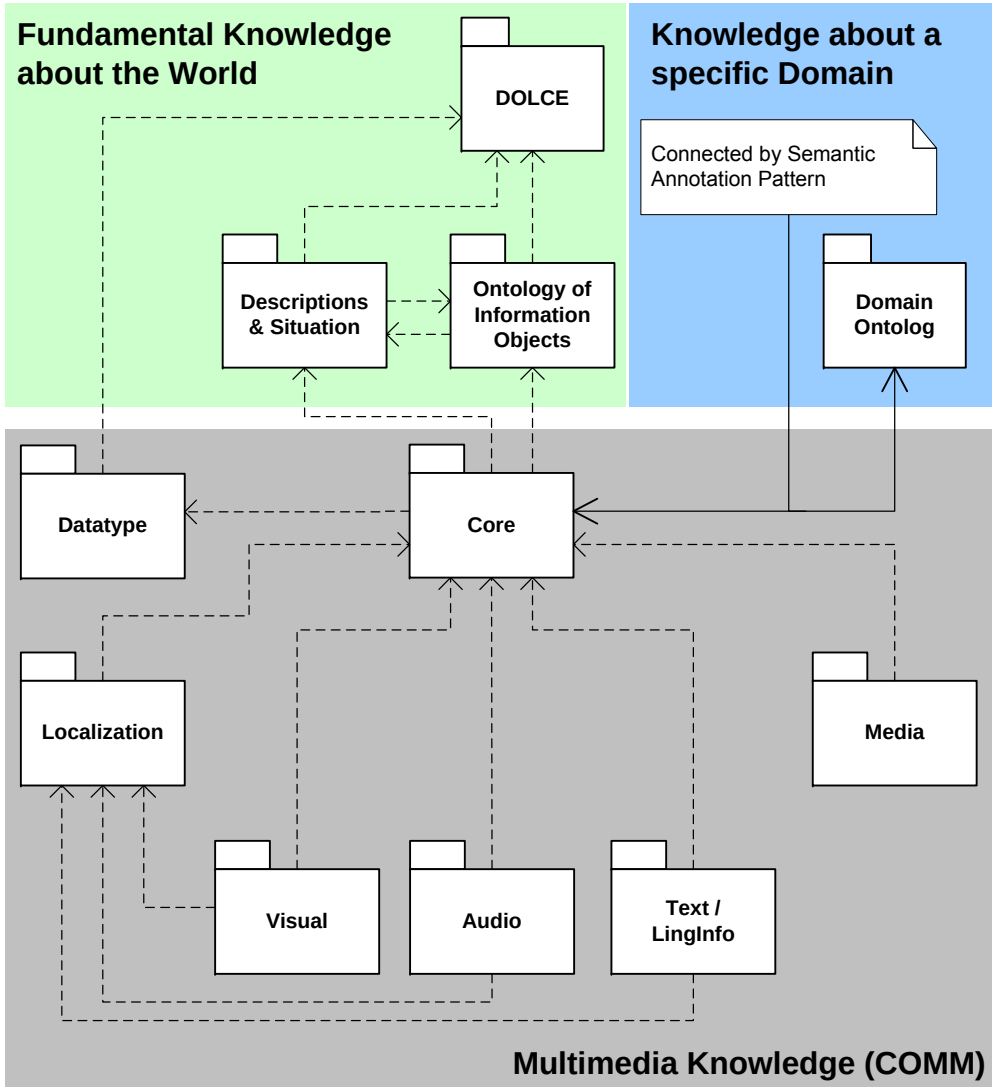
- Constraints prohibit arbitrary placement of MPEG-7 concepts into DOLCE
  - ***Similar concepts will be placed on similar locations of the taxonomy***
  - ***Things that are different, have to be separated***
    (e.g. data and the perceivable content that is carried)

- **Extensibility** due to underlying general taxonomy of DOLCE
  - ⇨ ***Possibility to describe multimedia domain at an arbitrary level of detail***
    (e.g. segments have pixels as atomic parts)

Rigorous application of the D&S and OIO patterns allows
description of digital data in different contexts
(e.g. data acting as input or output for an algorithm)

Multimedia ontology consists of

- **Core module** that contains the design patterns
- Modules that specialize the core module for **different media types**
- Modules that contain **media independent MPEG-7 description tools** such as media information or creation & production
- **Data type module** that formalizes MPEG-7 data types e.g. matrices, vectors, unsigned-int-5, float-vector, probability-vector, …



Reusability

- Clear separation between domain specific and multimedia related knowledge

MPEG-7-Compliance

- Design patterns enable the representation of description tools

Extensibility

- Design patterns are media independent → possibility to include
  - *further media types*
  - *arbitrary descriptors*
- Extensions of multimedia ontology will not affect legacy annotations due to DOLCE+D&S+OIO

Modularity

- Modular architecture allows customization

High degree of axiomatization

- Design patterns come with generic axiomatization that will be refined in derived ontology modules

Linkage with domain ontologies allows **meaningful semantic annotation of multimedia content**

- Semantic part can be entirely replaced with a domain ontology
- **Clear separation between domain ontologies and multimedia core ontology** through semantic annotation pattern

Easier queries

- Annotation pattern guarantees **equal representation of all annotations**
- Complex data type pattern guarantees **uniform access to nested data**
  ⇨ *No complex XML-structures to parse*
- Multimedia ontology only uses **restricted inventory of DOLCE predicates**

Higher interoperability through machine accessible semantics and underlying DOLCE axiomatization

COMM and COMM-API → All description tools that are present in the COMM can be used in Java applications

Currently supported (MPEG-7) description tools:

- All visual low level descriptors (MPEG-7 part 3)
- All media information descriptors (MPEG-7 part 5, clause 8)
- Decomposition tools for
  - Images (StillRegions, SpatialDecomposition, …)
  - Videos (VideoSegments, TemporalDecomposition, …)
  - Text (ASCIITextSegments, ASCIIDecomposition, …)
- Semantic Annotation

COMM is online

http://multimedia.semanticweb.org/COMM/

COMM sources (OWL and Java code) moved to KU SVN repository

Current coexistence of three multimedia ontologies is not problematic as the design patterns can be used to represent

- Visual descriptors and relationship annotations of **VDO 2.0**
- Decomposition tools and segment hierarchy of J**RS-CWI** ontology

Using the design patterns will allow to

- **Add urgently needed descriptors now**
- **Add arbitrary descriptors of MPEG-7 part 3, 4 or 5 in the future**
- **Add new descriptors**
  (e.g. adopt the LingInfo ontology)

[Bailer et al., 2005] Bailer, W., Schallauer, P., and Neuschmied, H. (2005). MPEG-7 Detailed Audiovisual Profile - Description of the Profile. Joanneum Research.

[Gangemi et al., 2005] Gangemi, A., Borgo, S., Catenacci, C., and Lehmann, J. (2005). Task taxonomies for knowledge content. Technical report, Metokis IST-2002-2.3.1.7.

[Hunter, 2003a] Hunter, J. (2003a). Adding multimedia to the semantic web - building and applying an MPEG-7 ontology. Technical report, University of Queensland.

[Oberle, 2005] Oberle, D. (2005). Semantic Management of Middleware. Springer Verlag.

[Oberle et al., 2006] Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S., Baumann, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Porzel, R., Zorn, H.-P., Micelli, V., Schmidt, C., Weiten, M., Burkhardt, F., and Zhou, J. (2006). Dolce ergo sumo: On foundational and domain models in swinto (smart web integrated ontology). Technical report, Institute AIFB, University of Karlsruhe.

[Franz et al., 2007] T. Franz, R. Arndt, S. Staab. The X-Cosim Integration Framework for a seamless semantic desktop. In: K-CAP 2007 – Proceedings of the Fourth International ACM Conference on Knowledge Capture. 28-31 October 2007 Whistler, BC.

[Arndt et al., 2007] R. Arndt, R. Troncy, S. Staab, L. Hardman: *Adding Formal Semantics to MPEG-7: Designing a Well-Founded Multimedia Ontology for the Web*. TechReport, Department of Computer Science, Univ. Koblenz-Landau, nr. 4/2007.

Thank you

Acknowledgements to Richard Arndt @ ISWeb for majority of slides

Appendix: API for COMM

How can I use the COMM-API to export annotations to an RDF-Store?

