

# An MPEG-7 framework enhancing the reuse of 3D models

Ioan Marius Bilasco, Jérôme Gensel, Marlène Villanova-Oliver, Hervé Martin  
Laboratoire LSR IMAG  
681 rue de la Passerelle, BP. 72  
38402, Saint Martin d'Hères Cedex, France  
(+33)476827280  
{bilasco, gensel, villanov, martin}@imag.fr

## ABSTRACT

The continuous evolution of computer capacities, as well as the emergence of the X3D standard has recently boosted the 3D domain. Even if efficient tools that support the designer's work exist, little attention is paid to the reuse of 3D models. Associating some semantics with 3D contents is an important issue for reusing such contents or pieces of content. In this paper, we address this issue by using a generic semantic annotation model for 3D, called 3DSEAM [Bilasco et al. 2005b] (3D SEmantics Annotation Model). 3DSEAM aims at indexing 3D contents considering visual, geometric and semantic aspects. A generic 3D Annotation Framework (called 3DAF) is proposed in order to manage the semantic annotations of 3D objects. 3DAF is instantiated using an MPEG-7-based architecture. An extension of MPEG-7 that addresses 3D content is used. [Bilasco et al. 2005a].

**CR Categories:** H.3.1. [Information Storage and Retrieval]: Content Analysis and indexing – *indexing methods*

**Keywords:** reusing 3D contents, semantic annotations, annotation framework, MPEG-7

## 1 Introduction

Nowadays, the evolution of computers capabilities facilitates the deployment of 3D contents. With the emergence of widely accepted standards such as X3D [Web3D 2001], 3D contents progressively cover more and more application domains: from spatial planning, transports, construction, defence to architecture and tourism, to name a few. What makes the success of 3D is probably the possibility of creating, modifying or running some simulations on 3D virtual representations of real world environments which are more realistic and more attractive than linear or 2D data.

Efforts have been made in order to facilitate the work of 3D model designers, providing them with powerful (commercial) tools (such as *Maya* by *Alias|wavefront*<sup>1</sup> or *3D Studio Max* by

*Discreet*<sup>2</sup>) for creating complex 3D objects and scenes. Although the construction of a scene consumes a lot of time and resources, 3D contents are usually designed for specific applications only.

The ability of reusing 3D scenes in various application contexts is very important for the multimedia community as it would reduce the development costs of 3D applications. We intend to offer a solution to the reusability problem of 3D models. Our work consists in providing an extensible framework whose final objective is to support the reuse of the main 3D model representation formats (3DS [Autodesk 1997], LWO [Hastings and Ferguson 1994], DXF [Autodesk 2006], COB [Caligari 2002], X3D [Web3D 2001]...).

By reuse, we mean both the reuse of parts of a model independently from a specific application (i.e. the same 3D model describing a tree could be reused to construct different 3D scenes where trees appear) and the reuse of the same model within various applications (i.e. specific information related to different applications can be attached to the same 3D model describing a tree: the wind resistance of the tree for a risk-prevention application or the specie of the tree for a botanical-oriented application).

Basic means of reusing parts of models are available. If parts of models are saved in separate files, then they can be included afterwards in some new models. Thus, a very simple research criterion, the exact address (the local address or the address on some server) of that file is used for the retrieval of reusable 3D model parts. Some interesting propositions [Del Bimbo and Pala 2004; Kim et al. 2004; Ohbuci et al. 2003] based on some alternate research criterion are available. In these propositions, models are automatically indexed using signal analysis. Then query by example techniques are applied in order to retrieve 3D contents. However, these solutions rely on the indexation of the geometry or the appearance of 3D models. Semantic retrieval queries are not available in such approaches despite the fact that, some high level research criteria (semantics, content ...) would increase the efficiency and the accuracy of the reuse techniques as the needs are expressed in a more accurate manner.

