Semantic Web and Multimedia

- Motivation & Agenda –

Steffen Staab
http://isweb.uni-koblenz.de
Semantic Web for Multimedia

Semantic Aspects

Semantics in Multimedia

Social dynamics – the Web as an IT enabler

Web Aspects

Multimedia on the Semantic Web

Formal semiotics – logics and probabilities as core tools

What is the meaning of some multimedia data?

How are things interwoven, i.e. networked?

Semiotics
Semantics in Multimedia: What for?

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?
Multimedia Semantics

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

Technical perspective

Multimedia content semantics
Multimedia situational semantics
Multimedia usage semantics
Multimedia social semantics

Human perspective

common vacation

And semiotics will tell you that this list is incomplete…

boy
cute
Turkey
Multimedia Semantics

- Multimedia social semantics
  - common vacation
- Multimedia retrieval semantics
  - cute
- Multimedia situational semantics
  - Turkey
- Multimedia content semantics
  - boy

- Manual Class. & Mgmt
- Feedback & AutoClassify
- Capture (during creation)
- Extract from (image) data
Find me a cute picture of Nico digging during our common vacation.
"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?*
Multimedia on the Semantic Web

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
Some are missing..

Students you may find by Wiki + social search:
Some are missing

New members I may find in my mail box:

M. Sc. Renata Dividino

Education
2004 - 2007
2007 - today

Personal Data

Privacy protected
Some are missing…

New members you may find on LinkedIn:

Maciej Janik
PhD candidate at University of Georgia
Athens, Georgia Area

Contact Directly
Get introduced through a connection

Public profile powered by: LinkedIn
The Semantic Web: Tying it together

- ISWeb
- SCIENTIFIC PHD
- RenataDividinoCV
- Maciej Janik - Link
- Friedrich Pfitzmann

Mail
File
Social Site
Web Page
Demo Parallax

http://www.vimeo.com/1513562?pg=embed&sec=1513562
Core Questions for Multimedia in Semantic Web

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?
Agenda Items

Ontology Representation Languages

Multimedia Ontologies

Ontology-based

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

Not considered in this tutorial!
Semantic Web and Multimedia
- Ontologies & Their Languages –

Steffen Staab
http://isweb.uni-koblenz.de
Ontologies
Ontology – in computer science

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

Gruber 93:

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

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

Domain and application-specific ontologies:
- Dublin Core, http://dublincore.org/

Semantic Desktop Ontologies
- X-COSIM Ontology, http://isweb.uni-koblenz.de/Research/X-cosim

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
Ontologies and Their Relatives (cont’d)

Front-End

- Thesauri
- Topic Maps

- Information Retrieval
- Navigation
- Sharing of Knowledge

- Query Expansion

Back-End

- Ontologies
- Semantic Networks

- Extended ER-Models
- Queries

- Mediation
- EAI
- Consistency Checking
- Reasoning

- Predicate Logic
<table>
<thead>
<tr>
<th><strong>Ontology Trade-off</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very formal (e.g. DOLCE)</strong></td>
<td><strong>Informal (e.g. Gene Ontology)</strong></td>
</tr>
<tr>
<td>A lot of reasoning power</td>
<td>Little to no reasoning possible</td>
</tr>
<tr>
<td>Expensive to build</td>
<td>Comparatively inexpensive (total costs for Gene Ontology are not low!)</td>
</tr>
<tr>
<td>Misunderstandings can be corrected by expert developers (costs may be incurred)</td>
<td>Misunderstandings due to ambiguity are hard to correct (very high costs may be incurred!)</td>
</tr>
<tr>
<td>Ontology MUST be (at least partially) hidden from its users</td>
<td>Ontology may appeal to intuition of user</td>
</tr>
</tbody>
</table>
RDF
RDF Data Model