- COMM-API throws exceptions if invalid annotations are exported
- COMM-API currently lacks a source code documentation
  → Future work

How can I retrieve existing multimedia annotations from an RDF store?

- COMM-API allows reconstruction of Java objects of the MPEG-7 class-interface from RDF triples if “entry-points” of annotation graphs are known
- Sophisticated filtering, e.g. “Give me all Videos that contain a segment showing George W. Bush”, need to be formulated as SPARQL queries by the application programmer
  - SPARQL queries need to return “entry-points” of the wanted annotations
  - Needs deep insight into the data model of the COMM

Annotations are composed of segment annotations, as in MPEG-7

Each annotation contains a “Root-Segment” that represents the whole multimedia content (c.f. MPEG-7)

One or more decomposition(s) can be attached to every segment (including the Root-Segment)

- Decompositions contain segments that are inside the decomposed segment (spatially, temporally, …)

A segment is valid, if it is annotated by

- A media-profile which specifies all required information about its physical support, containing
  - At least one media-instance descriptor which specifies a unique locator (e.g. an URL) of the multimedia content (segment)
- A mask (if it is not a Root-Segment) which specifies the boundaries of the segment within the decomposed parent segment, containing
  - At least one localization-descriptor

Additional segment annotations that are currently supported by the COMM and its API

- Semantic annotation
  - Arbitrarily many can be attached to one segment
  - Contains one or more labels (a label is an URI of an instance of a domain ontology concept)
- Low level descriptor annotation

```
MediaProfile mp = new MediaProfile();

MediaInstanceDescriptor mi = new MediaInstanceDescriptor();

UniqueIDDescriptor uid = new UniqueIDDescriptor();
uid.setUniqueID("unique-ID-of-Video");
mi.setInstanceIdentifier(uid);

MediaLocatorDescriptor mld = new MediaLocatorDescriptor();
mld.setMediaURI(“http://www.example.org/image1.jpg");
mi.setMediaLocator(mld);

mp.addMediaInstance(mi);

Image img0 = new Image();
StillRegion id0 = new StillRegion();
img0.setImage(id0);
id0.addMediaProfile(mp);
```

```
StillRegionSpatialDecomposition srsd = new StillRegionSpatialDecomposition();
img0.getImage().addSpatialDecomposition(srsd);

StillRegion id1 = new StillRegion();
id1.addMediaProfile(img0.getImage().getMediaProfile(0));
SpatialMask smd1 = new SpatialMask();
id1.setMask(smd1);

RegionLocatorDescriptor dd1 = new RegionLocatorDescriptor();
smd1.addSubRegion(dd1);
Polygon p = new Polygon();
p.addPoint(200, 300);
p.addPoint(245, 280);
p.addPoint(290, 250);
dd1.setPolygon(p);

srsd.addStillRegion(id1);
```

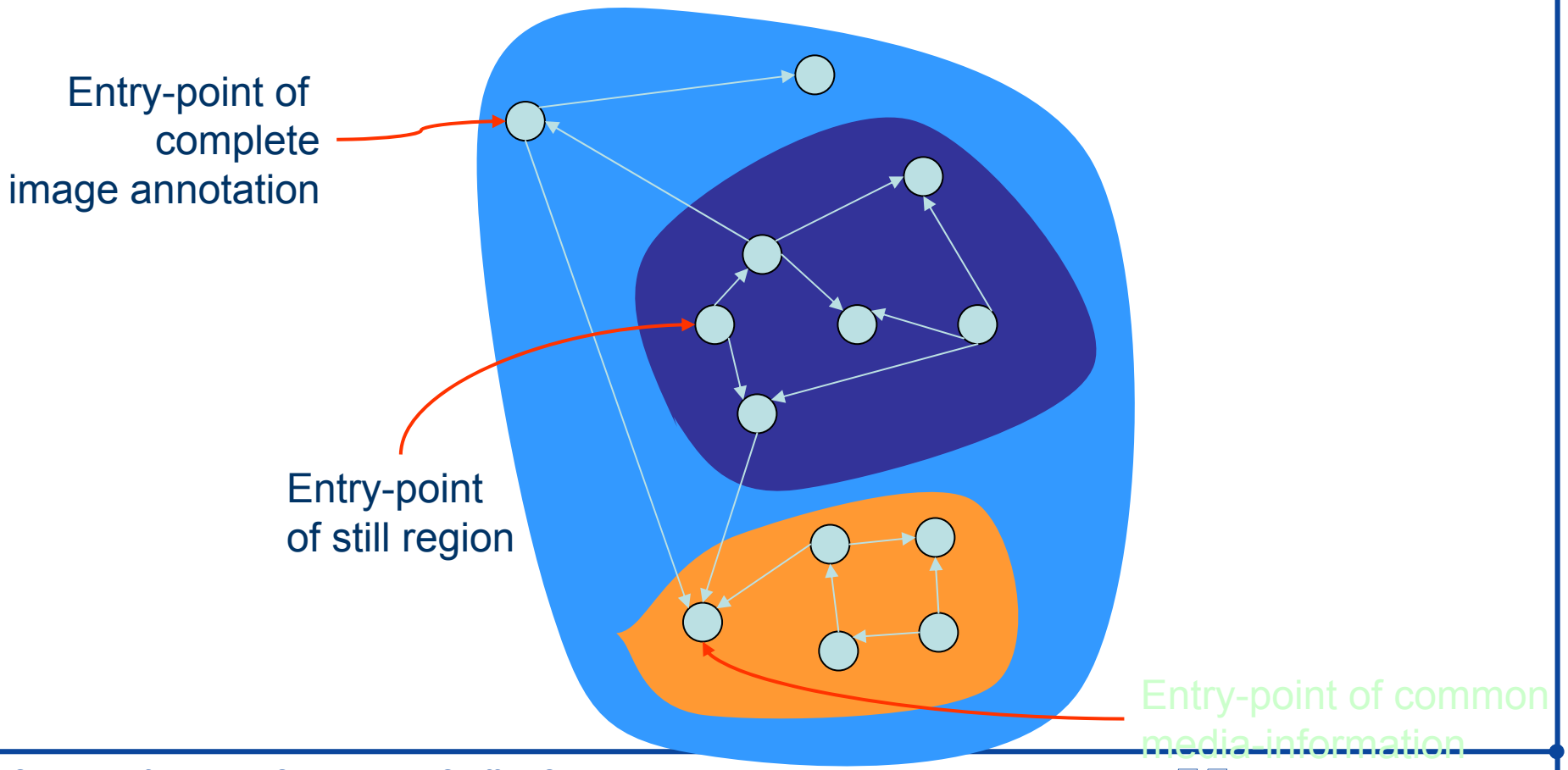
```
Semantic s1 = new Semantic();
s1.addLabel("http://www.ontologies.com/things.owl#georgeWBush");
s1.addLabel("http://www.ontologies.com/otherThings.owl#george_w_bush");
id1.addSemantic(s1);

ScalableColorDescriptor scd = new ScalableColorDescriptor();
Vector<Integer> vec = new Vector<Integer>();
vec.add(0); … vec.add(5645);
scd.setCoeff(vec);
scd.setNumOfCoeff(
  NumberOfCoefficentsEnumerationType.NUMBEROFCOEFFICIENTS_32);
scd.setNumOfBitplanesDiscarded(
  NumberOfBitplanesDiscardedEnumerationType.NUMBEROFBITPLANESDISCARDED_3);

id1.addVisualDescriptor(scd);
```

Using the Serializer-Interface of the COMM-API, the export of the example annotation is simple

```
RDFSerializer serializer = new RDFSerializer();
// Configure serializer according to RDF store e.g. Sesame
img0.serialize("http://www.example.org", serializer);
```

Exceptions will be thrown, if annotation, i.e. the Java object img0 is invalid

COMM-API reconstructs a Java object of the MPEG-7 class interface, if the correct entry-point of the corresponding RDF graph is provided



Entry-Points:
Instances of COMM-concepts which correspond to objects of the Java classes of the COMM-API

Java COMM-API class	COMM concept
Image, Video, Text, …	image-data, video-data, text-data, …
StillRegion, VideoSegment,	still-region-role, video-segment-role,
MediaProfile	media-profile