For instance, reuse techniques are of major interest in modelling and simulation applications. These kinds of applications usually deal with large and complex information spaces exploited in various models and simulations. The Defense Advanced Research Projects Agency (DARPA) initiated in 1994 a project called

<sup>1</sup> <http://www.alias.com>

Copyright © 2006 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail [permissions@acm.org](mailto:permissions@acm.org).

Web3D 2006, Columbia, Maryland, 18–21 April 2006.  
© 2006 ACM 1-59593-336-0/06/0004 \$5.00

<sup>2</sup> <http://www.autodesk.com>

SEDRIS<sup>3</sup> that aims at providing solutions to the complex problem of environmental data representation and interchange for networked heterogeneous applications. The project provides a complete (terrain, ocean, atmosphere, and space) data model of the physical environment [Foley et al. 1998]. A 3D-related application issued from the SEDRIS project is presented in [Schaefer et al. 2002] and deals with the reuse and the interchanging of environmental 3D data. A web-based repository of Synthetic Natural Environment (SNE) objects (3D model of buildings, bridges, tanks, aircrafts, etc.) built on top of SEDRIS data model supports user-friendly methods for searching and accessing objects. Keyword-based queries are implemented and accessible through web forms. However, complex queries that take into account spatial relations (e.g. retrieve the tank closest to the bridge A) are not yet available.

To cope with complex semantic queries we have proposed 3DSEAM, a semantic annotation model for 3D objects [Bilasco et al. 2005b]. 3DSEAM aims at locating, indexing and annotating objects contained in 3D scenes. Objects are defined by their (geometric or structural) location in the 3D scene. Each object is associated with a (real-world) entity. Different classes of semantic properties characterizing the entity are stored in some semantic profiles. The annotations contained in these semantic profiles enrich the basic geometric description of 3D models and enhance the execution of complex actions (i.e. reuse, adaptation ...) on 3D scenes.

The attachment of semantic information to multimedia objects is not a recent topic in the multimedia community. Important research efforts have been made in the characterisation (in terms of low-level features or high-level semantics) of audios, images or videos. For instance, MPEG-7 [Martinez and Koenen 2002] descriptors and description schemas are largely accepted as multimedia description standard tools. Due to its great flexibility and extensibility, we have decided to place the MPEG-7 at the very centre of our reusing framework. However, in its current state, MPEG-7 does not cover 3D objects. In order to cope with this problem, we have introduced some specific descriptors and description schemas for 3D data [Bilasco et al. 2005a]. Standard MPEG-7 Semantic Base description schemas are used in order to encode the semantic profiles defined in 3DSEAM.

The paper is organized as follows. The next section gives an insight of the 3D world representation, focusing on the X3D standard. 3DSEAM [Bilasco et al. 2005b] (3D SEmantics Annotation Model), a generic annotation model covering 3D content management concerns, is briefly presented in Section 3. Section 4 introduces a 3DSEAM-powered semantic annotation framework for 3D objects (called 3DAF). Section 5 presents an MPEG-7 instantiation of 3DAF and a 3DAF extension that enhances the reuse of 3D objects.

## 2 The design of 3D scenes using X3D

The Extensible 3D (X3D) standard, proposed by the Web3D consortium [Web3D 2001], defines a runtime environment and a delivery mechanism for 3D contents and applications running over a network. It combines geometry descriptions, runtime

behavioural descriptions and control features. It proposes different kinds of encodings, including an XML encoding.

An X3D document represents a 3D scene as a direct acyclic graph. The graph contains primitives (Cube, Box, IndexLineSet, etc) and geometric transformations (Transform) which apply translations and rotations to primitives. Some non-geometric elements are included: composite objects (Group), environmental elements (lights, viewpoints...), reuse primitives (DEF/USE, PROTOTYPE), metadata (WorldInfo) etc.

