

Domain Knowledge Representation in Semantic MPEG-7 Descriptions

Chrisa Tsinaraki and Stavros Christodoulakis

1. Introduction

Some important developments in the area of multimedia applications lead to an open, Internet based environment, for multimedia. These developments include: **(a)** The popularity of the audiovisual content and its associated services (including the traditional TV and video services); **(b)** The advent of the digital media; **(c)** The availability of low-cost audiovisual content management devices; and **(d)** The development of advanced network infrastructures.

In such an open environment, the syntactic interoperability provided by the standards (such as the MPEG standards) is necessary for the services offered by different vendors to interoperate. However, since the amount of the available audiovisual content grows exponentially, efficient semantic-based retrieval services should be offered, in order to allow the users to effectively manage the audiovisual content. Such services can be built on top of the semantic-based MPEG-7 audiovisual content descriptions.

The *MPEG-7* standard [7], which is the dominant standard for audiovisual content description, provides interoperability at the syntactic level and, at the same time, allows for the semantic description of the audiovisual content. There is, though, a serious limitation of the MPEG-7 standard: The MPEG-7 constructs intended for the semantic description of the audiovisual content are general-purpose constructs and the standard does not describe a formal mechanism for the systematic integration of domain knowledge in the MPEG-7 descriptions. Thus, the utilization of the MPEG-7 semantic description constructs, even in conjunction with textual or keyword-based descriptions of the audiovisual content, has serious limitations [5]. Consider, as an example, a query asking for the audiovisual content containing the goals of a soccer game. This approach would return, in addition to the requested material, audiovisual content that contains, in its description, the word “goal” (e.g. “shot-on-goal”, “near-goal” etc.) while it does not contain goal.

It is well accepted today that the utilization of domain knowledge can improve the functionality and effectiveness of the information system applications. It can, for example, allow for reasoning on top of the content metadata descriptions and improve the efficiency of the content retrieval and filtering. The integration of domain knowledge in the metadata descriptions allows more precise querying on a semantic vocabulary which is well understood by the domain communities [2][6].

A straightforward solution for the systematic representation of domain knowledge in the MPEG-7 framework is the definition of XML Schema types that extend the general-purpose MPEG-7 types in order to represent domain-specific entities (e.g. goals in the soccer domain). Such an approach, though, causes a serious interoperability problem, since the extended types are not part of the MPEG-7 standard and the standard-based software will not be able to process them.

The domain knowledge is usually expressed today in the form of domain ontologies and, since the OWL language [17] is the dominant standard in domain knowledge description and the Semantic Web environment offers tools (e.g. reasoners) for OWL ontology processing, several domain ontologies have been expressed (and more are expected to be expressed) in OWL syntax. A Semantic-Web based methodology for the specification of audiovisual content descriptions that exploit domain knowledge includes the following steps: **(a)** The expression of the MPEG-7 semantics in OWL/RDF syntax, resulting in MPEG-7 ontologies [4] [13] [3] [8]; **(b)** The integration of the OWL/RDF MPEG-7 ontologies with the OWL domain ontologies [14] [15]; and **(c)** The specification of OWL/RDF audiovisual content descriptions based on the integrated MPEG-7 and domain ontologies. Unfortunately, these descriptions cannot be exploited by the MPEG-7 community, since the MPEG-7 based software cannot interpret them. Furthermore, the MPEG-7 based search and filtering services cannot take into account such descriptions.

It is therefore clear from the above paragraphs that the systematic integration of domain knowledge in the MPEG-7 descriptions is necessary for the support of efficient, semantic-based audiovisual content retrieval and filtering in the open environment formed in the Internet today. The specification of a formal model for domain knowledge representation using the MPEG-7 constructs is of paramount importance for exploiting domain knowledge in order to perform semantic processing of the audiovisual content. Without such a formal model the complete semantics of the descriptions will not be unambiguously understood and automatically processable by software across organizations.

In this chapter we present a formal model for domain knowledge representation within MPEG-7. The proposed model allows for the systematic integration of domain knowledge in the MPEG-7 descriptions using MPEG-7 constructs, thus maintaining interoperability with existing MPEG-7 based software. In particular, the proposed formal model for domain knowledge representation using MPEG-7 constructs achieves the following objectives:

- It presents clearly and unambiguously a way to integrate domain knowledge in MPEG-7 using exclusively MPEG-7 constructs. Therefore, all the descriptions produced are completely within the MPEG-7 standard.
- It describes clearly and formally the axioms that hold, and therefore it allows reasoning to be performed by distributed applications that utilize these axioms. This allows advanced functionality (such as for retrieval) for multimedia applications to be implemented and exploited in distributed environments.
- The representations and axioms of the formal model clearly map to corresponding representations and axioms of OWL. The subset of the OWL axioms that hold for the domain knowledge representation is clearly specified. This allows the transformation of the domain knowledge in OWL syntax, its integration in MPEG-7 based ontological infrastructures like the one of the DS-MIRF framework [13] [14] [15], and the use of the existing OWL reasoners for semantic processing.

The model for domain knowledge representation using MPEG-7 constructs that we present here is a formal logic-based extension of the informal model we have developed in our previous research [12] [16], which essentially allowed only the representation of taxonomies using MPEG-7 syntax.

The rest of the paper is organized as follows: The proposed MPEG-7 based domain knowledge representation model is introduced in section 2 and is detailed in sections 3-7. The exploitation of the domain knowledge that is represented according to the proposed model is presented in section 8 and the paper concludes in section 9.

2. MPEG-7 Based Domain Knowledge Representation

In this section we describe our formal model for domain knowledge representation using MPEG-7 constructs. In our model, domain knowledge is usually represented by domain ontologies. Every domain ontology DO may be expressed, in MPEG-7 syntax, as a domain ontology MP7DO. To do this, the domain expert utilizes the general-purpose MPEG-7 semantic description constructs, which are a set of general-purpose XML Schema types, rooted at the *SemanticBaseType* of MPEG-7, that allow for the description of the audiovisual content semantics.

