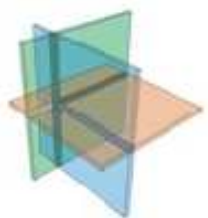


Semantic Web

- Multimedia Ontology-

Steffen Staab

<http://isweb.uni-koblenz.de>



UNIVERSITÄT
KOBLENZ · LANDAU



"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

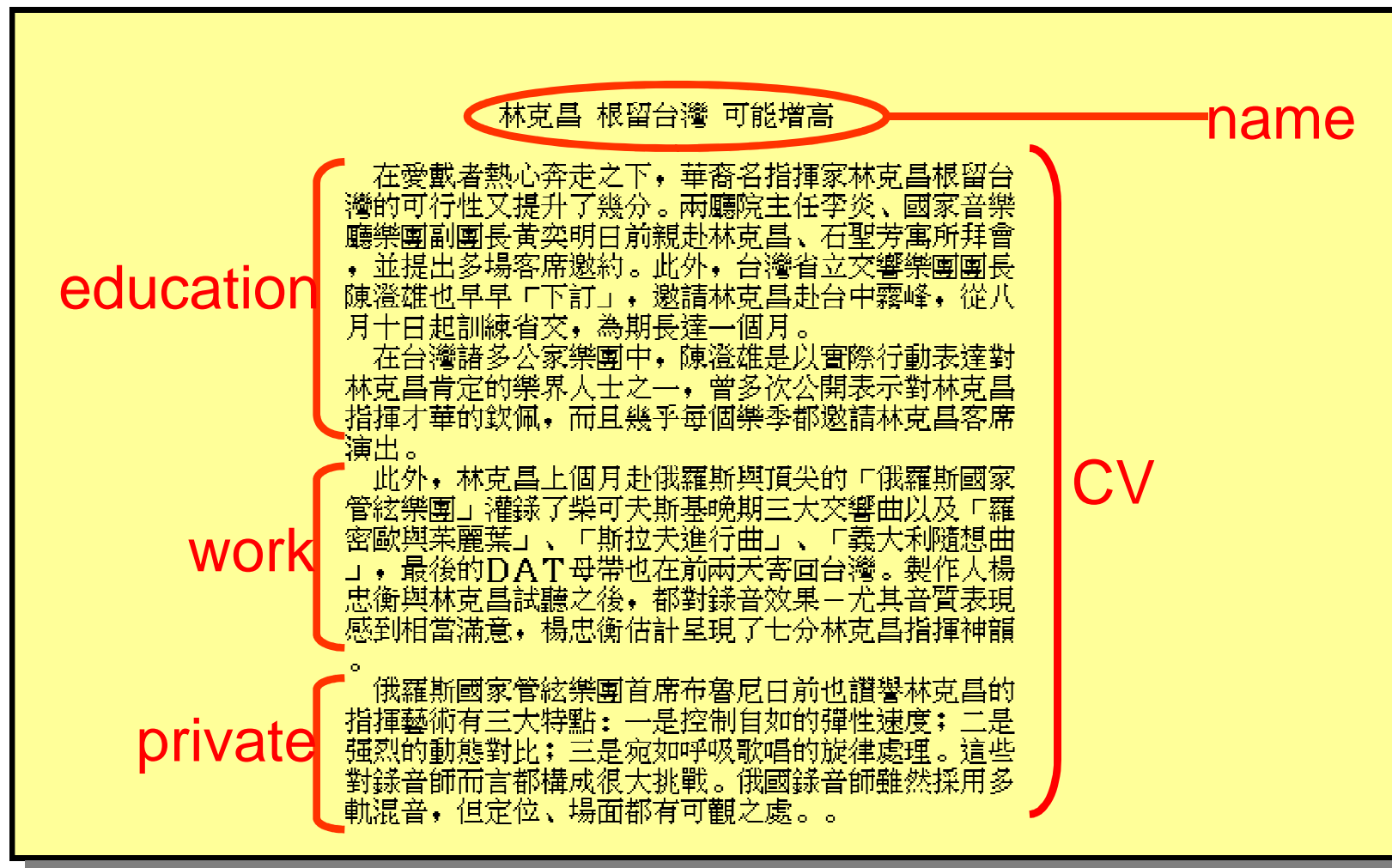
For ~99% of multimedia
people the answer for
content annotation is ...