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
A simple Example

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

http://www.w3.org/Home/Lassila s:Creator Ora Lassila
Another Example

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

http://www.w3.org/Home/Lassila

s:Creator

Person://fi/654645635

Name
Ora Lassila

Email
lassila@w3.org
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

 RDF

 bagid1
 /courses/6.001
 students

 Rdf:Bag

 /Students/Amy

 /Students/Tim

 /Students/John

 /Students/Mary

 /Students/Sue

 RDF

 RDF
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](http://www.mit.edu/~lieber)
- S:WalksOn
- AtlanticOcean
Datamodel does not enforce particular syntax
Specification suggests many different syntaxes based on XML
General form:

```xml
<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>
```
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>
```
Typing Information

<s:Homepage rdf:about="http://www.w3.org/Home/Lassila"
    s:Creator="Ora Lassila"/>

<s:Title>Ora's Home Page</s:Title>

<s:createdWith>
    <s:HTMLEditor rdf:about="http://www.w3c.org/amaya"/>
</s:createdWith>

</s:Homepage>

Property

http://www.w3.org/Home/Lassila rdf:type s:Homepage

s:Creator

Ora Lassila

s:createdWith

http://www.w3.org/Home/Lassila

rdf:type

HTMLEditor

http://www.w3c.org/amaya
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

rdfs:Class

xyz:MotorVehicle

xyz:Truck

xyz:PassengerVehicle

xyz:Van

xyz:MiniVan

xyz:myvan

s = rdf:type

t = rdfs:subClassOf
<rdfs:subclassOf about="Xyz:Minivan">
  <rdfs:subclassOf about="xyz:Van"/>
</rdfs:subclassOf>

<rdfs:subclassOf about="myvan">
  <rdf:type about="xyz:MiniVan"/>
</rdfs:subclassOf>

**Predicate Logic Consequences:**

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

```xml
<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 $\subseteq$ OWL DL $\subseteq$ OWL2
OWL Lite $\subseteq$ OWL DL $\subseteq$ OWL Full

RDFS is fragment of OWL Full
OWL DL is decidable
OWL DL = SHOIN(D) (description logics)
W3C-Document contains many more details that we cannot talk about here
OWL Overview

OWL – Syntax and semantics

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

Knowledge Base

Tbox (schema)

\[ \text{Man} \equiv \text{Human} \cap \text{Male} \]

\[ \text{Happy-Father} \equiv \text{Man} \cap \exists \text{has-child.Female} \cap \ldots \]

Abox (data)

\[ \text{Happy-Father(John)} \]

\[ \text{has-child(John, Mary)} \]

Inference System

Interface

- 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 $\cap$, $\cup$, $\neg$ (intersection, union, negation)
- Quantors define how roles are to be interpreted:
  
  Man $\cap \exists$hasChild.Female $\cap \exists$hasChild.Male
  $\cap \forall$hasChild.(Rich $\sqcup$ Happy)
Further DL concepts and role constructors

Number restrictions (cardinality constraints) for roles:
\[ \geq 3 \text{ hasChild}, \quad \leq 1 \text{ hasMother} \]

Qualified number restrictions:
\[ \geq 2 \text{ hasChild.Female}, \quad \leq 1 \text{ hasParent.Male} \]

Nominals (definition by extension):
\{Italy, France, Spain\}

Concrete domains (datatypes): \( \text{hasAge.(\geq 21)} \)

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

Transitive roles: \( \text{hasAncestor}^{\ast} \) (descendant)

Role composition: \( \text{hasParent}.\text{hasBrother} \) (uncle)
DL Knowledge Bases consist of two parts (in general):

- **TBox**: Axioms, describing the structure of a modelled domain (conceptual schema):
  - HappyFather $\equiv$ Man $\sqcap$ $\exists$hasChild.Female $\sqcap$ …
  - Elephant $\sqsubseteq$ Animal $\sqcap$ Large $\sqcap$ 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.
Content