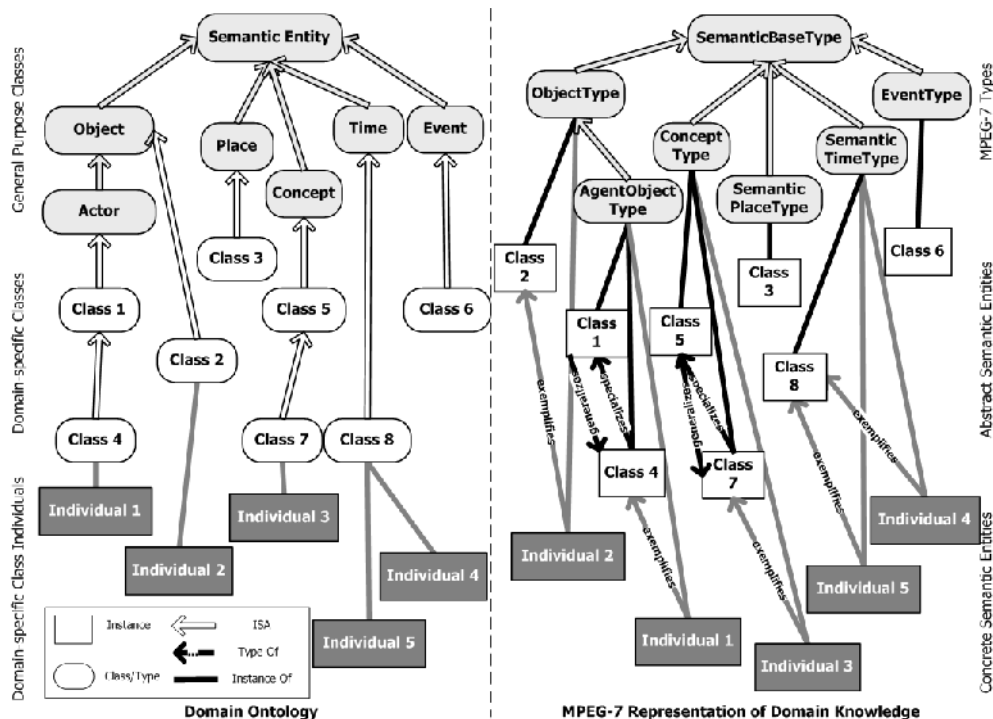


Figure 1: The Domain Knowledge Representation Model

Our formal model utilizes exclusively MPEG-7 constructs to describe domain knowledge, and therefore it remains strictly within the MPEG-7 standard.

The model, depicted in Figure 1, is based on the MPEG-7 relationships and on the capability of defining both abstract and concrete MPEG-7 semantic entities (essentially agents, objects, concepts, events, places, times and states). As shown in Figure 1, in to our model the general-purpose classes are represented by the standard MPEG-7 types, the domain specific classes are represented by abstract

MPEG-7 semantic entities and the individuals are represented by concrete MPEG-7 semantic entities. The MPEG-7 descriptions may utilize the domain knowledge structured according to the proposed model through references to the (abstract and concrete) semantic entities comprising it. Notice that both the abstract and the concrete semantic entities are defined at the MPEG-7/XML document level and not at the XML Schema level. Thus, the domain knowledge can be systematically represented and, at the same time, full compatibility with the MPEG-7 standard is maintained.

Table 1 provides an overview of the proposed model for domain knowledge representation using MPEG-7 constructs. In particular, the first column contains the ontology constructs modeled and the second column shows the MPEG-7 constructs used for their representation.

Ontology Construct	MPEG-7 Representation
Ontology Declaration	“Description” Element of type “SemanticDescriptionType”
Property	“Property” Element / Pair of “Relation” Elements of type “property”-“propertyOf”/ “AttributeValuePair” Element
Class	“SemanticBase” Element with “AbstractionLevel /@dimension” = 1
Individual	“SemanticBase” Element with “AbstractionLevel /@dimension” = 0
Subsumption	Pair of “Relation” Elements of type “exemplifies”-“exemplifiedBy”
Class Generalization / Specialization	Pair of “Relation” Elements of type “specializes”-“generalizes”
Property Value Restriction	“Term” Element / Pair of “Relation” Elements of type “Relation” of type “property”-“propertyOf”
Class Equivalence	“Relation” Element of type “equivalent”
Disjointness	“Relation” Element of type “disjoint”
Equivalence of Individuals	“Relation” Element of type “equals”
Separation of Individuals	“Relation” Element of type “separated”
Class Union	“Relation” Element of type “union”
Class Intersection	“Relation” Element of type “intersection”

Table 1: Overview of the Domain Knowledge Representation Model

As is shown in Table 1, the proposed model describes clearly and formally the axioms that hold, essentially a subset of the semantics of OWL. Thus, the semantics of the proposed model are mapped to the OWL semantics. As a consequence, every domain ontology MP7DO expressed according to our model, can be also expressed in OWL syntax as an OWL ontology ODO. Thus, the existing OWL reasoners can be used with ODO, making this way possible the semantic processing of the semantics of MP7DO. This allows the transformation of the domain knowledge in OWL syntax, its integration in MPEG-7 based ontological infrastructures like the one of the DS-MIRF framework [13] [14] [15], and the use of the existing OWL reasoners for semantic processing.

The domain knowledge representation model that utilizes MPEG-7 constructs is detailed in the following sections: The domain ontology representation is described in section 3, the representation of properties is described in section 4, the representation of classes is described in section 5, the representation of individuals is described in section 6 and the representation of axioms is described in section 7.

3. Domain Ontology Representation

In this section we describe the representation of domain ontologies using MPEG-7 constructs. Let DO be a domain ontology. We describe DO in a regular expression form, independent on the knowledge representation language in which it may have been expressed (thus allowing the application of the proposed model to domain ontologies expressed using different syntax):

$$DO(do_name, label, comment, imported_ontologies, classes, properties, individuals, relationships) \quad (1)$$

where:

- `do_name` is the name of DO.
- `label` is the (optional) label of DO.
- `comment` is an (optional) comment describing DO.
- `imported_ontologies` is a set, comprised of the ontologies imported in DO.
- `classes` are the classes of DO.
- `properties` are the properties of DO.
- `individuals` are the individuals of DO.

- relationships are the relationships defined in DO.
- axioms are the axioms defined in DO. Such axioms may specify class and property hierarchies, equivalence and difference relationships for classes, properties and individuals as well as value, type and cardinality restrictions (details on axioms are provided in section 7).

The domain ontology DO is represented, in MPEG-7 syntax by an MPEG-7 domain ontology MP7DO that is implemented by a “Description” element D, of type “SemanticDescriptionType”. The D element has a “Semantics” element S, of type “SemanticType”, formally described in regular expression 2:

$$S(id, alevel, l, d, iop, sop, p, sb, r) \quad (2)$$

where:

- id is the identity of S, is represented by the “id” attribute and has do_name as value.
- alevel is the value of the “dimension” attribute of the “AbstractionLevel” element of S and has 1 as value, in order to express that the current description is abstract.
- l is the label of S and is represented by the value of the “Name” subelement of the “Label” element of S. If DO has a label, l has label as value and if it does not l has do_name as value.
- d is the (optional) description of S. d is defined if a comment describing DO exists, is represented by the value of the “FreeTextAnnotation” element of a “Definition” element defined in S and has comment as value.
- iop is the set of the declarations of the ontologies imported in DO.
- sop is the declaration of DO.
- sb is the set of the “SemanticBase” elements of S, which represent the classes and the individuals of DO (see sections 5 and 6 for details).
- p is the set of the “Property” elements of S, which represent properties of DO.
- r is the set of the “Relation” elements of S, which represent properties and relationships of DO.