1-to-1 correspondences between

- Low-Level-Descriptors, e.g. the DominantColorDescriptor class of the COMM-API corresponds to the dominant-color-descriptor concept of the COMM
- Decomposition objects, e.g. the StillRegionSpatialDecomposition class of the COMM-API corresponds to the still-region-spatial-decomposition concept of the COMM

Given an entry-point-URI of an annotation graph, the corresponding Java object of the COMM-API can be reconstructed by calling

```
RDFDeserializer deserializer = new RDFDeserializer();
// Configure deserializer according to RDF store e.g. Sesame
String uriOfEntryPointOfAnImage;
// Has been determined by executing a SPARQL query before
Image img = (Image) COMMObject.constructFromURI(uriOfEntryPointOfAnImage, deserializer);
```

Exceptions will be thrown if entry-point-URI corresponds not to the requested COMM-API class (Image)

COMM-API provides some convenience methods for retrieving annotations from an RDF store

- Retrieving all objects of a COMM-API class (e.g. Image) by calling

```
RDFDeserializer deserializer = new RDFDeserializer();
// Configure deserializer according to RDF store e.g. Sesame
Vector<COMMObject> objs =
  COMMObject.constructAllFromClass(Image.getClass(),
  deserializer);
```

- Get the SPARQL query for retrieving all entry-point-URIs of a COMM-API class (e.g. Image)

```
String sparqlQuery =
  COMMObject.getRetrievalQuery(Image.getClass);
```

Importing the multimedia annotations which satisfy the query

"Give me all image segments that show George W. Bush"

into a Java-application can be solved in 3 steps using the COMM-API

1. Construct a SPARQL-query which selects the entry-points of the wanted segments
   → Has to be done by the application programmer
2. Execute the query
   → Only dependent on the API of the chosen RDF-Store
3. Use the COMM-API to reconstruct the objects of the Segment-class of the MPEG-7 like class interface

```
String query =
“SELECT ?SRR WHERE {
   ?SRR rdf:type comm:still-region-role .
   ?ID edns:plays ?SRR .
   ?ID rdf:type comm:image-data .
   ?SA edns:settingFor ?ID .
   ?SA rdf:type comm:semantic-annotation .
   ?SA edns:settingFor
   http://www.ontologies.com/things.owl#georgeWBush .
}”;
```

Query-string “query” can be evaluated using a RDF-Store API, e.g. Sesame

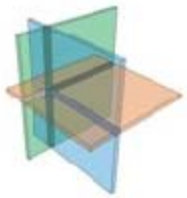
```
// Prepare Sesame for executing the query-string query
TupleQueryResult sesameResult =
    connection.prepareTupleQuery(QueryLanguage.SPARQL,
    query).evaluate();
// Copy URI’s from sesameResult into a Vector<String> vec
```

```
RDFDeserializer deserializer = new RDFDeserializer();
// Configure deserializer according to RDF store e.g. Sesame
Vector<StillRegion> result = new Vector<StillRegion>();
for (int i = 0; i < vec.size(); ++i) {
    String entryPointUri = vec.elementAt(i);
    StillRegion sr = (StillRegion)
        COMMObject.constructFromURI(entryPointUri, deserializer);
    result.add(sr);
}
```

Semantic Web

- Multimedia Annotation –

Steffen Staab
http://isweb.uni-koblenz.de



UNIVERSITÄT KOBLENZ · LANDAU

X-MEDIA

Different levels of annotations

- Metadata
  - Often technical metadata
  - EXIF, Dublin Core, access rights
- Content level
  - Semantic annotations
  - Keywords, domain ontologies, free-text
- Multimedia level
  - low-level annotations
  - Visual descriptors, such as dominant color

Metadata

refers to information about technical details

creation details

- creator, creationDate, …
- Dublin Core

camera details

- settings
- resolution
- format
- EXIF

access rights

- administrated by the OS
- owner, access rights, …

Content Level

Describes what is depicted and directly perceivable by a human

usually provided manually

- keywords/tags
- classification of content

seldom generated automatically

- scene classification
- object detection

different types of annotations

- global vs. local
- different semantic levels

Global vs. Local Annotations

Global annotations most widely used

- flickr: tagging is only global
- organization within categories
- free-text annotations
- provide information about the content as a whole
- no detailed information

Local annotations are rarely supported

- e.g. flickr, PhotoStuff allow to provide annotations of regions
- especially important for semantic image understanding
  - allow to extract relations
  - provide a more complete view of the scene
- provide information about different regions
- and about the depicted relations and arrangements of objects

Semantic Levels

Free-Text annotations cover large aspects, but less appropriate for sharing, organization and retrieval

- Free-Text Annotations probably most natural for the human, but provide least formal semantics

Tagging provides light-weight semantics

- More useful if a fixed vocabulary is used
- Allows some simple inference of related concepts by tag analysis (clustering)
- No formal semantics, but provides benefits due to (fixed) vocabulary
- Requires more effort from the user

Ontologies

- Provide syntax and semantic to define complex domain vocabularies
- Allow for the inference of additional knowledge
- Leverage interoperability
- Powerful way of semantic annotation, but hardly comprehensible by “normal users”