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)



林克昌 根留台灣 可能增高

< ναμε >

<εδυχατιον>

在愛戴者熱心奔走之下，華裔名指揮家林克昌根留台灣的可行性又提升了幾分。兩廳院主任李炎、國家音樂廳樂團副團長黃奕明日前親赴林克昌、石聖芳寓所拜會，並提出多場客席邀約。此外，台灣省立交響樂團團長陳澄雄也早早「下訂」，邀請林克昌赴台中霧峰，從八月十日起訓練省交，為期長達一個月。

在台灣諸多公家樂團中，陳澄雄是以實際行動表達對林克昌肯定的樂界人士之一，曾多次公開表示對林克昌指揮才華的欽佩，而且幾乎每個樂季都邀請林克昌客席演出。

<ωορκ>

此外，林克昌上個月赴俄羅斯與頂尖的「俄羅斯國家管絃樂團」灌錄了柴可夫斯基晚期三大交響曲以及「羅密歐與茱麗葉」、「斯拉夫進行曲」、「義大利隨想曲」，最後的DAT母帶也在前兩天寄回台灣。製作人楊忠衡與林克昌試聽之後，都對錄音效果—尤其音質表現感到相當滿意，楊忠衡估計呈現了七分林克昌指揮神韻。

< Xς >

<πριωατε>

俄羅斯國家管絃樂團首席布魯尼日前也讚譽林克昌的指揮藝術有三大特點：一是控制自如的彈性速度；二是強烈的動態對比；三是宛如呼吸歌唱的旋律處理。這些對錄音師而言都構成很大挑戰。俄國錄音師雖然採用多軌混音，但定位、場面都有可觀之處。

```
<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>
          ...
        </SpatialDecomposition>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Segment 1

Segment 2

Segment 3

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

First Shot:

XQL: //StillRegion[.//Keyword="Sky"]