As an example, consider a soccer ontology having the “soccer” identity. This ontology is represented by the “Description” element, of type “SemanticDescriptionType”, shown in Figure 2.

```

<Description xsi:type="SemanticDescriptionType">
  <Semantics id="soccer">
    <AbstractionLevel dimension="1"/>
    <Label>
      <Name>Soccer Ontology</Name>
    </Label>
    <Definition>
      <FreeTextAnnotation>OWL Ontology for Soccer</FreeTextAnnotation>
    </Definition>
    <Property>
      <Name>Ontology Self</Name>
      <Definition>socceragents</Definition>
      <Term>
        <Name>href</Name>
        <Definition>"http://soccer.org/socceragents#"</Definition>
      </Term>
    </Property>
    ...
  </Semantics>
</Description>

```

Figure 2: Soccer Ontology representation in MPEG-7 syntax

Ontology Declaration Representation. A domain ontology DO contains a domain ontology declaration that refers to itself and a set of domain ontology declarations for the domain ontologies imported in DO. Every domain ontology declaration OD contained in DO is represented by a “Property” element, which is defined in the S element of D (where D is the representation of DO) and is formally described by regular expression 3:

$$OD(name, def, uri) \quad (3)$$

where:

- name is the value of the “Name” element of OD and its value is “Ontology” if OD is an ontology imported in DO and “Ontology Self” if OD is the declaration of DO.
- def is the value of the (optional) “Definition” element defined in OD and has as value the name of the (optional) XML entity that represents the ontology declared in OD.
- uri is a “Term” element that represents the URI of the ontology declared in OD and is formally described by regular expression 4:

$$uri(tname, tdef) \tag{4}$$

where:

- tname is the value of the “Name” element defined in uri and has “href” as value.
- tdef is the value of the “Definition” element defined in uri and has as value the URI of the ontology declared by OD.

For example, the declaration of the soccer ontology of Figure 2 is represented by the “Property” element that has as value of its “Name” element “Ontology Self”.

4. Property Representation

The representation of the properties defined in a domain ontology DO using MPEG-7 constructs is detailed in this section. The domain-specific properties of a domain ontology DO are represented in MPEG-7 syntax: **(a)** By “Property” elements, if they are simple type properties or are of type “InlineTermDefinitionType” (or of an MPEG-7 type extending it); **(b)** By “Relation” elements, if they are complex type properties; and **(c)** By “AttributeValuePair” elements, if they have as domains classes that represent states and either are simple type properties or have as range one of the types “IntegerMatrixType”, “FloatMatrixType”, “TextualType” (or an MPEG-7 type extending it), “TextAnnotationType” (or an MPEG-7 type extending it), “ControlledTermUseType” (or an MPEG-7 type extending it) and “DType” (or an MPEG-7 type extending it).

Let P be a property defined in the domain ontology DO that is described in regular expression 5:

$$P(p_id, range, domain, value, label, comment) \tag{5}$$

where:

- p_id is the identity of P.
- domain is the domain of P.
- range is the range of P.
- value is the (optional) fixed value of P.
- label is the (optional) label of P.
- comment is the (optional) description of P.

Property Representation by “Property” Elements. Let prop be a “Property” element, which represents the P property of DO. prop is formally described by regular expression 6:

$$prop(name, type, fixed) \tag{6}$$

where:

- name is the name of P, is represented by the “Name” element of prop and has p_id as value.
- type is the type of P and is represented by the “Definition” element of prop.
- fixed is the representation of a fixed value axiom on P so that it has value as value.

Consider as an example that, in the soccer ontology of Figure 2, a “DateOfBirth” property, of type “Date”, has been defined for the soccer players. The MPEG-7 representation of the “DateOfBirth” property is shown in Figure 3.

```

<Property>
  <Name>DateOfBirth</Name>
  <Definition>Date</Definition>
</Property>
```

Figure 3: Representation of the “DateOfBirth” Property

Property Representation by “Relation” Elements. Let P a property defined in the domain ontology DO, which is represented by “Relation” elements. P is represented by a pair of “Relation”

elements for each of its domains, the elements `pr_relationship` and `prOf_relationship`, which are described by the regular expressions 7 and 8.

$pr_relationship(pr_type, pr_source, pr_target, pr_name)$ (7)

$prOf_relationship(por_type, por_source, por_target, por_name)$ (8)

The following hold for the `pr_relationship` element, which is formally described in regular expression 7:

- `pr_type` is the type of `pr_relationship` and has “property” as value.
- `pr_source` is the source of `pr_relationship` and has the property domain as value.
- `pr_target` is the target of `pr_relationship` and has the property range as value.
- `pr_name` is the name of P and has `p_id` as value.

The following hold for the `prOf_relationship` element, which is formally described in regular expression 8:

- `por_type` is the type of `prOf_relationship` and has “propertyOf” as value.
- `por_source` is the source of `prOf_relationship` and has the property range as value.
- `por_target` is the target of `prOf_relationship` and has the property domain as value.
- `por_name` is the name of P and has `p_id` as value.

Consider as an example that, in the soccer ontology of Figure 2, a “PlaceOfBirth” property has been defined for the soccer players (who are represented by the “SoccerPlayer” class), which associates them with the places they were born in (such places are represented by the “City” class). The MPEG-7 representation of the “PlaceOfBirth” property is shown in Figure 4.

```

<Relation type="property" source="#SoccerPlayer" target="#City">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation>PlaceOfBirth</FreeTextAnnotation>
    </Comment>
  </Header>
</Relation>
<Relation type="propertyOf" source="#City" target="#SoccerPlayer">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation>PlaceOfBirth</FreeTextAnnotation>
    </Comment>
  </Header>
</Relation>

```

Figure 4: Representation of the “PlaceOfBirth” Property

Property Representation by “AttributeValuePair” Elements. Let AVOP be an instance of the “AttributeValuePair” element that represents the P property of the domain ontology DO and is formally described in regular expression 9:

$AVOP(name, def, type, fixed)$ (9)

where:

- `name` is the value of the “Name” element of the “Attribute” element of AVOP and has `p_id` as value.
- `def` is the value of the (optional) “Definition” element of the “Attribute” element of AVOP and has `comment` as value.
- `type` is the value of an instance of the “TextValue” element of AVOP and has as value the identity of the domain of P.
- `fixed` is an (optional) element defined in AVOP, that represents a fixed value axiom on P so that P has `value` as value.

Consider as an example that in a soccer tournament every soccer team must start the game with eleven players and that the initial number of players is represented by the “InitialNumOfPlayers” property, of value 11, of the “TournamentSoccerTeamGameState” class, which represents the state of a soccer team that participates in the tournament. The MPEG-7 representation of the “InitialNumOfPlayers” property is shown in Figure 5.

```

<AttributeValuePair>
  <Attribute>
    <Name>InitialNumOfPlayers</Name>
  </Attribute>
  <TextValue>integer</TextValue>
  <IntegerValue>11</IntegerValue>
</AttributeValuePair>

```

Figure 5: Representation of the “InitialNumOfPlayers” Property

4.1 Property Value Representation

The property values of the individuals that are defined in a domain ontology DO are represented, in accordance with the representation of the corresponding properties, by “Property” elements, “Relation” elements and “AttributeValuePair” elements.