Web-based Tools

- flickr
- Riya

Stand-Alone Tools

- PhotoStuff
- AktiveMedia

Annotation for Feature Extraction

- KAT

flickr

<is web>

Web2.0 application

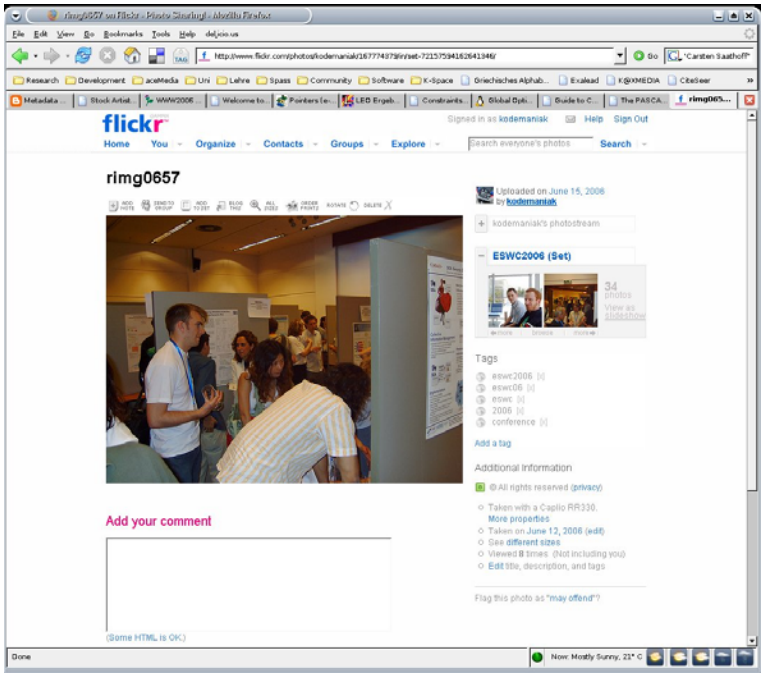
tagging photos globally

add comments to image regions
marked by bounding box

large user community and tagging
allows for easy sharing of images

partly fixed vocabularies evolved

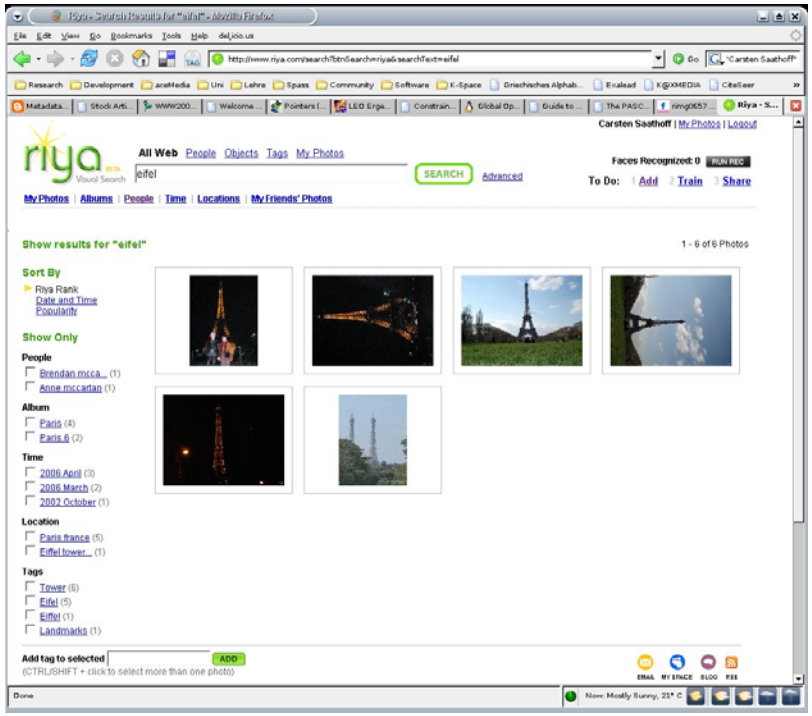
- e.g. Geo-Tagging



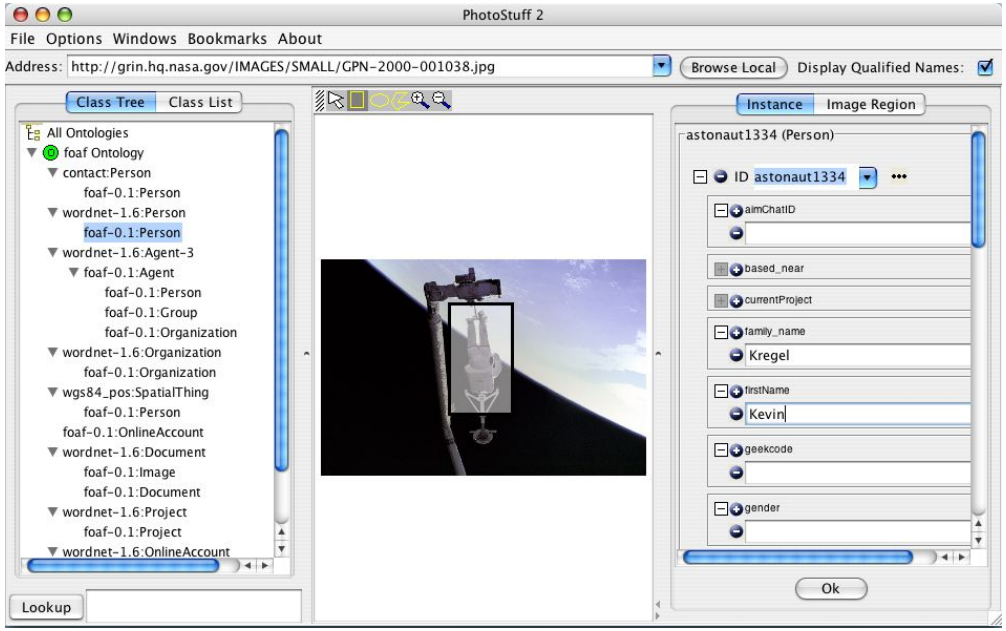
Similar to flickr in functionality

Adds automatic annotation features

- Face Recognition
  - Mark faces in photos
  - associate name
  - train system
  - automatic recognition of the person in the future



PhotoStuff



- Java application for the annotation of images and image regions with domain ontologies
- Used during ESWC2006 for annotating images and sharing metadata
- Developed within Mindswap

AktiveMedia

Text and image annotation tool

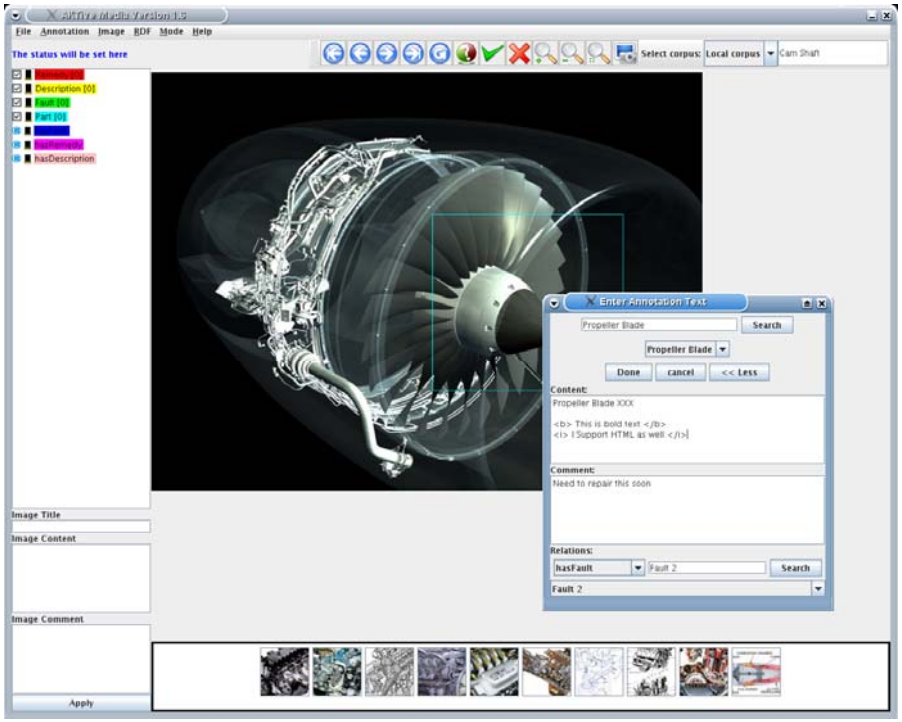
Region-based annotation

Uses ontologies

- suggests concepts during annotation
- providing a simpler interface for the user