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

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**

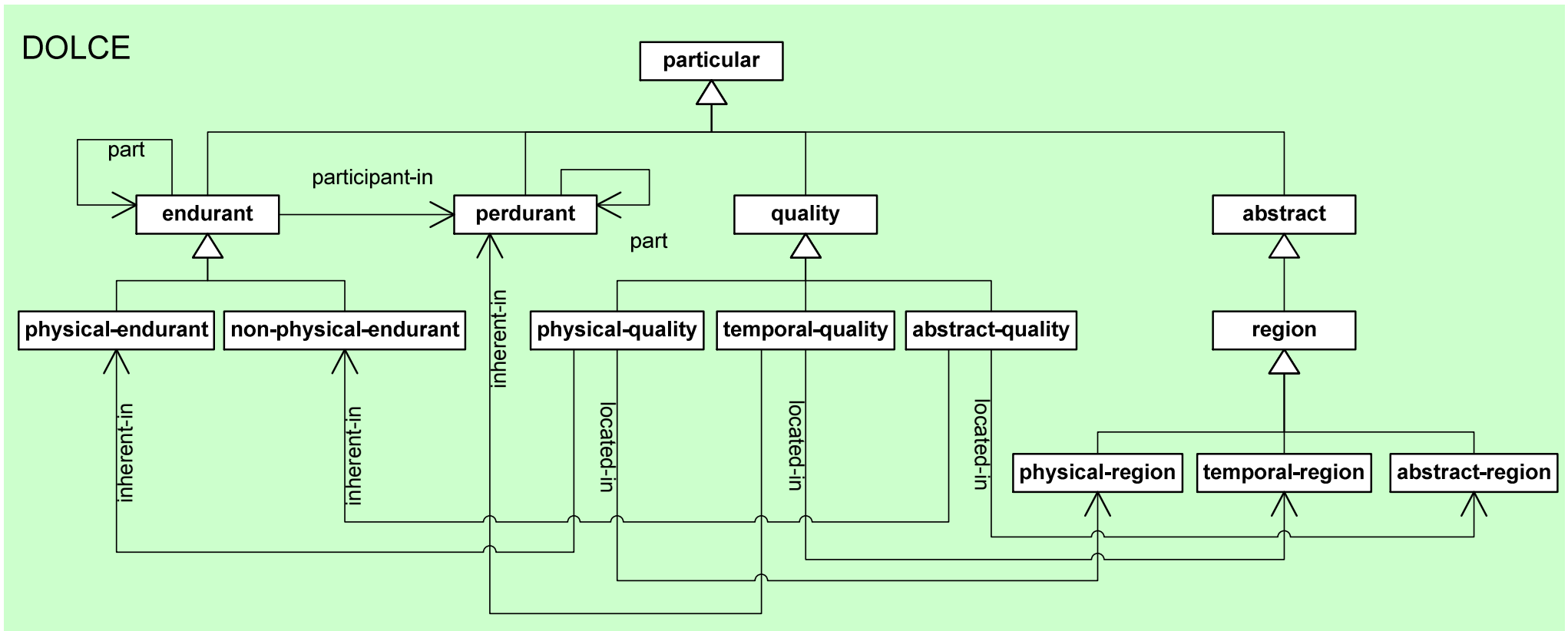
Identification of most important MPEG-7 functionalities

- ◆ **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*



4D world view centered around

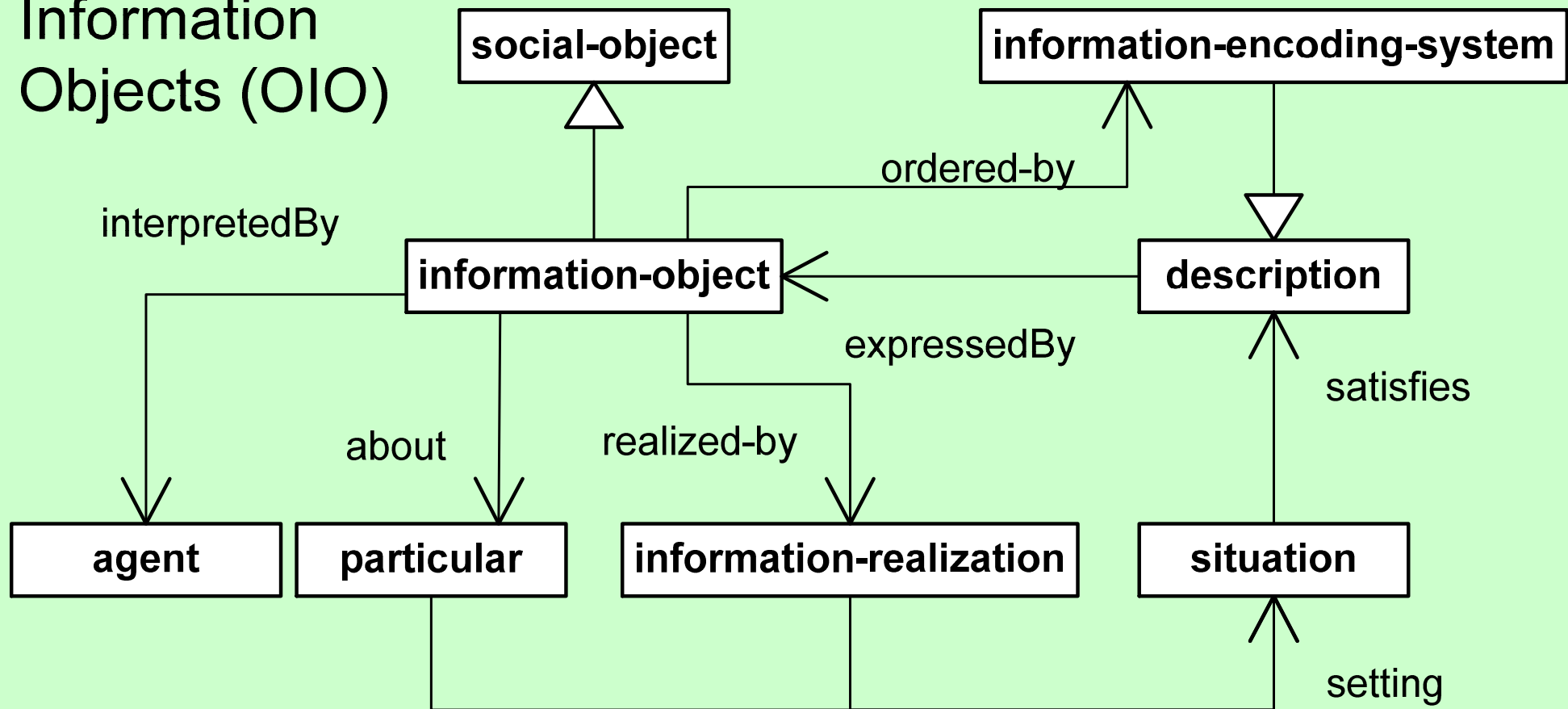
- ◆ **Endurants:** Independent wholes that exist in time and space