Property Value Representation by “Property” Elements. Let pr be a “Property” element that represents a property value defined in the domain ontology DO, which is described in regular expression 10:

$$pr(name, value) \quad (10)$$

where:

- name is the property name and is represented by the value of the “Name” element of pr .
- value is the property value and is represented by the value of the “Definition” element of pr .

Consider as an example that the “Ronaldinho” individual, which represents the soccer player Ronaldinho, exists in the soccer ontology of Figure 2 and that the value of its “DateOfBirth” property is “21/03/1980”. The MPEG-7 representation of the property value is shown in Figure 6.

```

<Property>
  <Name>DateOfBirth</Name>
  <Definition>21/03/1980</Definition>
</Property>

```

Figure 6: Representation of the Value of the “DateOfBirth” Property

Property Value Representation by “Relation” Elements. Let Ind be an individual having a property P defined in the domain ontology DO and that the value of P is represented by a pair of “Relation” elements. The property value is represented by the pair of “Relation” elements $pr_relationship$ and $prOf_relationship$, which are formally described by the regular expressions 11 and 12.

$$pr_relationship(pr_type, pr_source, pr_target, pr_name) \quad (11)$$

$$prOf_relationship(por_type, por_source, por_target, por_name) \quad (12)$$

The following hold for the $pr_relationship$ element, which is formally described in regular expression 11:

- pr_type is the type of $pr_relationship$ and has “property” as value.
- pr_source is the source of $pr_relationship$ and has the identity of Ind as value.
- pr_target is the target of $pr_relationship$ and has as value the identity of the property value.
- pr_name is the name of P .

The following hold for the $prOf_relationship$ element, which is formally defined in regular expression 12:

- por_type is the type of $prOf_relationship$ and has “propertyOf” as value.
- por_source is the source of $prOf_relationship$ and has as value the identity of the property value.
- por_target is the target of $prOf_relationship$ and has the identity of Ind as value.
- pr_name is the name of P .

Consider as an example that the “Ronaldinho” individual has the “PortoAllegre” as value of its “PlaceOfBirth” property. The MPEG-7 representation of the property value is shown in Figure 7.

```

<Relation type="property" source="#Ronaldinho" target="#PortoAllegre">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation>PlaceOfBirth</FreeTextAnnotation>
    </Comment>
  </Header>
</Relation>
<Relation type="propertyOf" source="#PortoAllegre" target="#Ronaldinho">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation>PlaceOfBirth</FreeTextAnnotation>
    </Comment>
  </Header>
</Relation>

```

Figure 7: Representation of the Value of the “PlaceOfBirth” Property

Property Value Representation by “AttributeValuePair” Elements. Let *Ind* be an individual having a property *P* defined in the domain ontology *DO* and that the value of *P* is *V* and is represented by an “AttributeValuePair” element. *V* is represented by the value of the appropriate element of “AttributeValuePair” (according to its type) and in particular: **(a)** The value of the element “BooleanValue” if *V* is a boolean value; **(b)** The value of the element “IntegerValue” if *V* is an integer value; **(c)** The value of the element “FloatValue” if *V* is a float number value; **(d)** The value of the element “TextValue” if *V* is a string value or a value of type “TextualType” (or of an MPEG-7 type that extends it); **(e)** The value of the element “IntegerMatrixValue” if *V* is of type “IntegerMatrixType”; **(f)** The value of the element “FloatMatrixValue” if *V* is of type “FloatMatrixType”; **(g)** The value of the element “TextAnnotationValue” if *V* is of type “TextAnnotationType” (or of an MPEG-7 type that extends it); **(h)** The value of the element “ControlledTermUseValue” if *V* is of type “ControlledTermUseType” (or of an MPEG-7 type that extends it); **(i)** The value of the element “DescriptorValue” if *V* is of type “DType” (or of an MPEG-7 type that extends it).

```

<AttributeValuePair>
  <Attribute>
    <Name>Status</Name>
  </Attribute>
  <TextAnnotationValue id="professional">
    <FreeTextAnnotation id="profFTA">
      Professional
    </FreeTextAnnotation>
  </TextAnnotationValue>
</AttributeValuePair>

```

Figure 8: Representation of the Value of the “Status” Property of a professional soccer team

Consider as an example that every soccer team has a status, which expresses its professionalism level. The soccer team status is represented in the soccer ontology of Figure 2 by the “Status” property of the “SoccerTeamState” class, which represents the soccer team state. The “Status” property may have the value “professional” if it refers to a professional soccer team, the value “semiprofessional” if it refers to a semi-professional soccer team or the value “amateur” if it refers to an amateur soccer team. The MPEG-7 representation of the value of the “Status” property of a professional soccer team is shown in Figure 7.

5. Class Representation

We present in this section the MPEG-7 representation of the classes defined in domain ontologies. The domain ontology classes are represented, according to our formal model, by abstract MPEG-7 semantic entities.

Consider as an example that the “SoccerPlayer” and “Goalkeeper” classes represent, respectively, the soccer players and the goalkeepers in the soccer ontology of Figure 2 and that “Goalkeeper” is a subclass of “SoccerPlayer”. Figure 9 shows the representation of the ontology classes using MPEG-7 constructs. As shown in Figure 9, both “SoccerPlayer” and “Goalkeeper” are represented by abstract semantic entities of type “AgentObjectType”, associated by a pair of “generalizes”/“specializes” relationships. These relationships express that “Goalkeeper” is a subclass of “SoccerPlayer”.

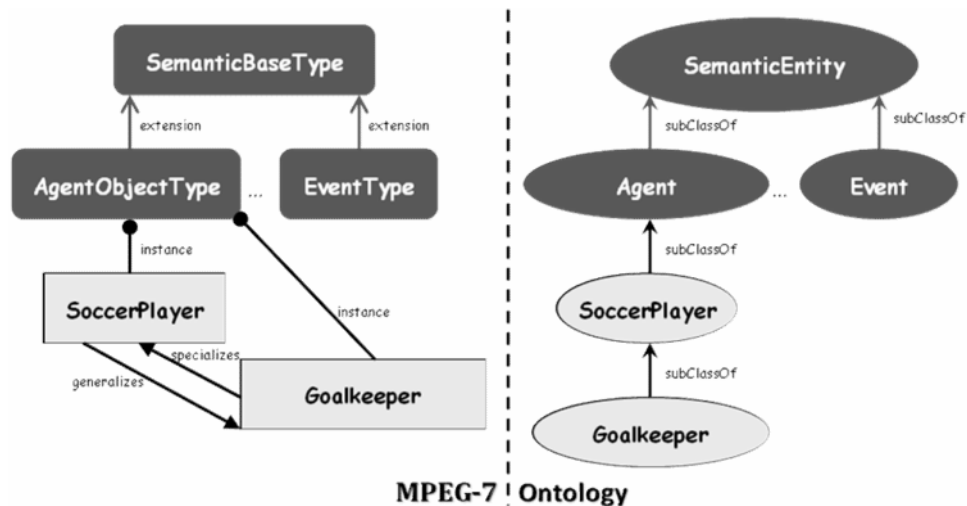


Figure 9: Representation of the “SoccerPlayer” and “Goalkeeper” Classes according to the Domain Knowledge Representation Model using MPEG-7 Constructs (left side) and in the original Ontology (right side)