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

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL Syntax</th>
<th>Example</th>
<th>FOL Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>intersectionOf</td>
<td>$C_1 \sqcap \ldots \sqcap C_n$</td>
<td>Human \sqcap Male</td>
<td>$C_1(x) \land \ldots \land C_n(x)$</td>
</tr>
<tr>
<td>unionOf</td>
<td>$C_1 \sqcup \ldots \sqcup C_n$</td>
<td>Doctor \sqcap Lawyer</td>
<td>$C_1(x) \lor \ldots \lor C_n(x)$</td>
</tr>
<tr>
<td>complementOf</td>
<td>$\neg C$</td>
<td>\neg Male</td>
<td>$\neg C(x)$</td>
</tr>
<tr>
<td>oneOf</td>
<td>${x_1} \sqcup \ldots \sqcup {x_n}$</td>
<td>{john} \sqcup {mary}</td>
<td>$x = x_1 \lor \ldots \lor x = x_n$</td>
</tr>
<tr>
<td>allValuesFrom</td>
<td>$\forall P.C$</td>
<td>\forall hasChild.Doctor</td>
<td>$\forall y.P(x, y) \rightarrow C(y)$</td>
</tr>
<tr>
<td>someValuesFrom</td>
<td>$\exists P.C$</td>
<td>\exists hasChild.Lawyer</td>
<td>$\exists y.P(x, y) \land C(y)$</td>
</tr>
<tr>
<td>maxCardinality</td>
<td>$\leq n P$</td>
<td>\leq 1 hasChild</td>
<td>$\exists y.P(x, y)$</td>
</tr>
<tr>
<td>minCardinality</td>
<td>$\geq n P$</td>
<td>\geq 2 hasChild</td>
<td>$\exists \geq n y.P(x, y)$</td>
</tr>
</tbody>
</table>

Nesting of expression is allowed at arbitrary depth:

\[
\text{Person} \sqcap \forall \text{hasChild.(Doctor} \sqcap \exists \text{hasChild.Doctor})
\]
## OWL DL as DL: Axioms

<table>
<thead>
<tr>
<th>Axiom</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>subClassOf</td>
<td>$C_1 \sqsubseteq C_2$</td>
<td>Human $\sqsubseteq$ Animal $\cap$ Biped</td>
</tr>
<tr>
<td>equivalentClass</td>
<td>$C_1 \equiv C_2$</td>
<td>Man $\equiv$ Human $\cap$ Male</td>
</tr>
<tr>
<td>disjointWith</td>
<td>$C_1 \sqsubseteq \neg C_2$</td>
<td>Male $\sqsubseteq \neg$ Female</td>
</tr>
<tr>
<td>sameIndividualAs</td>
<td>${x_1} \equiv {x_2}$</td>
<td>${\text{President Bush}} \equiv {\text{G_W Bush}}$</td>
</tr>
<tr>
<td>differentFrom</td>
<td>${x_1} \sqsubseteq \neg {x_2}$</td>
<td>${\text{john}} \sqsubseteq \neg {\text{peter}}$</td>
</tr>
<tr>
<td>subPropertyOf</td>
<td>$P_1 \sqsubseteq P_2$</td>
<td>hasDaughter $\sqsubseteq$ hasChild</td>
</tr>
<tr>
<td>equivalentProperty</td>
<td>$P_1 \equiv P_2$</td>
<td>cost $\equiv$ price</td>
</tr>
<tr>
<td>inverseOf</td>
<td>$P_1 \equiv P_2^-$</td>
<td>hasChild $\equiv$ hasParent$^-$</td>
</tr>
<tr>
<td>transitiveProperty</td>
<td>$P^+ \sqsubseteq P$</td>
<td>ancestor$^+$ $\sqsubseteq$ ancestor</td>
</tr>
<tr>
<td>functionalProperty</td>
<td>$\top \sqsubseteq \leq 1P$</td>
<td>$\top \leq 1\text{hasMother}$</td>
</tr>
<tr>
<td>inverseFunctionalProperty</td>
<td>$\top \sqsubseteq \leq 1P^-$</td>
<td>$\top \leq 1\text{hasSSN}^-$</td>
</tr>
</tbody>
</table>