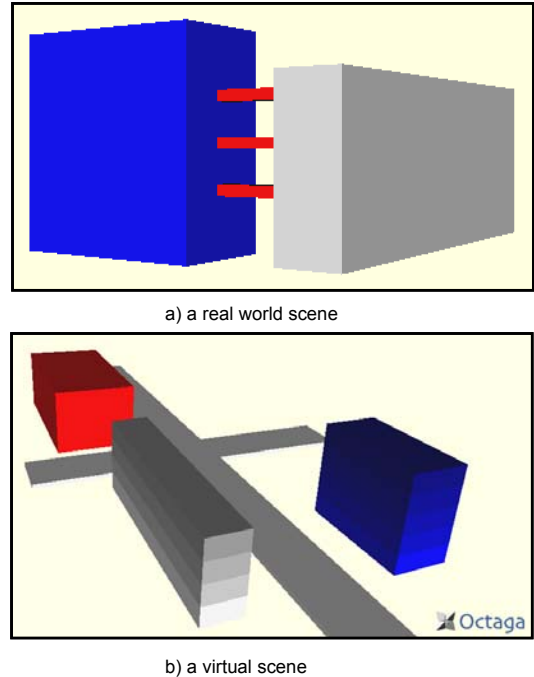


Figure 1: The White and the Dark buildings modelled using X3D.

In order to illustrate the use of X3D, we show in Figure 1 two scenes presenting the building hosting our laboratory. The first scene (Figure 1a.) is extracted from a 3D scene modelling some building located on the Campus of the University. The (White and the Dark) buildings are positioned following the real-world spatial organisation. This scene gives a coarse representation of buildings, which are modelled by simple 3D boxes. The scene also contains three aerial passages modelled by 3 boxes. The second scene (Figure 1b.) presents a virtual scene in which the two buildings (the White and the Dark ones) are not placed following their real-world positions. The representation is more detailed since floors are modelled. Consequently, a building is represented as a group of boxes. Furthermore, we have introduced some variations of the main colour for each building in order to better distinguish between the floors.

The main goal of an X3D scene model is to give a precise and rich representation in terms of geometric and environmental features (lights, ambient sounds, etc.). But, the semantic aspects are scarcely addressed. This fact should be acceptable since the type of the semantic information associated with the 3D objects contained in a particular scene varies from one application domain to another. However, the X3D language provides some basic tools that associate some textual information with the objects of a 3D scene by means of `WorldInfo` nodes that can be

<sup>3</sup> <http://www.sedris.com>

attached to any element in the document. Researches that exploit the WorldInfo node for adding some semantics have been conducted. For instance, in [Hetherington and Scott 2004] WorldInfo nodes are used like time stamps attached to X3D nodes in order to add time into 3D scenes. But still, we consider this solution limited to specific application in which the nature (structure) and the use of the metadata are known in advance. If the scenes in which timestamps have been introduced are now to be used in another application, the (X3D) file might be modified in order to substitute/enrich/delete previous (possibly now obsolete) metadata. To cope with the variability of metadata throughout different applications, the WorldInfo node should point at an external file that contains metadata related to the object. Then, issues related to the adoption of a common standard for metadata representation and querying should be addressed.

Although the XML encoding of an X3D file facilitates the retrieval of attributory information (position, appearance ...), complex queries still cannot be answered (for instance, queries like: *retrieve the floor hosting the students' office in the scene 1b*). A complementary description, by means of annotations, enhances the use of query criteria on the basis of semantic aspects. First, annotations localize the geometric elements that correspond to real world objects. Second, semantic should be associated with the related object (the White building hosts the LSR laboratory).

### 3 Towards a generic model for 3D content management

The 3D object localization process is a prerequisite step for the annotation process. The use of structural and/or spatial locators [Bilasco et al. 2005a] of objects eligible for reuse is a basic form of indexation. However, more information is to be added in order to better characterize 3D objects. In [Bilasco et al. 2005b] we have presented a generic model for annotating 3D contents with semantics: 3DSEAM (3D SEmantic Annotation Model). The model deals with the characterisation of 3D objects seen in the same time as: geometric entities (shape, dimension), semantic entities (real-world attributes and its relations with other objects) and multimedia fragments (type of serialisation, hardware

requirements and software constraints).