The MPEG-7 representation of the “SoccerPlayer” and “Goalkeeper” classes is shown in Figure 10.

```

<SemanticBase xsi:type="AgentObjectType" id="SoccerPlayer">
  <AbstractionLevel dimension="1"/>
  <Label>
    <Name>Soccer Player</Name>
  </Label>
  ...
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="Goalkeeper">
  <AbstractionLevel dimension="1"/>
  <Label>
    <Name>Goalkeeper Object</Name>
  </Label>
  <Relation type="specializes" source="#Goalkeeper" target="#SoccerPlayer"/>
  <Relation type="generalizes" source="#SoccerPlayer" target="#Goalkeeper"/>
</SemanticBase>

```

Figure 10: Representation of the “SoccerPlayer” and “Goalkeeper” Classes

Let C be a class defined in a domain ontology DO that is described in regular expression 13:

$$C(cid, superclass, subclasses, label, comment, MPEG7_type, properties, relationships) \quad (13)$$

where:

- cid is the identity of C. For example, the “SoccerPlayer” class has “SoccerPlayer” as identity.
- superclass is the MPEG-7 representation of the superclass of C. For example, the “SoccerPlayer” class has as superclass the value “AgentObjectType” and the “Goalkeeper” class has as superclass the value “SoccerPlayer”.
- subclasses is the set of the subclasses of C.
- label is the (optional) label of C.
- comment is an (optional) description of C.
- MPEG7_type is the identity of the MPEG-7 type that represents the closest general-purpose MPEG-7 concept of C. For example, both “SoccerPlayer” and “Goalkeeper” have as MPEG7_type the type “AgentObjectType”. Notice that for the classes extending a general-purpose MPEG-7 concept superclass = MPEG7_type.
- properties are the properties of C.
- relationships are the relationships of C.

C is represented by the abstract semantic entity AI, formally described in regular expression 14:

$AI(ai_id, label, type, abstraction_level, spec_relationship, gen_relationships, property_elements, pr_relationships, prOf_relationships, exBy_relationships, relationships)$ (14)

where:

- ai_id is the identity of the abstract semantic entity AI, is represented by the “id” attribute and has cid as value. For example, the abstract semantic entity “SoccerPlayer” defined in Figure 10 has “SoccerPlayer” as identity.
- $label$ is a label that describes AI and is represented by the “Label” element. For example, the abstract semantic entity “SoccerPlayer” defined in Figure 10 has “Soccer Player” as label.
- $type$ is the MPEG-7 type having AI as an instance, is represented by the “type” attribute and has $MPEG7_type$ as value. For example, the abstract semantic entities “SoccerPlayer” and “Goalkeeper” defined in Figure 10 have “AgentObjectType” as type.
- $abstraction_level$ expresses that AI is an abstract semantic entity and is represented by the “dimension” attribute of the “AbstractionLevel” element, which has a value greater than 0. For example, the abstract semantic entities “SoccerPlayer” and “Goalkeeper” defined in Figure 10 have $AbstractionLevel.dimension=1$.
- $spec_relationship$ is an MPEG-7 relationship of type “specializes”, which is defined only if $superclass \quad MPEG7_type$ and associates the abstract semantic entity AI with its superclass if the later is a domain specific class.
- $gen_relationships$ is the set of the MPEG-7 relationships, of type “generalizes”, that associate the abstract semantic entity AI with the abstract semantic entities that represent its subclasses.
- $property_elements$ is the set of the “Property” elements of AI, which represent properties of C.
- $pr_relationships$ is the set of the MPEG-7 relationships, of type “property”, which represent complex type properties of C.
- $prOf_relationships$ is the set of the MPEG-7 relationships, of type “propertyOf”, that associate the abstract semantic entity AI with the abstract semantic entities that represent classes with properties having C as domain.
- $exBy_relationships$ is the set of the MPEG-7 relationships, of type “exemplifiedBy”, that associate the abstract semantic entity AI with the concrete semantic entities that represent the individuals belonging to C.
- $relationships$ is the set of the relationships of C.

6. Representation of Individuals

We describe in this section the representation of individuals defined in domain ontologies using MPEG-7 constructs. The domain ontology individuals are represented, according to our formal model, by concrete MPEG-7 semantic entities.

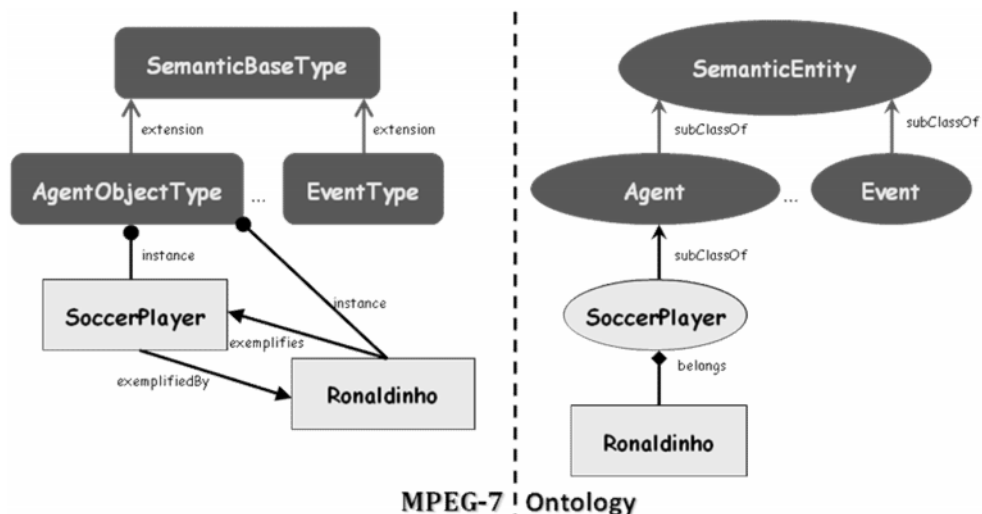


Figure 11: Representation of the Individual “Ronaldo” according to the Domain Knowledge Representation Model using MPEG-7 Constructs (left side) and in the original Ontology (right side)

Consider as an example the “Ronaldo” individual, which represents the soccer player Ronaldo. The representations of the “Ronaldo” individual, according to the domain knowledge representation model using MPEG-7 constructs and in the original ontology are depicted in Figure 11. Notice that the “Ronaldo” individual is represented by a concrete semantic entity of type “AgentObjectType”, which is associated with the abstract semantic entity “SoccerPlayer” that represents the class of the soccer players through a pair of “exemplifies”/“exemplifiedBy” relationships.