### General Class Inclusion ($\sqsubseteq$):

$$C \equiv D \iff (C \sqsubseteq D \text{ und } D \sqsubseteq C)$$

### Obvious equivalances with FOL:

$$C \equiv D \iff (\forall x) \ (C(x) \iff D(x))$$

$$C \sqsubseteq D \iff (\forall x) \ (C(x) \rightarrow D(x))$$
Terminological Knowledge (TBox):
Human $\sqsubseteq \exists$hasParent.Human
Orphan $\equiv$ Human $\sqcap \neg \exists$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

<table>
<thead>
<tr>
<th>Expression</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>A</code></td>
<td>Subset of $\Delta^I$</td>
</tr>
<tr>
<td>$\neg C$</td>
<td>$\Delta^I \setminus C^I$</td>
</tr>
<tr>
<td><code>C \land D</code></td>
<td>${x</td>
</tr>
<tr>
<td><code>C \lor D</code></td>
<td>${x</td>
</tr>
<tr>
<td>$\exists R.C$</td>
<td>${x</td>
</tr>
<tr>
<td>$\forall R.C$</td>
<td>${x</td>
</tr>
<tr>
<td>$\geq n R.C$</td>
<td>${x</td>
</tr>
<tr>
<td>$\leq n R.C$</td>
<td>${x</td>
</tr>
<tr>
<td>${i_1, \ldots, i_n}$</td>
<td>${i_1^I, \ldots, i_n^I}$</td>
</tr>
</tbody>
</table>

Role expressions

<table>
<thead>
<tr>
<th>Role</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R</code></td>
<td>Subset of $\Delta \times \Delta$</td>
</tr>
<tr>
<td><code>R^-</code></td>
<td>${(y,x)</td>
</tr>
</tbody>
</table>

Ontology (=Knowledge Base)

Concept Axioms (TBox)

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>C \subseteq D</code></td>
<td>$C^I \subseteq D^I$</td>
</tr>
<tr>
<td><code>C \equiv D</code></td>
<td>$C^I \equiv D^I$</td>
</tr>
</tbody>
</table>

Role Axioms (rarely: RBox)

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R \subseteq S</code></td>
<td>$R^I \subseteq S^I$</td>
</tr>
</tbody>
</table>

Assertional Axioms (ABox)

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>C(a)</code></td>
<td>$a^I \in C^I$</td>
</tr>
<tr>
<td><code>R(a,b)</code></td>
<td>$(a^I,b^I) \in R^I$</td>
</tr>
<tr>
<td><code>a = b</code></td>
<td>$a^I = b^I$</td>
</tr>
<tr>
<td><code>a \neq b</code></td>
<td>$a^I \neq b^I$</td>
</tr>
</tbody>
</table>
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
## Serializations/different Syntaxes

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWL RDF Syntax</td>
<td>W3C recommendation</td>
</tr>
<tr>
<td>OWL Abstract Syntax</td>
<td>W3C recommendation See next section</td>
</tr>
<tr>
<td>OWL XML Syntax</td>
<td>W3C document</td>
</tr>
<tr>
<td>DL Notation</td>
<td>widely used in scientific contexts</td>
</tr>
<tr>
<td>FOL Notation</td>
<td>uncommon</td>
</tr>
</tbody>
</table>
Example: RDF Syntax

\[ \text{Person} \sqcap \forall \text{hasChild}. (\text{Doctor} \sqcup \exists \text{hasChild}. \text{Doctor}) : \]

\[
<\text{owl:Class}>
  <\text{owl:intersectionOf} rdf:parseType="collection">
    <\text{owl:Class} rdf:about="#\text{Person}"/>
    <\text{owl:Restriction}>
      <\text{owl:onProperty} rdf:resource="#\text{hasChild}"/>
      <\text{owl:allValuesFrom}>
        <\text{owl:unionOf} rdf:parseType="collection">
          <\text{owl:Class} rdf:about="#\text{Doctor}"/>
          <\text{owl:Restriction}>
            <\text{owl:onProperty} rdf:resource="#\text{hasChild}"/>
            <\text{owl:someValuesFrom} rdf:resource="#\text{Doctor}"/>
          </\text{owl:Restriction}>
        </\text{owl:unionOf}>
      </\text{owl:allValuesFrom}>
    </\text{owl:Restriction}>
  </\text{owl:intersectionOf}>
</\text{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))
  ...
)
Knowledge modelling: examples

