

# A Formal Model for Domain Knowledge Representation and Reasoning using MPEG-7 Constructs

Chrisa Tsinaraki

TUC/MUSIC

Technical University of Crete Campus

73100 Kounoupidiana, Crete, Greece

Telephone number: +302821037404

chrisa@ced.tuc.gr

Stavros Christodoulakis

TUC/MUSIC

Technical University of Crete Campus

73100 Kounoupidiana, Crete, Greece

Telephone number: +302821037399

stavros@ced.tuc.gr

## ABSTRACT

The use of domain knowledge in semantic audiovisual content descriptions enhances the functionality and effectiveness of the multimedia applications. However, the dominant standard for audiovisual content description today (MPEG-7) does not describe a formal mechanism for the systematic integration of domain knowledge and reasoning capabilities in the MPEG-7 descriptions. The specification of a formal model for domain knowledge representation and reasoning using the MPEG-7 constructs is of paramount importance for exploiting domain knowledge in order to perform semantic processing of the multimedia content. In this paper we present a formal model that allows the systematic representation of domain knowledge using MPEG-7 constructs and its exploitation in reasoning. The formal model that we describe exploits exclusively MPEG-7 constructs, and the descriptions that are structured according to the model are completely within the MPEG-7 standard.

## Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – video. H.2.4 [Systems]: Multimedia Databases.

## General Terms

Design, Standardization, Languages.

## Keywords

MPEG-7, Domain Knowledge, Ontologies, Reasoning.

## 1. INTRODUCTION

The popularity of the audiovisual content and its associated services (including the traditional TV and video services), the advent of the digital media, the availability of low-cost audiovisual content management devices and the development of advanced network infrastructures have led to the development of an open audiovisual content consumption environment. In such an open environment, interoperability supported by standards is necessary for the services provided by different vendors to interoperate. In addition, 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 descriptions of the audiovisual content.

The *MPEG-7* [5], 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 they do not describe the systematic integration and exploitation of domain knowledge in the semantic descriptions. On the other hand, it is well accepted that the utilization of domain knowledge, usually expressed in the form of domain ontologies, can improve the functionality and effectiveness of the multimedia applications. It can, for example, allow for reasoning on top of the multimedia content descriptions and improve the efficiency of the retrieval and filtering of the audiovisual content [1][4], since it allows the classification of semantic entities that are represented by the same MPEG-7 construct. This way, queries about specific classes of semantic entities in a domain of knowledge can be accurately expressed and efficiently answered.

It is 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 and reasoning using the MPEG-7 constructs is of paramount importance for clearly understanding, communicating, specifying, implementing and exploiting domain knowledge, in the Web application environment of independent organizations, in order to perform semantic processing of the multimedia 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 paper we present a formal model for domain knowledge representation and reasoning 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 formal model for domain knowledge representation and reasoning using MPEG-7 constructs which is presented in this paper achieves the following objectives:

- It presents clearly and unambiguously a way to integrate domain knowledge in MPEG-7 using only 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 formal model representations and axioms clearly map to corresponding representations and axioms of OWL. The subset of the OWL axioms that hold for the domain knowledge representation and reasoning are clearly specified. This allows the transformation of the domain knowledge in OWL syntax, their integration in MPEG-7 based ontological infrastructures like the one of the DS-MIRF framework [6], and the use of the existing OWL reasoners for semantic processing.

The model for domain knowledge representation and reasoning 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 [2], which essentially allowed only the representation of taxonomies using MPEG-7 syntax.

The rest of the paper is organized as follows: The semantic description constructs of MPEG-7 are described in section 2, the proposed MPEG-7 based domain knowledge representation model is presented in section 3 and the paper concludes in section 4.

## 2. THE SEMANTIC DESCRIPTION CONSTRUCTS OF MPEG-7

We describe here the semantic description constructs of the MPEG-7 standard. The MPEG-7 standard is expressed using XML Schema syntax, provides a set of general-purpose types that allow for the description of the audiovisual content semantics. These types are rooted at the *SemanticBaseType* of MPEG-7 which represents semantic entities. The subtypes of *SemanticBaseType* that represent specific types of semantic entities are listed below:

- The *SemanticType* is a concrete type that represents semantic entity collections.
- The *AgentObjectType* is a concrete type that represents the actors appearing in an audiovisual segment. The actors are specified in the *Agent* element of *AgentObjectType* and are represented by the subtypes of the abstract type *AgentType*. *PersonType*, *OrganizationType* and *PersonGroupType* are the subtypes of *AgentType* and represent, respectively, persons (e.g. soccer players), organizations (e.g. soccer teams) and groups of persons.
- The *ObjectType* is a concrete type that represents objects and object abstractions (e.g. a ball).
- The *EventType* is a concrete type that represents events (e.g. goal, penalty etc.).
- The *ConceptType* is a concrete type that represents concepts (e.g. co-operation).
- The *SemanticStateType* is a concrete type that represents states and allows the parametric description of their features (e.g. the score of a soccer game).