The representation of the “Ronaldo” individual in MPEG-7 syntax is shown in **Figure 12**.

```

<SemanticBase xsi:type="AgentObjectType" id="Ronaldo">
  <AbstractionLevel dimension="0"/>
  <Label>
    <Name>Ronaldo</Name>
  </Label>
  <Definition>
    <FreeTextAnnotation>Ronaldo</FreeTextAnnotation>
  </Definition>
  <Property>
    <Name>DateOfBirth</Name>
    <Definition>21/03/1980</Definition>
  </Property>
  <Relation type="exemplifies" source="#Ronaldo" target="#SoccerPlayer"/>
  <Relation type="exemplifiedBy" source="#Goalkeeper"
target="#SoccerPlayer"/>
  <Agent xsi:type="PersonType" id="RonaldoPerson">
    <Name>
      <GivenName>Ronaldo</GivenName>
    </Name>
  </Agent>
</SemanticBase>

```

Figure 12: Representation of the Individual “Ronaldo”

Let Ind be an individual that belongs to C and C be a domain specific class defined in the domain ontology DO . Ind is formally described in regular expression 15:

$$Ind(ind_id, class, MPEG7_type, properties, relationships, label, comment) \quad (15)$$

where:

- ind_id is the identity of the Ind individual.
- $class$ is the identity of C .
- $MPEG7_type$ is the identity of the MPEG-7 type that represents the closest general-purpose MPEG-7 concept of the class C where Ind belongs.
- $properties$ are the properties of Ind .
- $relationships$ are the relationships of Ind .
- $label$ is the (optional) label of Ind .
- $comment$ is an (optional) description of Ind .

The Ind individual is represented by the concrete semantic entity CI , formally described in regular expression 16:

$$CI(ci_id, label, type, abstraction_level, property_elements, pr_relationships, prOf_relationships, ex_relationship, relationships) \quad (16)$$

where:

- ci_id is the identity of CI and has ind_id as value.
- $label$ is a label that describes CI .
- $type$ is the MPEG-7 type, having CI as an instance, is represented by the “type” attribute and has $MPEG7_type$ as value.
- $abstraction_level$ has a value of 0 and expresses that CI is a concrete semantic entity.

- `property_elements` is the set of the “Property” elements of CI that represent properties of Ind.
- `pr_relationships` is the set of the MPEG-7 relationships, of type “property”, which represent complex type properties of Ind.
- `prOf_relationships` is the set of the MPEG-7 relationships, of type “propertyOf”, which associate the concrete semantic entity CI with the concrete semantic entities having a property with Ind as value.
- `ex_relationship` is an MPEG-7 relationship of type “exemplifies” that associates the concrete semantic entity CI with the abstract semantic entity that represents C. `ex_relationship` is defined only if class `MPEG7_type`.
- `relationships` is the set of the relationships of Ind.

7. Representation of Axioms

We describe in this section the representation of ontology axioms using MPEG-7 constructs. The axioms supported are class generalization and specialization, subsumption, property value restriction specification, equivalence and disjointness of classes, equivalence and separation of individuals, class union and class intersection.

Class Generalization & Specialization. The axioms of class generalization and specialization are represented by MPEG-7 relationships of type “specializes” and “generalizes” respectively.

Let `spec_relationship` be a “Relation” element that represents an MPEG-7 relationship expressing that the subclass abstract semantic entity specializes the class represented by the superclass abstract semantic entity. `spec_relationship` is formally described in regular expression 17:

$$\text{spec_relationship}(sr_type, sr_source, sr_target) \quad (17)$$

where:

- `sr_type` is the type of `spec_relationship` and has “specializes” as value.
- `sr_source` is the source of `spec_relationship` and has as value the identity of subclass.
- `sr_target` is the target of `spec_relationship` and has as value the identity of superclass.

Let `gen_relationship` be a “Relation” element that represents an MPEG-7 relationship expressing that the superclass abstract semantic entity generalizes the class represented by the subclass abstract semantic entity. `gen_relationship` is formally described in regular expression 18:

$$\text{gen_relationship}(gr_type, gr_source, gr_target) \quad (18)$$

where:

- `gr_type` is the type of `gen_relationship` and has “generalizes” as value.
- `gr_source` is the source of `gen_relationship` and has as value the identity of superclass.
- `gr_target` is the target of `gen_relationship` and has as value the identity of subclass.

Consider as an example the “specializes”/“generalizes” pair of relationships between the “SoccerPlayer” and the “Goalkeeper” classes, shown in Figure 10.

Subsumption. The subsumption of an individual Ind to a class C is represented by the pair of MPEG-7 relationships `ex_relationship` and `exBy_relationship`, of type “exemplifies” and “exemplifiedBy” respectively.

`exBy_relationship` is represented by a “Relation” element that is formally described in regular expression 19:

$$\text{exBy_relationship}(ebr_type, ebr_source, ebr_target) \quad (19)$$

where:

- `ebr_type` is the type of `exBy_relationship` and has “exemplifiedBy” as value.
- `ebr_source` is the source of `exBy_relationship` and has as value the identity of the abstract semantic entity that represents C.

- `ex_by_target` is the target of `ex_by_relationship` and has the identity of `Ind` as value.

`ex_relationship` is represented by a “Relation” element that is formally described in regular expression 20:

$$ex_relationship(er_type, er_source, er_target) \quad (20)$$

where:

- `er_type` is the type of `ex_relationship` and has “`exemplifies`” as value.
- `er_source` is the source of `ex_relationship` and has as value the identity of `Ind`.
- `er_target` is the target of `ex_relationship` and has as value the identity of the abstract semantic entity that represents `C`.

Consider as an example the “`exemplifies`”/“`exemplifiedBy`” pair of relationships between the “`Ronaldinho`” and the “`SoccerPlayer`” semantic entities, shown in **Figure 12**.

Value Restrictions. The specification of a fixed value for a property `P` is specified in a different way, depending on the representation of `P`.

If `P` is represented by a “Property” element, the value restriction is represented by the fixed “Term” element that is formally described in regular expression 21:

$$fixed(fname, (fdef|fhref)) \quad (21)$$

where:

- `fname` is the value of the “Name” element of `fixed` and has “`fixed`” as value.
- `fdef` is the value of the “Definition” element of `fixed` and defines the fixed value of `P`.
- `fhref` is the value of the “href” element of `fixed` and has as value a reference to the fixed value of `P`.

If `P` is represented by “Relation” elements, the value restriction is represented, for each class `D` that belongs to the domain of `P`, by a pair of MPEG-7 relationships of type “`property`”/“`propertyOf`” that associate `D` with the concrete semantic entity `V`, representing the fixed value.