Provides semi-automatic annotation of content, using

- Context
- Simple image understanding techniques
- flickr tagging data

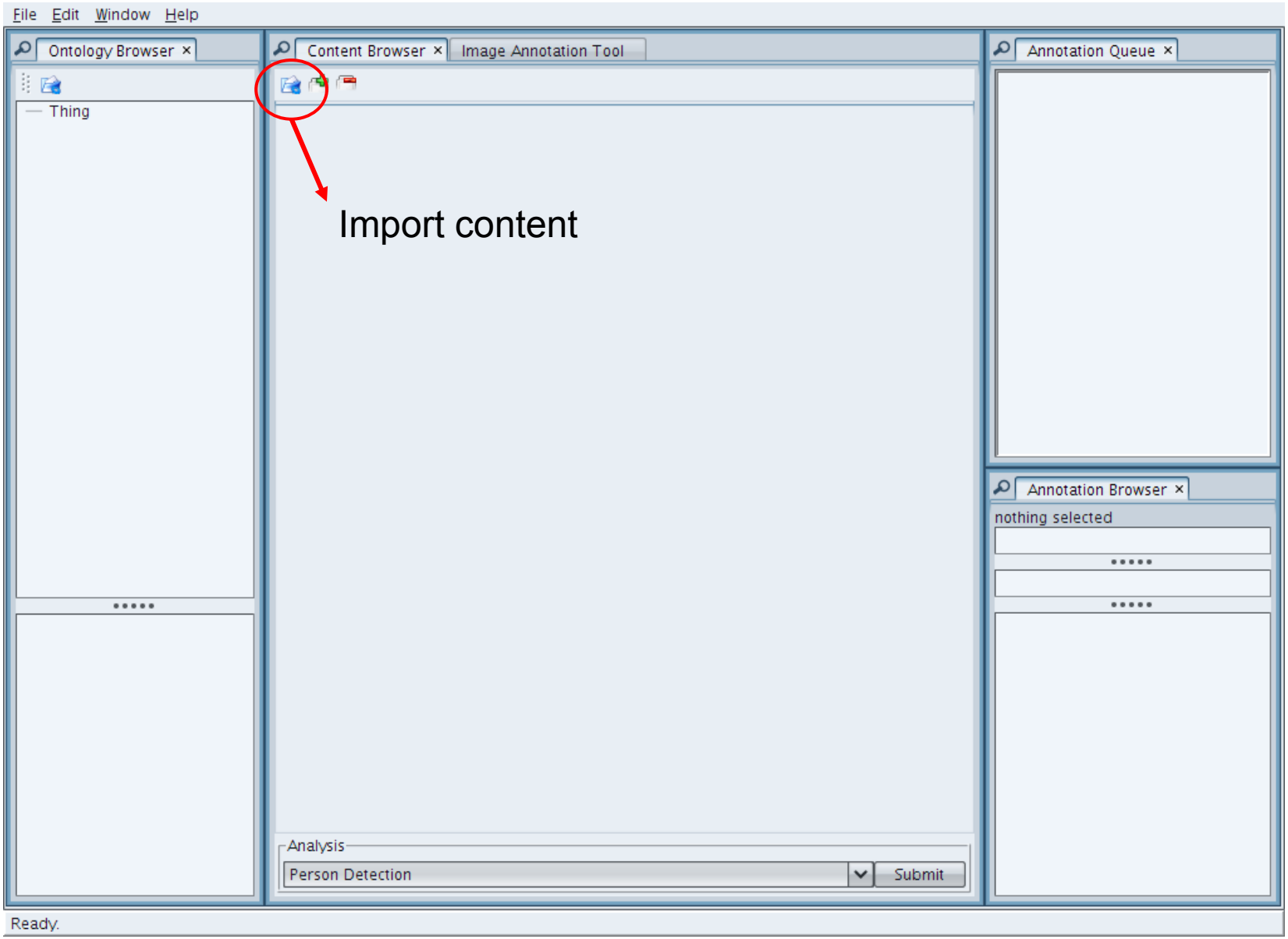


Knowledge Annotation Tool – KAT

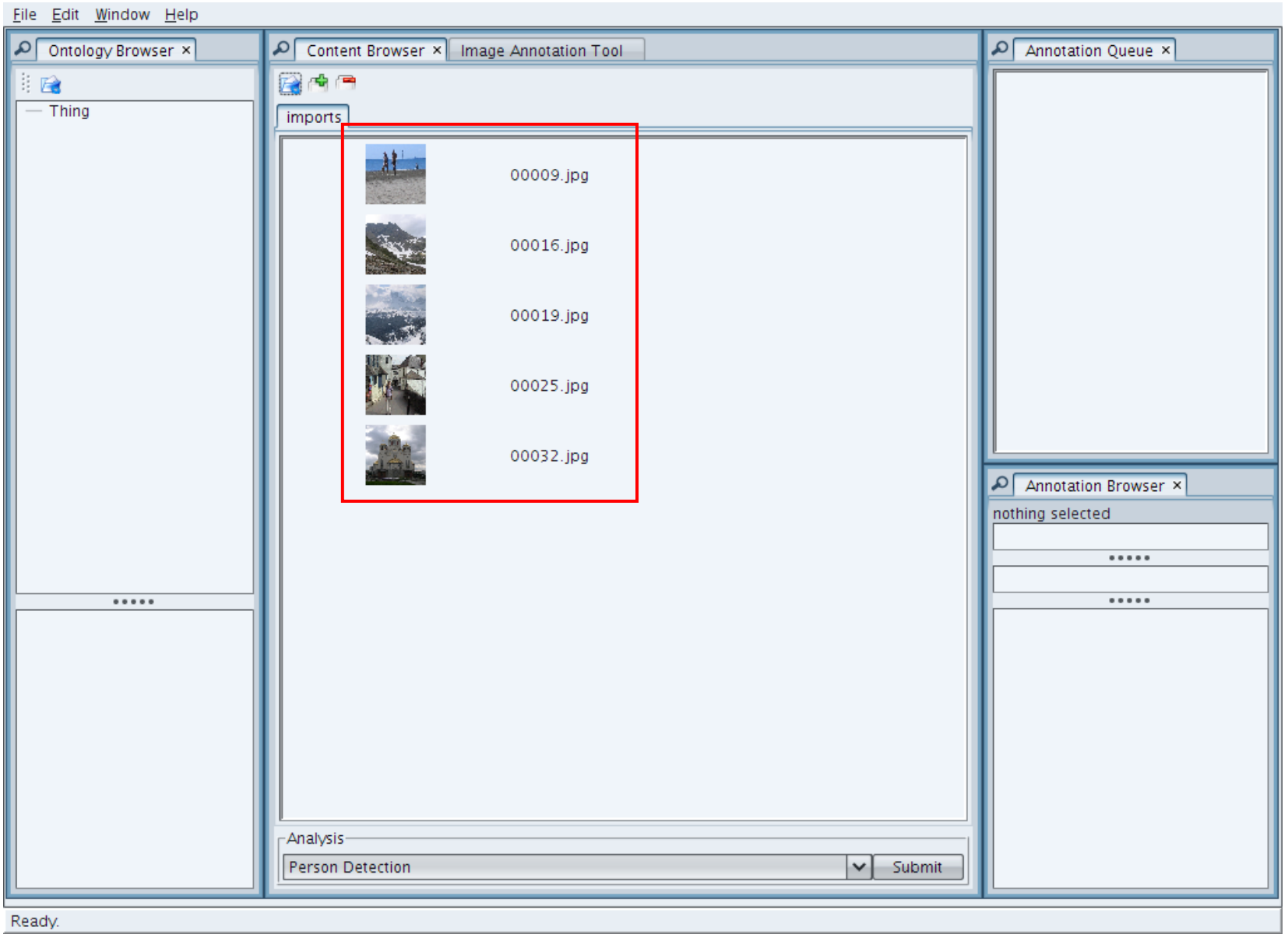
Open source framework for fine-grained annotation

Per default not useful for organizing your private photo show
– but framework allows for new plug-ins – go ahead!