Figure 2, partially depicts the 3DSEAM model using UML [Booch et al. 1995]. The central part of the model is the concept of *Entity*, which defines any real-world object. *Entities* are characterized by the semantic information associated with them (class *Semantics*), as well as by the way they are materialized within the scene document (class *MultimediaFragment*).

Since the needs in terms of semantic information can vary the semantic information is organized into *semantic profiles* (which can be seen as specific domain ontologies [Gruber 1993]). A *semantic profile* gathers properties and relations related to a specific application domain.

Materializations (*MultimediaFragment* class) of the same entity can appear in different documents. Each fragment could vary in terms of the nature of the medium used to represent this entity (text, image, video, 3D object), the encoding (primitives-based or polygons-based 3D scenes, etc), the size, the complexity, etc. These data together with some management information (creator, copyright information ...) define the *MediaProfile* of the fragment. The *MediaLocator* collects notations used for locating fragments inside a multimedia document. Media fragments are identified using Temporal, spatial or structural locators. A fourth locator *CompositeLocator* combines the three basic ones leading to the management of complex locators (i.e. spatio-temporal locators, spatial-structural locators).

The first level of our model (on top of the Figure 2) is composed of the following classes: *Semantics*, *Entity*, *MultimediaFragments*, *MediaProfile*, *Media Locator* and its extensions. This level can be seen as a generic multimedia annotation model. No particular constraint on the nature of the annotated content (3D or other) is imposed. We consider this choice as essential since by extending the basic classes, the integration of annotation related to audio, 2D and 3D contents is possible in the same model. We adopt this approach, because in 3D scenes, 3D objects can be decorated using multimedia textures (images or even videos). Besides, sounds can be attached to

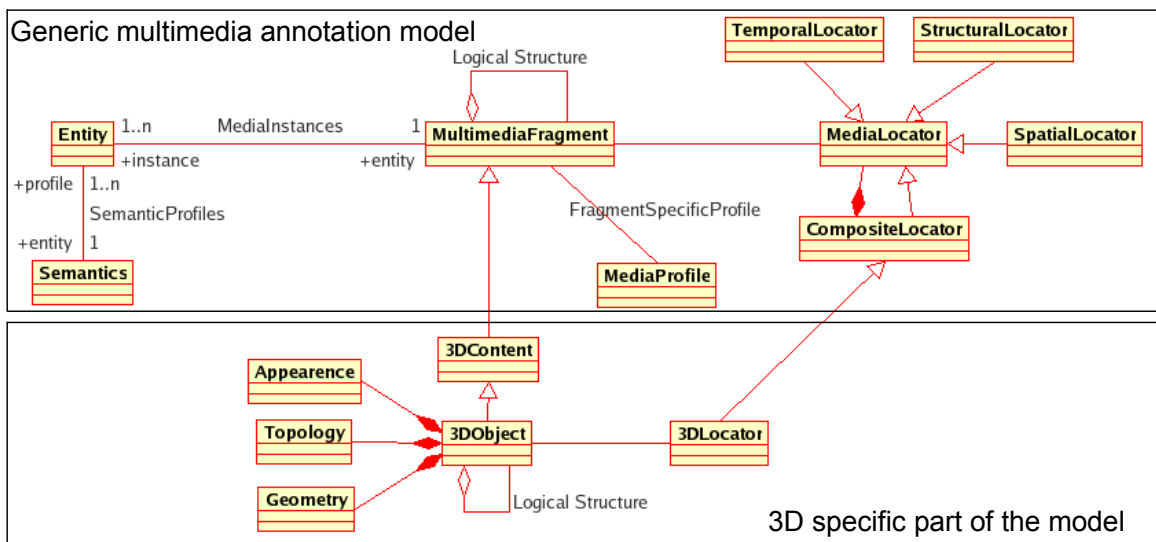


Figure 2: Partial view of the 3DSEAM semantic annotation model

specific location inside a 3D scene. Hence, a complete and coherent annotation of a 3D scene must also take into account – besides the 3D objects – multimedia elements such as images, video or audio clips contained into the scene. Though we do not insist here on this aspect, the integration of audio or 2D content can be implemented into the 3DSEAM model by extending the `MultimediaFragment` class. We focus here on the extension concerning the 3D content which is presented in the lower part of Figure 2.