If `P` is represented by an “AttributeValuePair” element, the value restriction is represented, according to the type of the fixed value `V` by the value of the appropriate element of “AttributeValuePair” and in particular: **(a)** The value of the element “`BooleanValue`” if `V` is a boolean value; **(b)** The value of the element “`IntegerValue`” if `V` is an integer value; **(c)** The value of the element “`FloatValue`” if `V` is a float number value; **(d)** The value of the element “`TextValue`” if `V` is a string value or a value of type “`TextualType`” (or a type that extends it); **(e)** The value of the element “`IntegerMatrixValue`” if `V` is of type “`IntegerMatrixType`”; **(f)** The value of the element “`FloatMatrixValue`” if `V` is of type “`FloatMatrixType`”; **(g)** The value of the element “`TextAnnotationValue`” if `V` is of type “`TextAnnotationType`” (or a type that extends it); **(h)** The value of the element “`ControlledTermUseValue`” if `V` is of type “`ControlledTermUseType`” (or a type that extends it); **(i)** The value of the element “`DescriptorValue`” if `V` is of type “`DType`” (or a type that extends it).

Consider as an example the fixed value of the “`InitialNumOfPlayers`” property, which is represented by the “`IntegerValue`” element shown in **Figure 5**.

Equivalence of Classes. The specification of a class `A` as equivalent to a class `B` is represented by the “Relation” element `eq_relationship` that is formally described in regular expression 22:

$$eq_relationship(eq_type, eq_source, eq_target) \quad (22)$$

where:

- `eq_type` is the type of `eq_relationship` and has “`equivalent`” as value.
- `eq_source` is the source of `eq_relationship` and has the identity of `A` as value.
- `eq_target` is the target of `eq_relationship` and has the identity of `B` as value.

Consider as an example that the class “`Goalie`” of an ontology “`O1`” is equivalent with the class “`Goalkeeper`” of the ontology of **Figure 2**. The MPEG-7 representation of the equivalence of the classes “`Goalie`” and “`Goalkeeper`” is shown in **Figure 13**.

```
<Relation type="equivalent" source="O1#Goalie" target="#Goalkeeper"/>
```

Figure 13: Representation of the Equivalence of the Class “Goalie” of the Ontology “O1” with the Class “Goalkeeper” of the Ontology of Figure 2

Disjointness. The specification of a class A as disjoint with a class B is represented by a “Relation” element `disjoint_relationship` that is formally described in regular expression 23:

$$\text{disjoint_relationship}(d_type, d_source, d_target) \quad (23)$$

where:

- `d_type` is the type of `disjoint_relationship` and has “disjoint” as value.
- `d_source` is the source of `disjoint_relationship` and has the identity of A as value.
- `d_target` is the target of `disjoint_relationship` and has the identity of B as value.

Equivalence of Individuals. The specification of an individual A as equivalent to an individual B is represented by a “Relation” element `equals_relationship` that is formally described in regular expression 24:

$$\text{equals_relationship}(eq_type, eq_source, eq_target) \quad (24)$$

where:

- `eq_type` is the type of `equals_relationship` and has “equals” as value.
- `eq_source` is the source of `equals_relationship` and has the identity of A as value.
- `eq_target` is the target of `equals_relationship` and has the identity of B as value.

Separation of Individuals. The specification of an individual A as separated from an individual B is represented by a “Relation” element `sep_relationship` that is formally described in regular expression 25:

$$\text{sep_relationship}(sep_type, sep_source, sep_target) \quad (25)$$

where:

- `sep_type` is the type of `sep_relationship` and has “separated” as value.
- `sep_source` is the source of `sep_relationship` and has the identity of A as value.
- `sep_target` is the target of `sep_relationship` and has the identity of B as value.

Consider as an example that the individual “Ronaldo” of an ontology “O1” is separated from the individual “Ronaldinho” of the ontology of Figure 2, since the two individuals represent different soccer players. The MPEG-7 representation of the separation of the individuals “Ronaldo” and “Ronaldinho” is shown in Figure 14.

```
<Relation type="separated" source="O1#Ronaldo" target="#Ronaldinho"/>
```

Figure 14: Representation of the Separation of the Individual “Ronaldo” of the Ontology “O1” with the Individual “Ronaldinho” of the Ontology of Figure 2

Class Union. The definition of a class A as the union of the N (N>0) classes A_1, A_2, \dots, A_N is represented by a “Relation” element `union_relationship` of type “union”.

Let S be an MPEG-7 semantic entity S of type “SemanticType” that represents a collection comprised of the abstract semantic entities that represent the classes A_1, A_2, \dots, A_N . The `union_relationship` “Relation” element is formally described in regular expression 26:

$$\text{union_relationship}(u_type, u_source, u_target) \quad (26)$$

where:

- `u_type` is the type of `union_relationship` and has “union” as value.
- `u_source` is the source of `union_relationship` and has the identity of A as value.
- `u_target` is the target of `union_relationship` and has the identity of S as value.

Class Intersection. The definition of a class A as the intersection of the N (N>0) classes A_1, A_2, \dots, A_N is represented by a “Relation” element `in_relationship` of type “intersection”.

Let S be an MPEG-7 semantic entity S of type “SemanticType” that represents a collection comprised of the abstract semantic entities that represent the classes A_1, A_2, \dots, A_N . The `in_relationship` “Relation” element is formally described in regular expression 27:

$$\text{in_relationship}(i_type, i_source, i_target) \quad (27)$$

where:

- `i_type` is the type of `in_relationship` and has “intersection” as value.
- `i_source` is the source of `in_relationship` and has the identity of A as value.
- `i_target` is the target of `in_relationship` and has the identity of S as value.

8. Exploitation of the Domain Knowledge Representation in Multimedia Applications and Services

Once the domain knowledge regarding the multimedia semantics has been captured and systematically represented, it can be exploited by semantic-based applications and services. In this section we present how the domain knowledge that has been represented according to the proposed model can be exploited both for reasoning support (in subsection 8.1) and in the context of the semantic-based multimedia content retrieval (in subsection 8.2) and filtering (in subsection 8.3).

8.1 Reasoning Support

Since the representations and axioms of the formal model clearly map to corresponding representations and axioms of OWL, the subset of the OWL axioms that hold for the domain knowledge representation are clearly specified. This allows the transformation of the domain knowledge in OWL syntax, its integration in MPEG-7 based ontological infrastructures like the one of the DS-MIRF framework [13] [14] [15], and the use of the existing OWL reasoners for semantic processing. In order to support this scenario in the DS-MIRF framework, we have developed an MPEG-7 to OWL mapping model that allows us to automatically transform domain ontologies expressed in MPEG-7 syntax, according to our formal model, into OWL ontologies. In addition, the audiovisual content descriptions that have been defined according to these ontologies can also be transformed into OWL/RDF descriptions and then used and exploited in the Semantic Web environment. In particular, the OWL/RDF descriptions may be enriched through reasoning that will be performed on both the descriptions and the ontologies. Then, the enriched descriptions can be transformed back to the MPEG-7 syntax and be used in the MPEG-7 working environment.