- The *SemanticPlaceType* is a concrete type that represents places (e.g. a soccer field).
- The *SemanticTimeType* is a concrete type that represents semantic time (e.g. the first half-time of a soccer game).

The MPEG-7 semantic entities may be either abstract or concrete, depending of the value of their *AbstractionLevel* element. *AbstractionLevel* has the *dimension* attribute, of non-negative integer type. When *AbstractionLevel* is not present in a semantic description, the description refers to specific audiovisual material. When *AbstractionLevel.dimension=0*, it is a description of a reusable semantic entity that is referenced from every segment where the semantic entity appears. When *AbstractionLevel* has a non-zero *dimension*, it specifies classes for the description of abstract semantic entities.

The MPEG-7 also specifies more than 100 types of relationships that allow associating semantic entities (e.g. before, over, part of etc.). The MPEG-7 relationships are represented by *Relation* elements that may be placed independently in the MPEG-7 descriptions or inside semantic entities.

The major limitation of the semantic description constructs of the MPEG-7 standard is that they are general-purpose and they do not allow for the systematic representation of domain knowledge. 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 [3]. 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.

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

## 3. MPEG-7 BASED DOMAIN KNOWLEDGE REPRESENTATION

In this section we describe our formal model for domain knowledge representation and reasoning 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.

The domain knowledge representation model that utilizes the MPEG-7 constructs is detailed in the following paragraphs. The domain ontology representation is described in subsection 3.1, the representation of properties is described in subsection 3.2, the representation of classes is described in subsection 3.3, the

representation of individuals is described in subsection 3.4 and the representation of axioms is described in subsection 3.5. Finally, an overview of the proposed model for domain knowledge representation using MPEG-7 constructs is presented in subsection 3.6.

### 3.1 Domain Ontology Representation

In this subsection we describe the representation of domain ontologies. 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):

$DO(do\_name, label, comment, imported\_ontologies,$  (1)  
 $classes, properties, individuals, relationships)$

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 constraints (details on axioms are provided in subsection 3.5).

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:

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

where:

- $id$  is the identity of  $S$  and 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 and 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  $DO$ .
- $p$  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 1.

```
<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 1: 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:

$OD(name, def, uri)$  (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:

$uri(tname, tdef)$  (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 1 is represented by the “Property” element, which has as value of its “Name” element “Ontology Self”.

### 3.2 Property Representation

The representation of the properties defined in a domain ontology *DO* using MPEG-7 constructs is detailed in this subsection. 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 by the regular expression 5:

$$P(p\_id, range, domain, value, label, comment) \quad (5)$$

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:

$$prop(name, type, fixed) \quad (6)$$

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 1, a “DateOfBirth” property, of type “Date”, has been defined for the soccer players. The MPEG-7 representation of the property is shown in Figure 2.

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

**Figure 2: 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.

$$pr\_relationship(pr\_type, pr\_source, pr\_target, pr\_name) \quad (7)$$

$$prOf\_relationship(por\_type, por\_source, por\_target, por\_name) \quad (8)$$

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 1, 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 (which are represented by the “City” class). The MPEG-7 representation of the property is shown in Figure 3.

```
<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 3: 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 by the regular expression 9:

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

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 this is expressed by the “InitialNumOfPlayers” property 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 4.

```

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

```

**Figure 4: Representation of the “InitialNumOfPlayers” Property**

### 3.2.1 Property Value Representation

The property values of the individuals that are defined in a domain ontology *DO* are represented, according to 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 by the 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 “Buffon” individual, which represents the goalkeeper Buffon, exists in the soccer ontology of Figure 1 and that the value of its “DateOfBirth” property is “21/01/1978”. The MPEG-7 representation of the property value is shown in Figure 5.

```

<Property>
  <Name>DateOfBirth</Name>
  <Definition>21/01/1978</Definition>
</Property>