A `3DObject` refers to a fragment (a *sub-scene*) of a 3D scene. A fragment includes the geometry (`Geometry` class), the appearance (`Appearance` class), and any other environmental aspect (lights...) related to it. Besides, the geometry and the appearance, the topology of 3D objects – which is not directly coded but implicit in the scene representation – is an important feature in the majority of 3D related projects [Pfund 2001; Zlatanova et al. 2004] as it facilitates reasoning on the relative position of these objects inside a 3D model. The environmental aspects as well as all the behavioural aspects (some presented in [Dinesh et al. 2001]) are not yet included in our model. However, we can think of some extensions of the `3DObject` class like an `Animated3DObject` class which would group information describing the behaviour.

The logical structure of a 3D object is stored as it provides some interesting information concerning the organisation of the content inside a 3D model or scene. Queries based on composition criteria (for instance, *find all the buildings having four floors*) can be easily answered by simply analyzing the logical structure. The result can be constructed by simply analysing the logical structure of existing scenes and retrieving those which match the query (here, building objects that have a logical structure that includes several objects, four of them being floor objects).

Figure 3 shows a partial view (the logical structure of multimedia fragments) of the indexation result obtained by using the 3DSEAM model (presented in Figure 2) for the real-world scene (presented in Figure 1a). Dashed lines attach properties (here, the location) to the `3DObject` instances. Solid lines correspond to composition relations existing between `3DObject` instances

(defining the `LogicalStructure`) and `MediaLocator` instances. The composite element is drawn above its components.

The whole scene is seen as a `3DObject` (`RealScene`). A structural locator – pointing at the root element (`X3D`) of the scene – is used in order to associate the `RealScene` with the whole scene. The `RealScene` contains three main objects: the White building, the Dark building and the Passages. A third decomposition level is considered for the White building (White first floor ...) and the Passages (Level1 to Level3). The Level1 object is localized using a `StructuralLocator`. A `CompositeLocator` is used to point at the first floor (White first floor). In a similar way, `CompositeLocator` are used to point at the other floors. Every multimedia fragment identified can now be associated with a real world entity. Semantic information attached to the real-world entity is coupled with the corresponding scene object.

#### 4 A framework for reusable 3D objects

We propose a generic framework, called 3DAF, whose goal is to ease the use of semantic information attached to 3D objects. This framework enhances the annotating process and manages the existing annotations. 3DAF introduces an intermediate layer between the agent's (user or application) needs in terms of knowledge and the actual representation of the knowledge (MPEG-7, RDF, OWL, DBMS ...) using the 3DSEAM model.

3DAF (see Figure 4) is composed of three main components: the *Annotations Repository*, the *Annotation Manager*, and the *Query Manager*.

The *Annotations Repository* controls the storage of 3DSEAM model instances. The repository is composed of three components. Each component treats independently one dimension: the semantics, the entities and the multimedia fragments. Transversal links are used in order to attach semantics and multimedia fragments with the corresponding entity. There is no direct link between the multimedia fragment and the semantics. Semantics associated with multimedia fragments are accessible in two steps: 1) get the entity associated to the