8.2 Semantic-based Multimedia Content Retrieval

The MPEG-7 descriptions that utilize the domain knowledge that is systematically represented according to our formal model allow for the development of advanced semantic-based retrieval capabilities on top of them. In particular, semantic-based queries that cannot be accurately answered if the domain knowledge is not systematically integrated in the MPEG-7 descriptions can now be supported.

Consider, as an example, the query “give me the goals scored by the national team of Greece”. If the domain knowledge is not systematically integrated in the MPEG-7 descriptions, the query results will include, in addition to the goals scored by Greece, other events caused by the national team of Greece that contain the word “goal” in their descriptions (“shot-on-goal”, “near goal” etc.). The false drops are not included in the query results if the domain knowledge has been systematically integrated and the query language allows the accurate specification of the query conditions. This can be achieved using expressive query languages like the MP7QL query language [11] that we have developed. The MP7QL is a powerful query language that has the MPEG-7 as data model and allows for querying every aspect of an MPEG-7 multimedia content description, while it fully supports the exploitation of domain knowledge in semantic-based queries.

8.3 Semantic-based Multimedia Content Filtering

In addition to the support of advanced semantic-based queries, our formal model also allows the support of advanced semantic-based multimedia content filtering. This can be achieved if, instead of the MPEG-7 user preferences that have limited expressive power, user preference descriptions that are isomorphic with the MPEG-7 content descriptions are supported [1] [9] [10] [11]. Thus, a user preference description model that is isomorphic with the MPEG-7 content description model allows the retrieval of audiovisual content that has been described according to the proposed domain knowledge description knowledge and contains “the goals scored by Greece” (instead of the audiovisual content that contains in its description the keywords “goal” and “Greece”, that will also retrieve the goals scored against Greece). Such a user preference model is the MP7QL Filtering And Search Preferences (FASP) model [10] [11] that we have developed, which is compatible with the MP7QL query language. The MP7QL FASP model allows multimedia content filtering based on every aspect of an MPEG-7 multimedia content description as well as the exploitation of domain knowledge in multimedia content filtering. It is also remarkable that the MPEG-7 user preference descriptions are a special case of the MP7QL user preferences.

9. Conclusions

We have presented in this chapter a formal model that allows the systematic representation and exploitation of domain knowledge using MPEG-7 constructs. The formal model for domain knowledge representation using MPEG-7 constructs proposed here presents clearly and unambiguously a way to integrate domain knowledge in MPEG-7 using exclusively MPEG-7 constructs. Therefore, all the descriptions produced are completely within the MPEG-7 standard.

The proposed model describes clearly and formally the axioms that hold (a subset of the semantics of OWL), and therefore it allows reasoning to be performed by distributed applications that utilize these axioms. This allows advanced functionality for multimedia applications to be implemented and exploited in distributed environments.

10. References

- [1] H. Agius and M.C. Angelides, "From MPEG-7 user interaction tools to hanging basket models: bridging the gap," *Multimedia Tools and Applications*, vol 41, no 3, pp 375-406, 2009.
- [2] The DELOS Network of Excellence in Digital Libraries [Online]. Available: <http://www.delos.info>.
- [3] R. García and O. Celma, "Semantic Integration and Retrieval of Multimedia Metadata," presented at the Knowledge Markup and Semantic Annotation Workshop, Semannot'05, November 2005, Galway, Ireland, CEUR Workshop Proceedings, Vol. 185, pp. 69-80, 2006, ISSN 1613-0073.
- [4] J. Hunter, "Adding Multimedia to the Semantic Web: Building an MPEG-7 ontology," in *Proc. 1st Semantic Web Working Symposium (SWWS 2001)*, Stanford University, California, USA, July 30 - August 1, 2001, pp. 261-283.
- [5] S. Hammiche, S. Benbernou and A. Vakali: "A Logic Based Approach for the Multimedia Data Representation and Retrieval," in *Proc. IEEE International Symposium on Multimedia (ISM 2005)*, 12-14 December 2005, Irvine, CA, USA, pp. 241-248
- [6] The OntoWeb project [Online]. Available: <http://www.ontoweb.org/>.
- [7] P. Salembier, "MPEG-7 Multimedia Description Schemes," *IEEE TA on Circuits and Systems for Video Technology*, 11, 6, 2001, 748-759.
- [8] R. Arndt, R. Troncy, S. Staab, L. Hardman, M. Vacura, "COMM: Designing a Well-Founded Multimedia Ontology for the Web," in *Proc. ISWC/ASWC 2007*, pp. 30-43.
- [9] C. Tsinaraki and S. Christodoulakis, "A Multimedia User Preference Model that Supports Semantics and its Application to MPEG 7/21," in *Proc. IEEE Multimedia Modeling 2006 Conference (IEEE MMM 2006)*, January 2006, Beijing, China, pp. 35-42.
- [10] C. Tsinaraki and S. Christodoulakis, "A User Preference Model and a Query Language that allow Semantic Retrieval and Filtering of Multimedia Content," in *Proc. Semantic Media Adaptation and Personalization Workshop (SMAP 2006)*, December 2006, Athens, Greece, pp. 121-128.
- [11] C. Tsinaraki and S. Christodoulakis, "An MPEG-7 Query Language and a User Preference Model that allow Semantic Retrieval and Filtering of Multimedia Content," *ACM Multimedia Systems Journal*, Special Issue on Semantic Multimedia Adaptation and Personalization, 13(2):131-153 (2007).
- [12] C. Tsinaraki, E. Fatourou and S. Christodoulakis, "An Ontology-Driven Framework for the Management of Semantic Metadata Describing Audiovisual Information," in *Proc. 15th International Conference on Advanced Information Systems Engineering (CAISE)*, June 2003, Klagenfurt/Velden, Austria, pp. 340-356.
- [13] C. Tsinaraki, P. Polydoros and S. Christodoulakis, "Interoperability support for Ontology-based Video Retrieval Applications," in *Proc. Conference on Image and Video Retrieval (CIVR) 2004*, July 2004, Dublin, Ireland, pp. 582-591.
- [14] C. Tsinaraki, P. Polydoros and S. Christodoulakis, "Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant Semantic Indexing," in *Proc. 16th International Conference on Advanced Information Systems Engineering (CAISE)*, June 2004, Riga, Latvia, pp. 398-413.
- [15] C. Tsinaraki, P. Polydoros and S. Christodoulakis, "Interoperability support between MPEG-7/21 and OWL in DS-MIRF," *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, Special Issue on the Semantic Web Era, 2007, 19(2): 219-232.
- [16] C. Tsinaraki, P. Polydoros, F. Kazasis and S. Christodoulakis, "Ontology-based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content," *Multimedia Tools and Application Journal (MTAP)*, Special Issue of on Video Segmentation for Semantic Annotation and Transcoding, 26, pp. 299-325, August 2005.
- [17] W3C OWL Working Group, "OWL 2 Web Ontology Language Document Overview," *W3C Recommendation* (27 October 2009) [Online]. Available: <http://www.w3.org/TR/owl2-overview/>.