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

\[ \text{bus\_driver} \equiv \text{person} \land \exists \text{drives}.bus \]

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

\[ \text{driver} \equiv \text{person} \land \exists \text{drives}.vehicle \]

Class(a:bus partial a:vehicle)

\[ \text{bus} \sqsubseteq \text{vehicle} \]

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))))

\( \text{driver} \equiv \text{person} \cap \exists \text{drives.vehicle} \)

Class(a:driver partial a:adult)

\( \text{driver} \subseteq \text{adult} \)

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

\( \text{grownup} \equiv \text{adult} \cap \text{person} \)

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).
Knowledge modelling: OWA vs. CWA

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.

<table>
<thead>
<tr>
<th>child(Bill, Bob)</th>
<th>Man(Bob)</th>
<th>( \leq 1 ) child.T(Bill)</th>
</tr>
</thead>
<tbody>
<tr>
<td>? \models \forall \text{child}.\text{Man}(\text{Bill})</td>
<td>\text{Prolog yes}</td>
<td>\text{Yes}</td>
</tr>
<tr>
<td>? \models \forall \text{child}.\text{Man}(\text{Bill})</td>
<td>\text{DL answers don't know}</td>
<td>\text{No idea, since we do not know all children of Bill.}</td>
</tr>
<tr>
<td>? \models \forall \text{child}.\text{Man}(\text{Bill})</td>
<td>\text{If we assume that we know everything about Bill, then all of his children are male.}</td>
<td></td>
</tr>
</tbody>
</table>
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
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

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

Query:
SELECT ?title
WHERE

Query Result:

<table>
<thead>
<tr>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;SPARQL Tutorial&quot;</td>
</tr>
</tbody>
</table>
Prefixes

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

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

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dcore: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE
{
  <book1> dcore:title ?title
}
Basic Graph Pattern Matching

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:

<table>
<thead>
<tr>
<th>mbox</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:outlaw@example.com">mailto:outlaw@example.com</a></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>name</th>
<th>mbox</th>
</tr>
</thead>
<tbody>
<tr>
<td>„Alice“</td>
<td><a href="mailto:alice@example.com">mailto:alice@example.com</a></td>
</tr>
<tr>
<td>„Alice“</td>
<td><a href="mailto:alice@work.example">mailto:alice@work.example</a></td>
</tr>
<tr>
<td>„Bob“</td>
<td></td>
</tr>
</tbody>
</table>
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)


Thank You
Semantic Web and Multimedia

- Multimedia Ontology -

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

<name>

<education>

<work>

<private>

林冠英 根留台湾 可能者高

在受聘者热忱奔走之下，省府名望者林冠英根留台湾的可行性也提升了几分。院府总长林俊、国家委员林俊在日前提呈林冠英、石维友等//[换行]，并接日本等国交涉。此外，在台商法的政策区内，除振振及千年不变率新时，与八月十日起連省交，为期是第一期。

在台湾省省长的评论中，取之途径以採取行动来解决林冠英留台的问题上之一，晋台商法为同时林冠英根留台湾的可能性，而且几乎每期报告都提及林冠英根留台湾。

此外，林冠英上月对恩派斯馆主的「恩派斯馆主」，邀访了恩派斯馆主的三大友军以及「恩派斯馆主」、「恩派斯大记行」、「恩派斯大记行」。最后的D.A.T.也将在未来大有奥援，但未在恩派斯馆主的新闻中提及。

恩派斯国家馆主馆长布鲁尼日前也作出林冠英的指示，国家馆主馆长布鲁尼的指示是三大特色：一是控制材料的优劣程度；二是对恩派斯馆主的恩派斯馆主特点；三是宛如恩派斯馆主的恩派斯馆主。恩派斯馆主在恩派斯馆主材料，恩派斯馆主恩派斯馆主采用多联联音，恩派斯馆主恩派斯馆主。
What is the Problem with MPEG-7?