```

**Figure 5: 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.
- *por\_name* is the name of *P*.

Consider as an example that the “Buffon” individual has the “Carrara” as value of its “PlaceOfBirth” property. The MPEG-7 representation of the property value is shown in Figure 6.

```

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

```

**Figure 6: 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 a type that extends it).

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 1 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 6.

```

<AttributeValuePair>
  <Attribute>
    <Name>Status</Name>
  </Attribute>
  <TextValue>TextAnnotationType</TextValue>
  <TextAnnotationValue id="professional">
    <FreeTextAnnotation id="professionalFTA">
      Professional
    </FreeTextAnnotation>
  </TextAnnotationValue>
</AttributeValuePair>

```

**Figure 7: Representation of the Value of the “Status” Property of a professional soccer team**

### 3.3 Class Representation

We present in this subsection the MPEG-7 representation of the classes defined in domain ontologies, which are represented 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 1 and that “Goalkeeper” is a subclass of “SoccerPlayer”. The MPEG-7 representation of these classes is shown in Figure 8.

```

<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 8: Representation of the “SoccerPlayer” and “Goalkeeper” Classes**

Let  $C$  be a class defined in a domain ontology  $DO$  that is described by the 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”. 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 by the 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 8 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 8 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 8 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.
- spec\_relationship is an MPEG-7 relationship of type “specializes”, which is defined only if superclass  $\neq$  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.

### 3.4 Representation of Individuals

We describe in this subsection the representation of individuals defined in domain ontologies using MPEG-7 constructs. We represent the individuals as concrete semantic entities. Consider as an example the “Buffon” individual, which represents the goalkeeper Buffon and is shown in Figure 9.

```
<SemanticBase xsi:type="AgentObjectType"
id="Buffon">
  <AbstractionLevel dimension="0"/>
  <Label>
    <Name>Buffon</Name>
  </Label>
  <Definition>
    <FreeTextAnnotation>
      Gianluigi Buffon
    </FreeTextAnnotation>
  </Definition>
  <Property>
    <Name>DateOfBirth</Name>
    <Definition>21/01/1978</Definition>
  </Property>
  <Relation type="exemplifies"
source="#Buffon" target="#Goalkeeper"/>
  <Relation type="exemplifiedBy"
source="#Goalkeeper" target="#Buffon"/>
  <Agent xsi:type="PersonType"
id="BuffonPerson">
    <Name>
      <FamilyName initial="B">
        Buffon
      </FamilyName>
      <GivenName>Gianluigi</GivenName>
    </Name>
  </Agent>
</SemanticBase>
```

**Figure 9: Representation of the Individual “Buffon”**

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 by the regular expression 15:

$Ind(ind\_id, class, MPEG7\_type, properties, relationships, label, comment)$  (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 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  $\neq$  MPEG7\_type.
- relationships is the set of the relationships of Ind.

### 3.5 Representation of Axioms

We describe in this subsection the representation of ontology axioms using MPEG-7 constructs. The axioms supported are class generalization and specialization, subsumption, property value constraint 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 by the regular expression 17:

$$spec\_relationship(sr\_type, sr\_source, sr\_target) \quad (17)$$

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 by the regular expression 18:

$$gen\_relationship(gr\_type, gr\_source, gr\_target) \quad (18)$$

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 8.