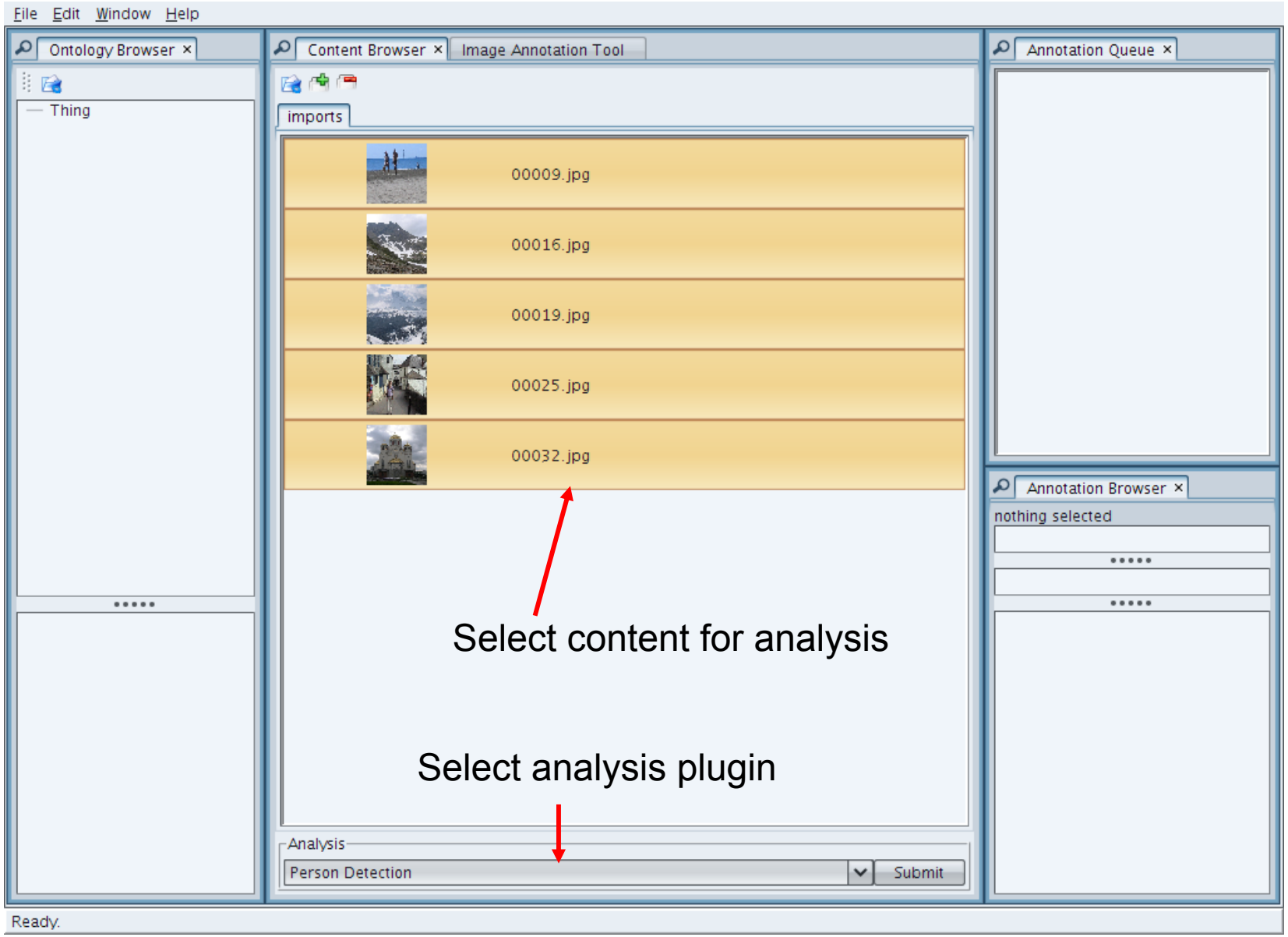
After start-up



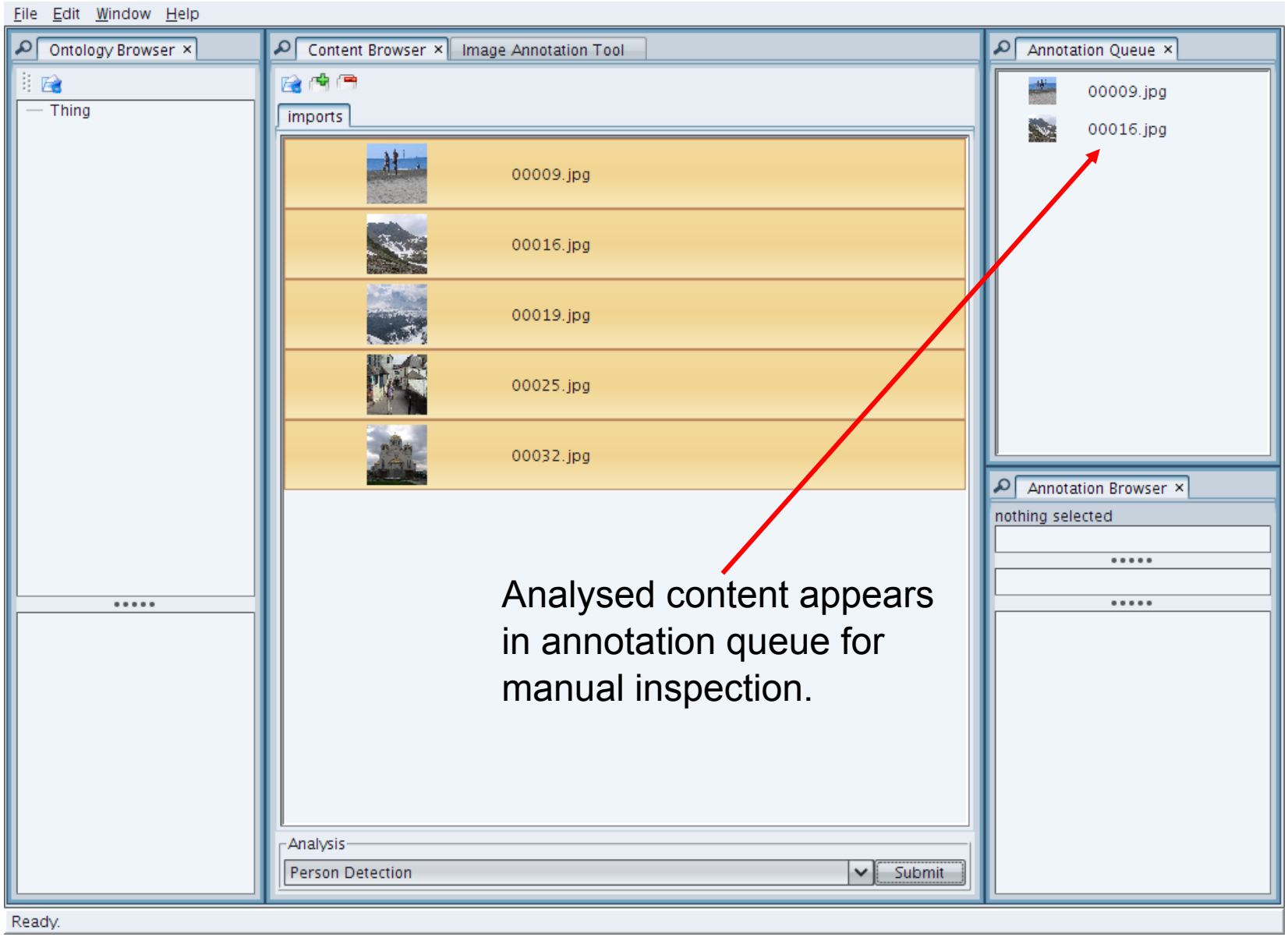
Imported content



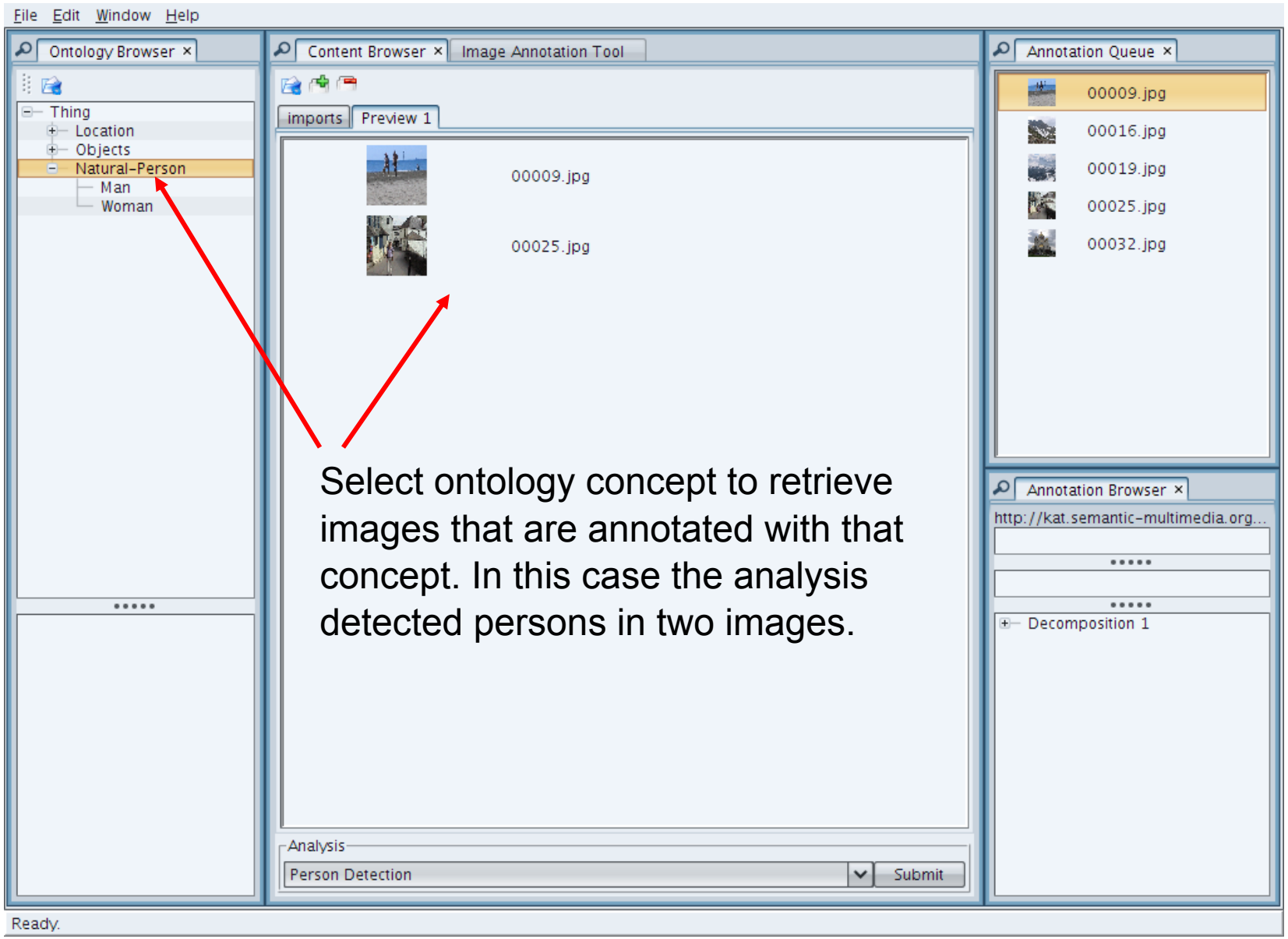
Select for analysis



Annotation queue