- ◆ **Perdurants:** Events, processes, phenomena, ...

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)



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

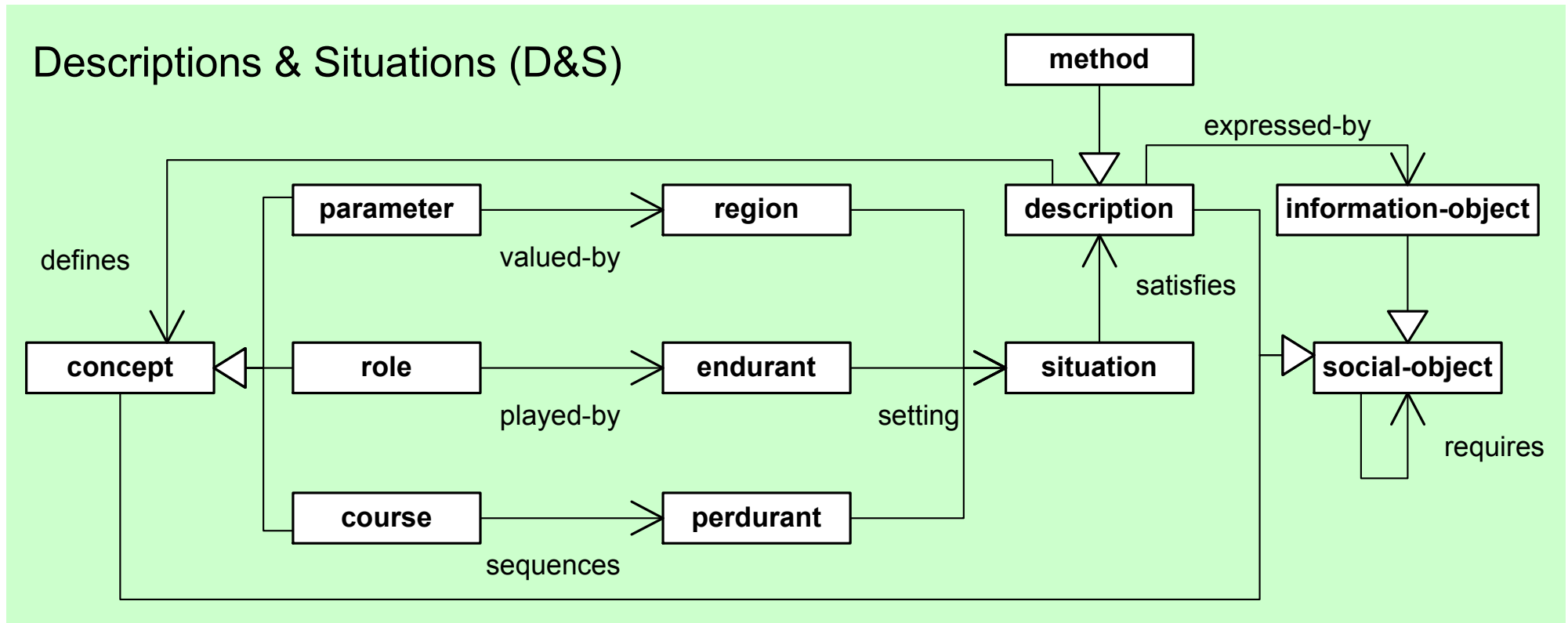
Formalization of information exchange

- ◆ 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



Distinction between:

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

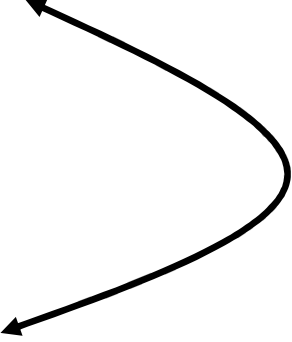
Descriptions