**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:

$$exBy\_relationship(ebr\_type, ebr\_source, ebr\_target) \quad (19)$$

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.
- ebr\_target is the target of exBy\_relationship and has the identity of Ind as value.

ex\_relationship is represented by a “Relation” element that is formally described by the 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 “Buffon” and the “Goalkeeper” semantic entities, shown in Figure 9.



**Value Constraints.** 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 constraint is represented by the fixed “Term” element that is formally described in regular expression 21:

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

where:

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

If  $P$  is represented by “Relation” elements, the value constraint 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 constraint 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 4.

**Equivalence of Classes.** The specification of a class  $A$  as equivalent to a class  $B$  is represented by the “Relation” element  $\text{eq\_relationship}$  that is formally described in regular expression 22:

$$\text{eq\_relationship}(\text{eq\_type}, \text{eq\_source}, \text{eq\_target}) \quad (22)$$

where:

- $\text{eq\_type}$  is the type of  $\text{eq\_relationship}$  and has “equivalent” as value.
- $\text{eq\_source}$  is the source of  $\text{eq\_relationship}$  and has the identity of  $A$  as value.
- $\text{eq\_target}$  is the target of  $\text{eq\_relationship}$  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 1. The MPEG-7 representation of the equivalence of the classes “Goalie” and “Goalkeeper” is shown in Figure 10.

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

**Figure 10: Representation of the Equivalence of the Class “Goalie” of the Ontology “O1” with the Class “Goalkeeper” of the Ontology of Figure 1**

**Disjointness.** The specification of a class  $A$  as disjoint with a class  $B$  is represented by a “Relation” element  $\text{disjoint\_relationship}$  that is formally described in regular expression 23:

$$\text{disjoint\_relationship}(\text{d\_type}, \text{d\_source}, \text{d\_target}) \quad (23)$$

where:

- $\text{d\_type}$  is the type of  $\text{disjoint\_relationship}$  and has “disjoint” as value.
- $\text{d\_source}$  is the source of  $\text{disjoint\_relationship}$  and has the identity of  $A$  as value.
- $\text{d\_target}$  is the target of  $\text{disjoint\_relationship}$  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  $\text{equals\_relationship}$  that is formally described in regular expression 24:

$$\text{equals\_relationship}(\text{eq\_type}, \text{eq\_source}, \text{eq\_target}) \quad (24)$$

where:

- $\text{eq\_type}$  is the type of  $\text{equals\_relationship}$  and has “equals” as value.
- $\text{eq\_source}$  is the source of  $\text{equals\_relationship}$  and has the identity of  $A$  as value.
- $\text{eq\_target}$  is the target of  $\text{equals\_relationship}$  and has the identity of  $B$  as value.

**Separation of Individuals.** The specification of an individual  $A$  as different from an individual  $B$  is represented by a “Relation” element  $\text{nequals\_relationship}$  that is formally described in regular expression 25:

$$\text{nequals\_relationship}(\text{eq\_type}, \text{eq\_source}, \text{eq\_target}) \quad (25)$$

where:

- $\text{eq\_type}$  is the type of  $\text{nequals\_relationship}$  and has “separated” as value.
- $\text{eq\_source}$  is the source of  $\text{nequals\_relationship}$  and has the identity of  $A$  as value.
- $\text{eq\_target}$  is the target of  $\text{nequals\_relationship}$  and has the identity of  $B$  as value.

Consider as an example that the individual “Valdes” of an ontology “O1” is not equal to the individual “Buffon” of the ontology of Figure 1, since they represent different goalkeepers. The MPEG-7 representation of the separation of the individuals “Valdes” and “Buffon” is shown in Figure 11.