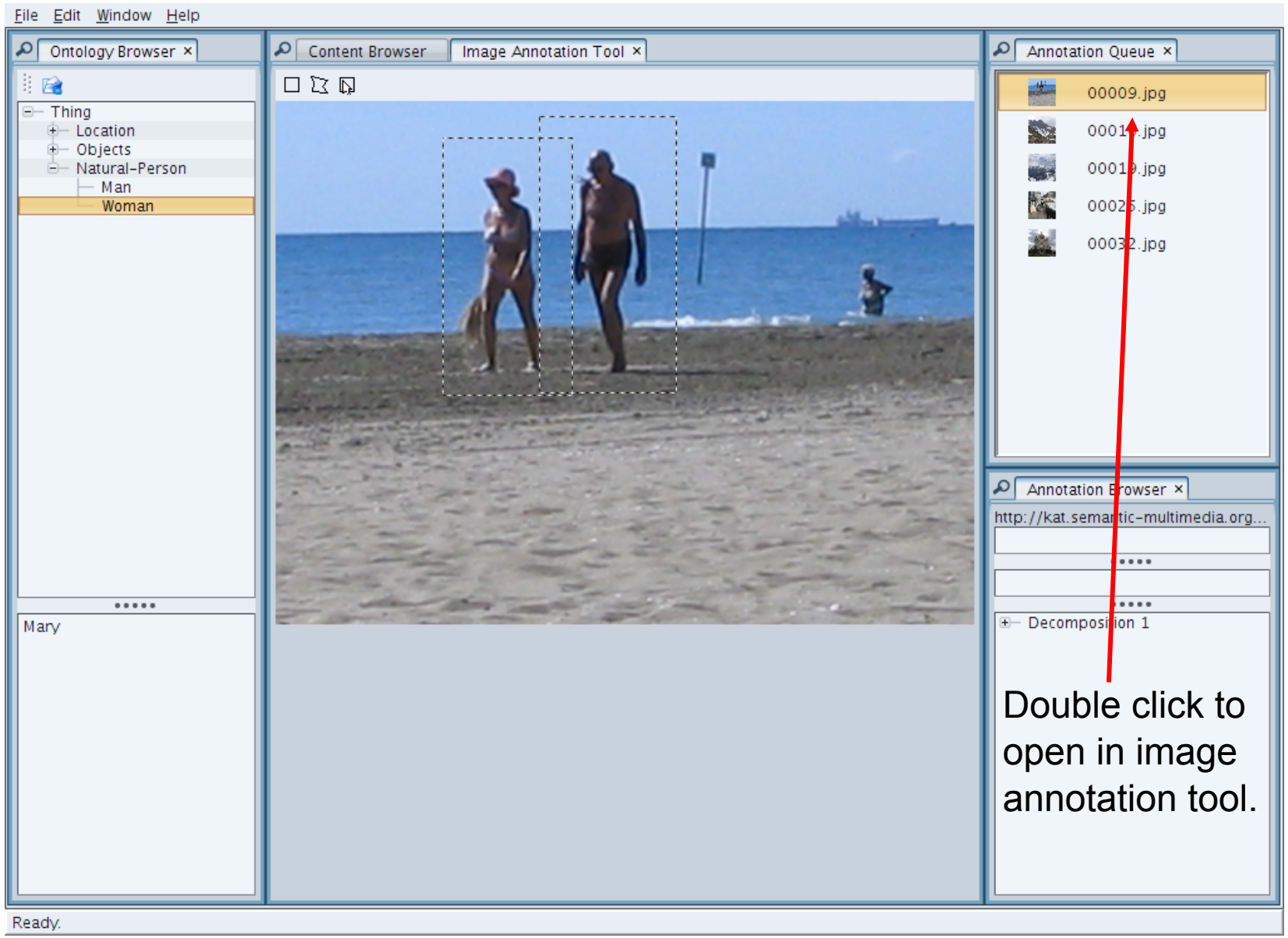


Retrieval

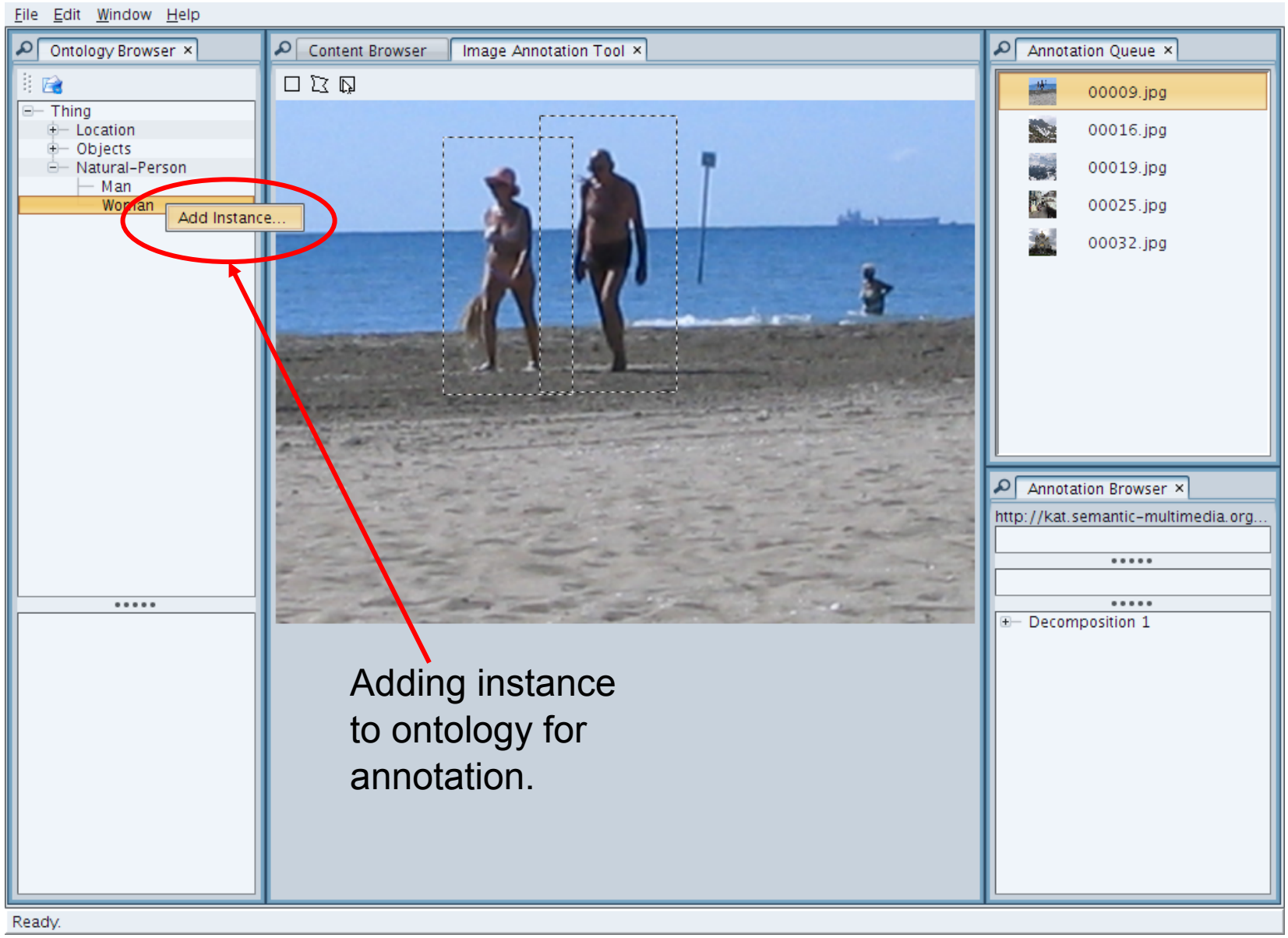


Select ontology concept to retrieve images that are annotated with that concept. In this case the analysis detected persons in two images.