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*
Capabilities and Maturity Levels

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
How to overcome the drawbacks of MPEG-7?

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!
How to design a high Quality Multimedia Ontology?

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**
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

part

participant-in

endurant

perdurant

part

quality

abstract

physical-endurant

non-physical-endurant

physical-quality

temporal-quality

abstract-quality

region

inherent-in

located-in

physical-region
temporal-region

abstract-region
DOLCE Foundational Ontology

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

Information Encoding: ASCII-Code decimal

About: the White House

Situation: Securing the president


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)

- Concept
  - Parameter
    - Role
      - Course
        - Region
          - Role
            - Endurant
              - Situation
                - Method
                  - Expressed-by
                    - Social-object
                      - Requires
                        - Information-object
                          - Satisfies
                            - Description
                              - Played-by
                                - Setting
                                  - Valued-by
                                    - Parameter
                                      - Defines
                                        - Concept
                                          - Course
                                            - Sequences
                                              - Perdurant
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)
Decomposition Pattern

D&S / OIO

description
information-object
role
method
situation

structured-data-description
digital-data
processing-role
algorithm
segment-decomposition

localization-descriptor
multimedia-data
output-role
input-role
segmentation-algorithm

mask-role
output-segment-role
input-segment-role

plays
requires
plays
satisfies
defines

setting
Example

Image1 playsRole SegmentationInput

Segment1 playsRole SegmOutp
Segment3 playsRole SegmOutp
Segment2 playsRole SegmOutp
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, ...

---

**Diagram:**

- **Fundamental Knowledge about the World**
  - DOLCE
  - Descriptions & Situation
  - Ontology of Information Objects

- **Knowledge about a specific Domain**
  - Connected by Semantic Annotation Pattern
  - Domain Ontolog

- **Multimedia Knowledge (COMM)**
  - Datatype
  - Core
  - Localization
  - Visual
  - Audio
  - Text / LingInfo
  - Media
Does the Multimedia Ontology fulfil the Requirements?

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
Benefits compared to MPEG-7

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 **JRS-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)
References


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
Generation of Valid Annotations

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
Example: Generation of a Valid 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);
Example: Generation of a Valid Annotation

```java
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);
```
Example: Generation of a Valid Annotation

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(NumberOfCoefficientsEnumerationType.NUMBEROFCOEFFICIENTS_32);
scd.setNumOfBitplanesDiscarded(NumberOfBitplanesDiscardedEnumerationType.NUMBEROFBITPLANESDISCARDED_3);

id1.addVisualDescriptor(scd);
Export of the Annotation

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

```java
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

<table>
<thead>
<tr>
<th>Java COMM-API class</th>
<th>COMM concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image, Video, Text, …</td>
<td>image-data, video-data, text-data, …</td>
</tr>
<tr>
<td>StillRegion, VideoSegment, …</td>
<td>still-region-role, video-segment-role, …</td>
</tr>
<tr>
<td>MediaProfile</td>
<td>media-profile</td>
</tr>
</tbody>
</table>

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

```java
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());
Workflow for Sophisticated Filtering

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 {
  ?ID rdf:type comm:image-data .
}";
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
Reconstruct COMM-API Objects

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
Multimedia Annotation

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”
Tools

Web-based Tools
- flickr
- Riya

Stand-Alone Tools
- PhotoStuff
- AktiveMedia

Annotation for Feature Extraction
- KAT
flickr

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
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

Import content
Select for analysis

Select content for analysis

Select analysis plugin
Annotation queue

Analysed content appears in annotation queue for manual inspection.
Select ontology concept to retrieve images that are annotated with that concept. In this case the analysis detected persons in two images.
Double click to open in image annotation tool.
Adding instances

Adding instance to ontology for annotation.
Refine annotation

Drag&drop

Add annotation with instance.
Selection of instance now retrieves manually refined image.
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