```
<Relation type="separated" source="O1#Valdes"
target="#Buffon"/>
```

**Figure 11: Representation of the Separation of the Individual “Goalie” of the Ontology “O1” with the Individual “Buffon” of the Ontology of Figure 1**

**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 by the 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.

### 3.6 Overview of the Domain Knowledge Representation and Reasoning Model

An overview of the proposed model for domain knowledge representation using MPEG-7 constructs is presented in this subsection. As is shown in Figure 12, the model is based on the MPEG-7 relationships and on the capability of defining both abstract and 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.

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 Constraint	“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, our model for domain knowledge representation and reasoning using MPEG-7 constructs covers a subset of the semantics of OWL. In particular, the proposed model cannot represent, due to the limitations of the MPEG-7 syntax the following axioms: Class complement, inverse properties, cardinality constraints, property equivalence, property specialization and separation of properties. 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  $ODO$ , thus allowing the semantic processing of the semantics of  $MP7DO$ .

The proposed model for domain knowledge representation and reasoning using MPEG-7 constructs is a formal logic-based extension of the informal model we have developed in our previous research [2], which essentially allowed the representation of taxonomies using MPEG-7 syntax and was implemented on top of relational and XML databases [2] [7].

## 4. Conclusions

We have presented in this paper a formal model that allows the systematic representation and exploitation of domain knowledge using MPEG-7 constructs.

The formal model for domain knowledge representation and reasoning using MPEG-7 constructs proposed in this paper presents clearly and unambiguously a way to integrate domain knowledge in MPEG-7 using only MPEG-7 constructs. Therefore, all the descriptions produced are completely within the MPEG-7 standard.

In addition, the proposed model 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 for multimedia applications to be implemented and exploited in distributed environments.

Finally, the formal model representations and axioms clearly map to corresponding representations and axioms of OWL. The subset of the OWL axioms that hold for the domain knowledge representation and reasoning are specified. This allows the transformation of the domain knowledge in OWL syntax, their integration in MPEG-7 based ontological infrastructures like the one of the DS-MIRF framework [6], and the use of the existing OWL reasoners for semantic processing.

## 5. REFERENCES

[1] The DELOS Network of Excellence in Digital Libraries. (<http://www.delos.info>).

[2] Tsinarakis C., Fatourou E., Christodoulakis S.: "An Ontology-Driven Framework for the Management of Semantic Metadata Describing Audiovisual Information". In the proceedings of the 15<sup>th</sup> International Conference on Advanced Information Systems Engineering (CAISE), pp. 340-356, June 2003, Klagenfurt/Velden, Austria

[3] Hammiche S., Benbernou S., Vakali A.: "A Logic Based Approach for the Multimedia Data Representation and Retrieval", ISM 2005: 241-248

[4] The OntoWeb project. (<http://www.ontoweb.org/>).

[5] Salembier P.: "MPEG-7 Multimedia Description Schemes". IEEE TA on Circuits and Systems for Video Technology, 11, 6, 2001, 748-759.

[6] Tsinarakis C., Polydoros P., Christodoulakis S.: "Interoperability support between MPEG-7/21 and OWL in DS-MIRF". In IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), Special Issue on the Semantic Web Era, 2007, 19(2): 219-232

[7] Tsinarakis C., Polydoros P., Syntzanaki A., Martin A., Perego A., Christodoulakis S.: "Semantic, Constraint & Preference Based Authoring of Multi-topic Multimedia Presentations". In Journal of Digital Information Management (JDIM), Volume 4, Issue 4, pp. 207-213, December 2006.

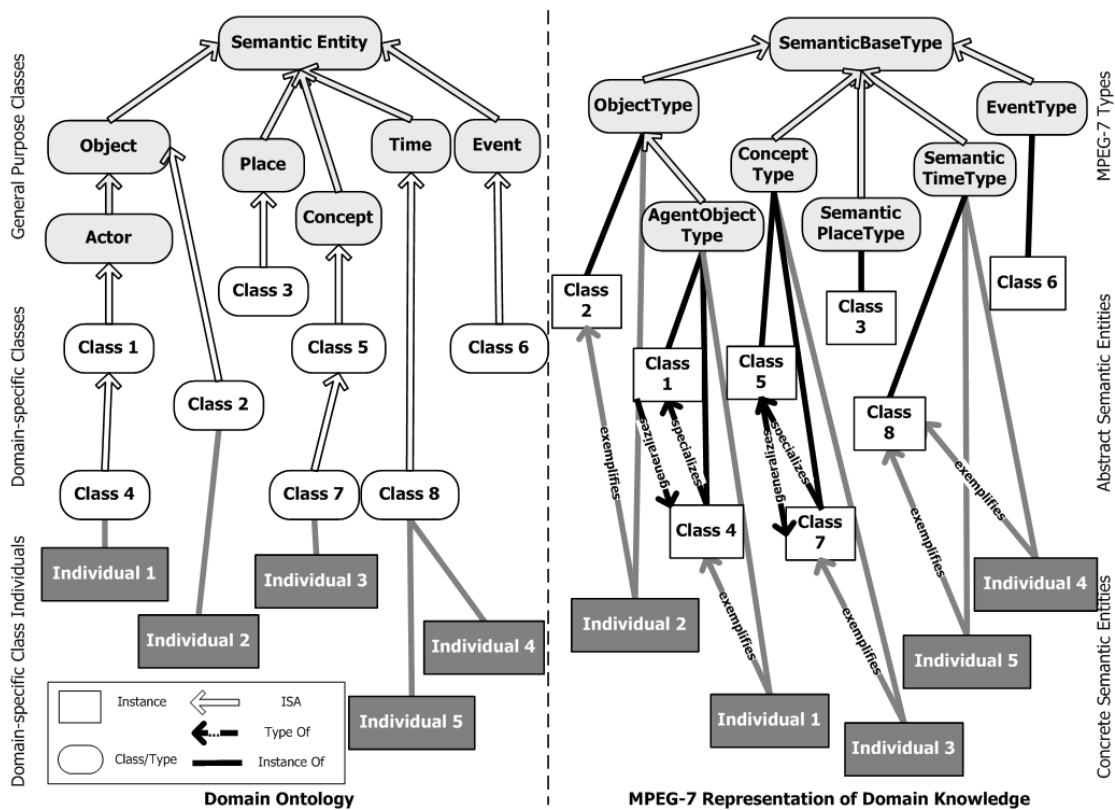


Figure 12: The Domain Knowledge Representation Model