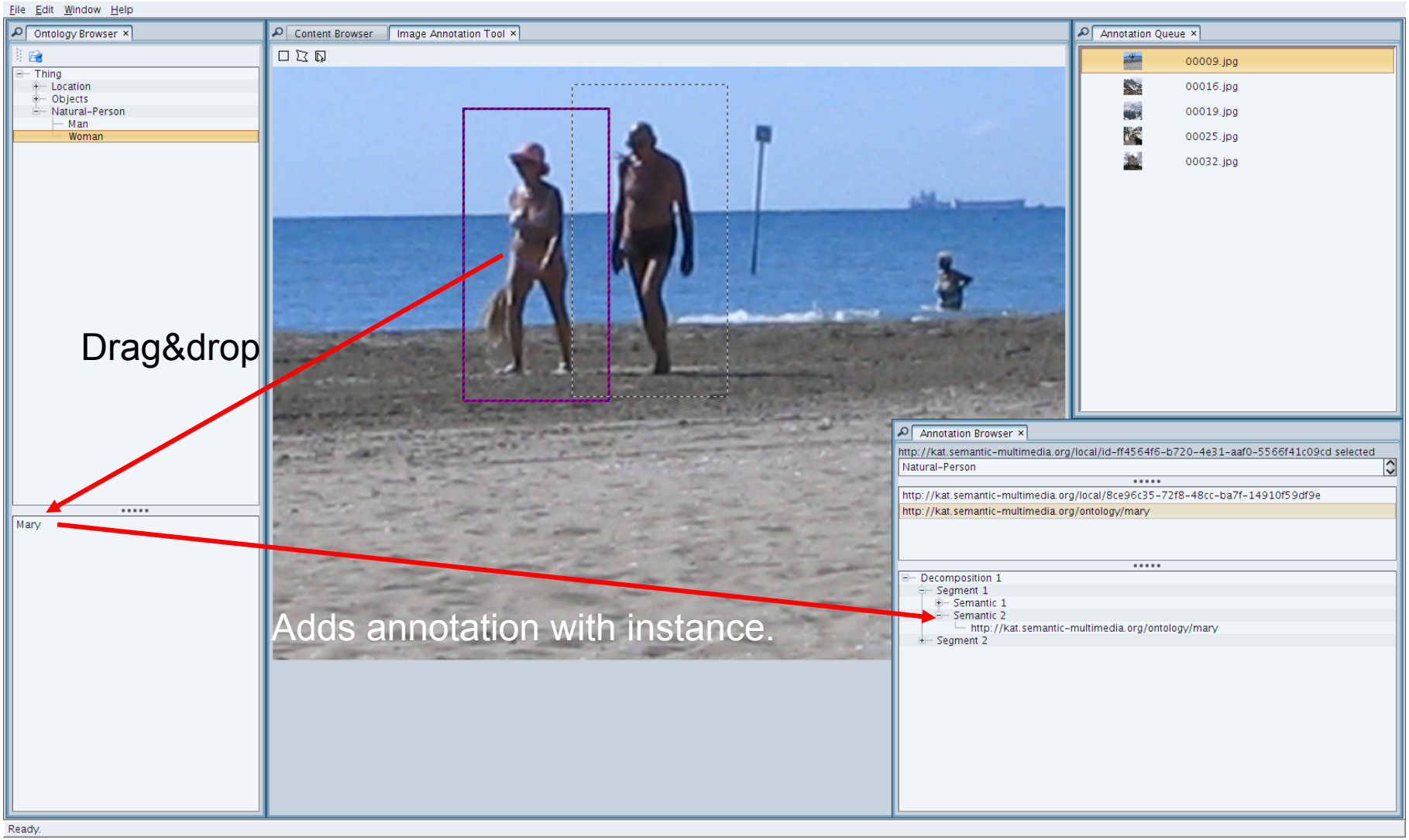
Image Annotation Tool



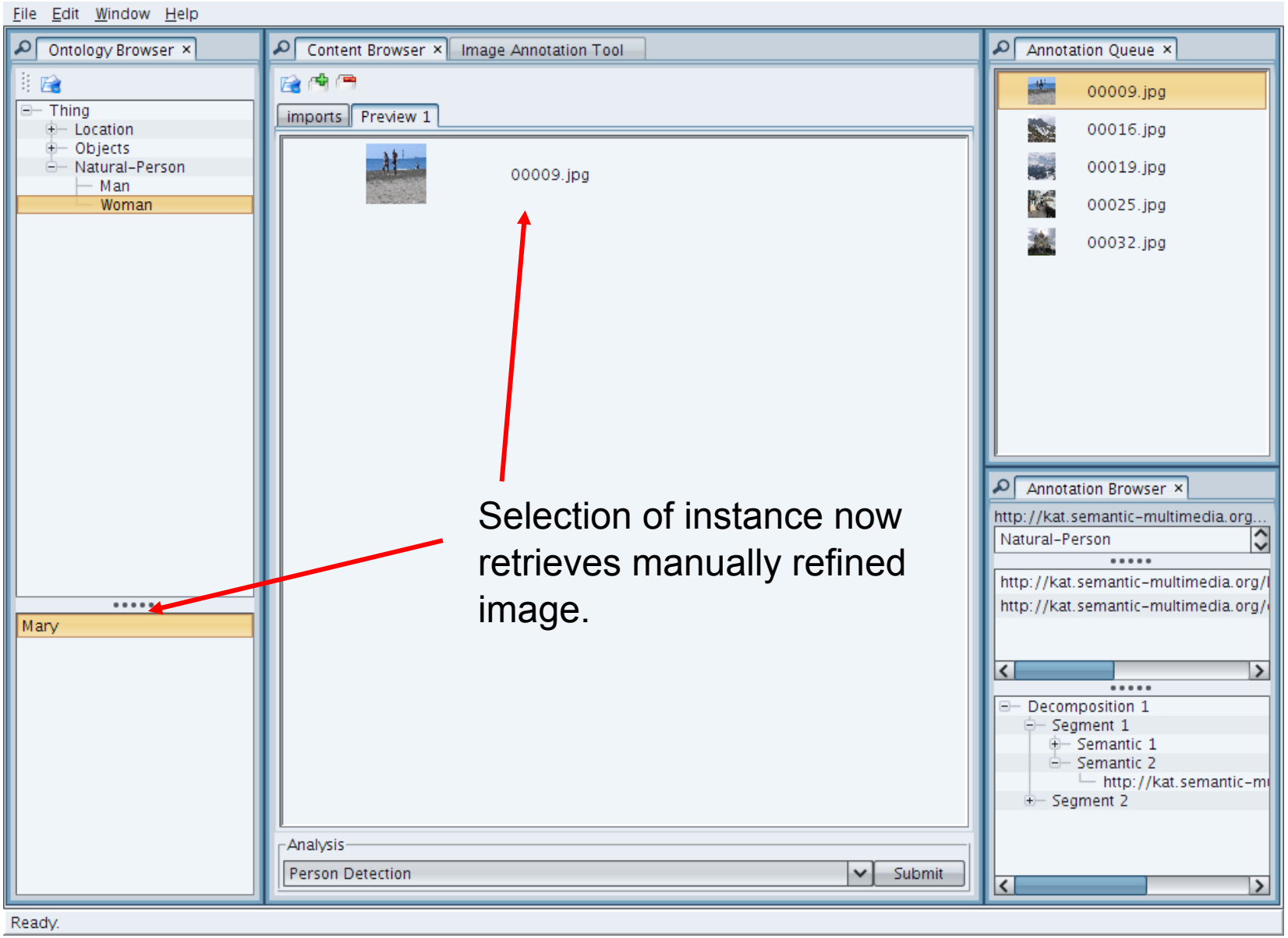
Adding instances



Refine annotation



Retrieval using refined annotation



Next steps

Existing analysis tools (in C) will be provided via Web Service

Provide your own tool!

- In Java
- In C via web service

Thank you

Acknowledgements to Carsten Saathoff @ ISWeb for majority of slides