- ◆ 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!



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

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)

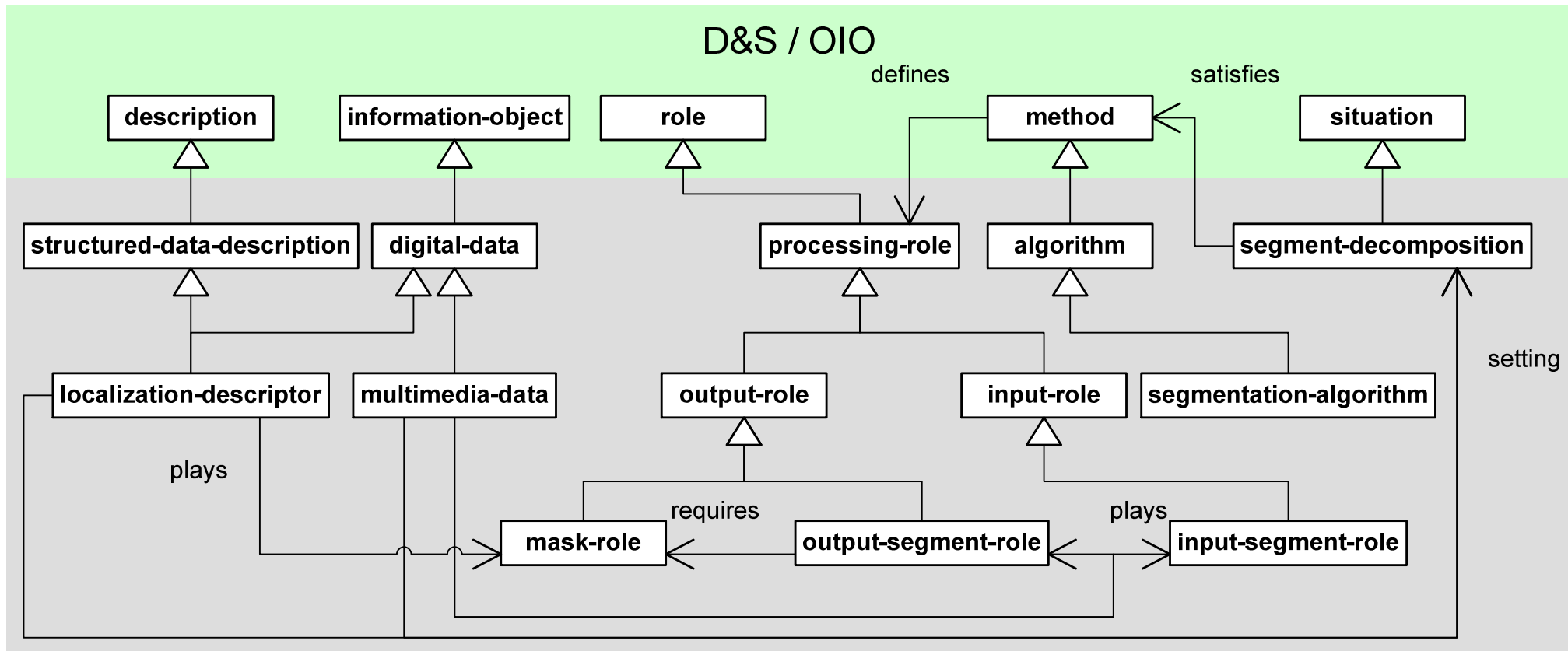




Image1 playsRole
SegmentationInput

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

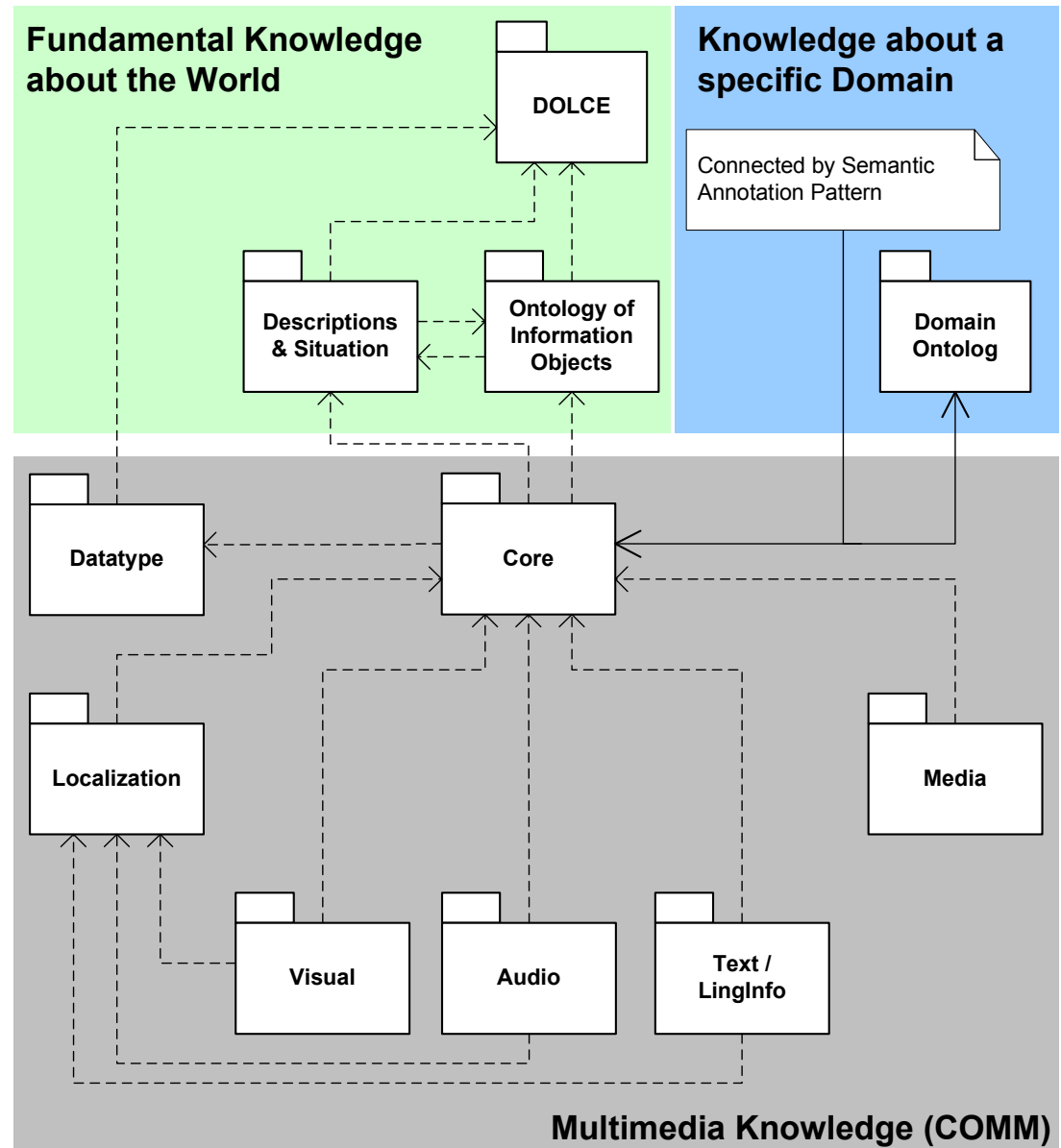
Usage of DOLCE enforces clean design

- ◆ 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 have been jointly extended → 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)

- [**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(
    NumberOfCoefficientsEnumerationType.NUMBEROFCOEFFICIENTS_32);
scd.setNumOfBitplanesDiscarded(
    NumberOfBitplanesDiscardedEnumerationType.NUMBEROFBITPLANESDISCARDED_3);

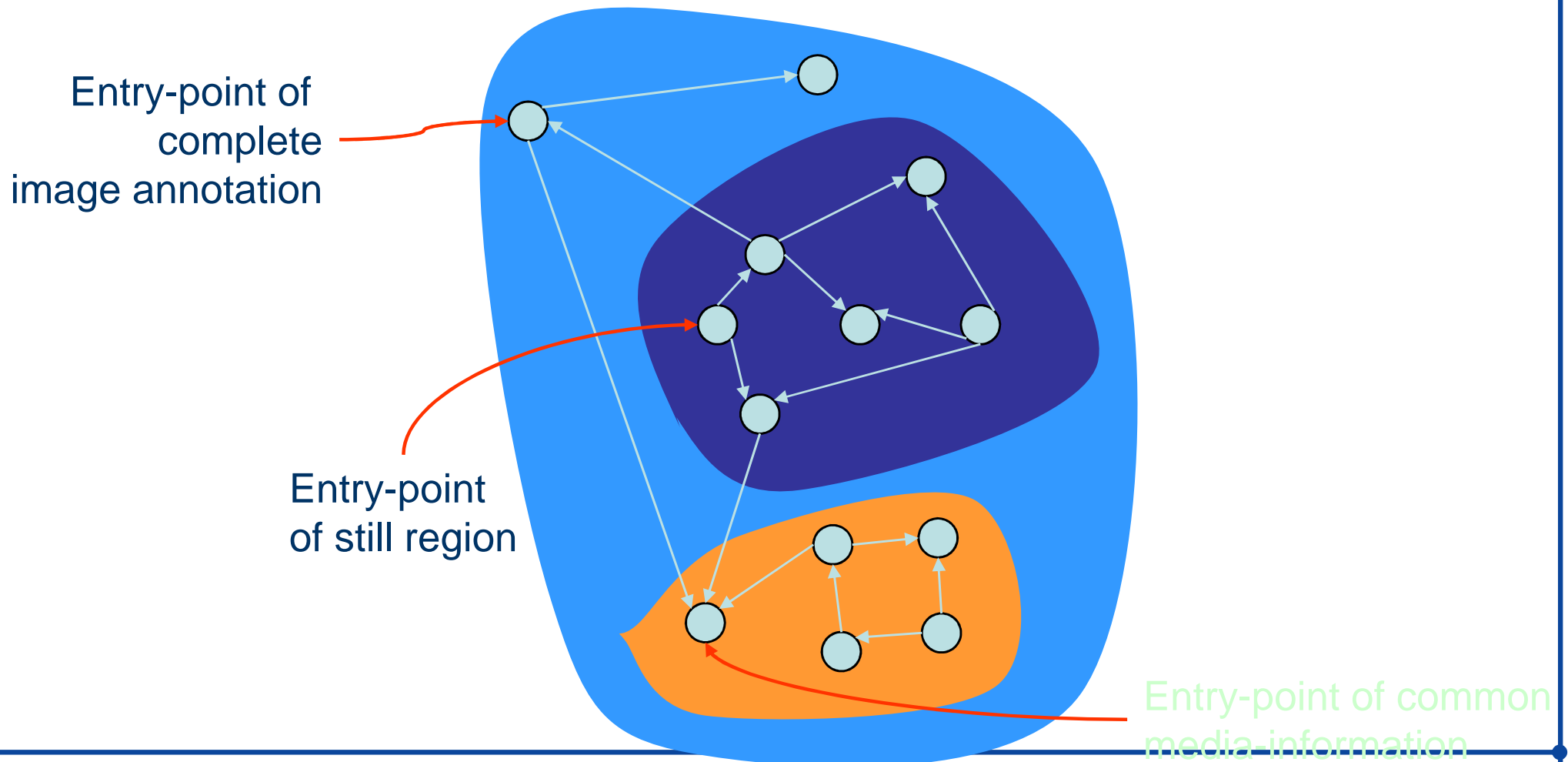
id1.addVisualDescriptor(scd);
```

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);
}
```