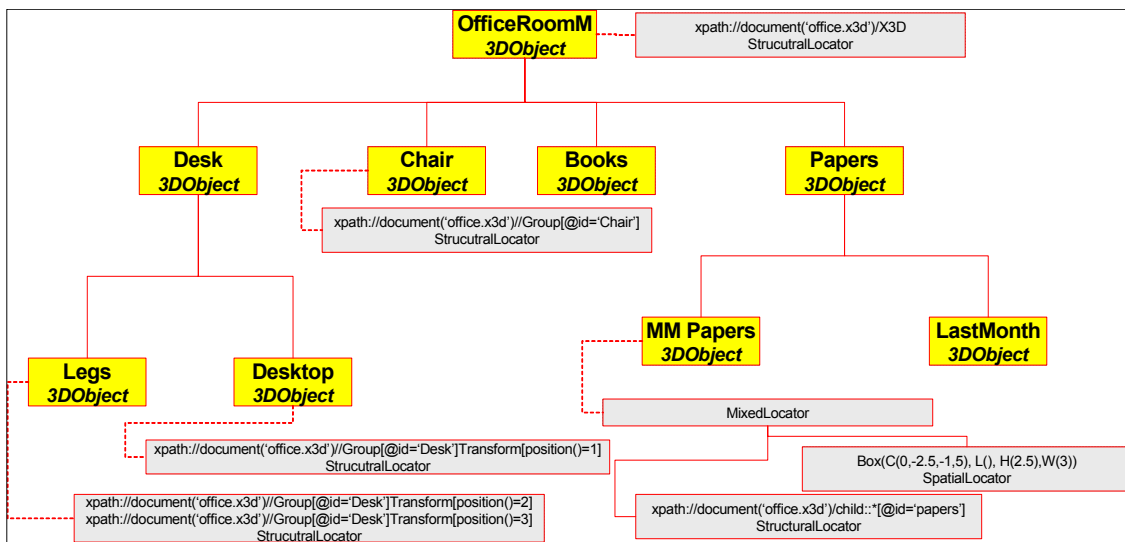


Figure 3: Logical structure of the real world scene.

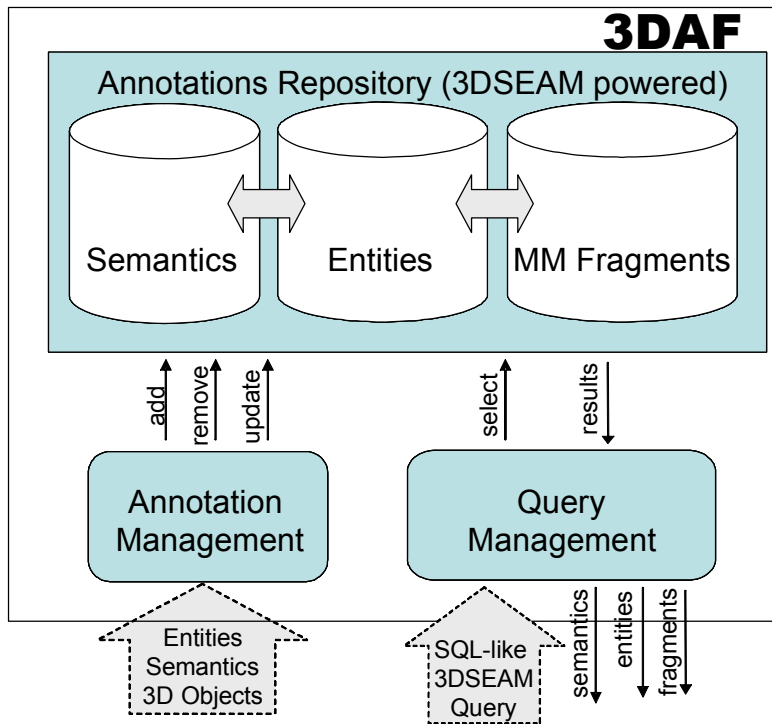


Figure 4: 3DAF architecture

multimedia fragment, 2) get the semantics associated to the entity. We have made this choice in order to preserve the maintainability. Semantics can change (adding new information, removing obsolete data) without modifying the multimedia fragment and vice-versa.

The *Annotation Management* and the *Query Management* modules introduce an intermediate layer between the repository organisation and the external management of the framework. So, the outside world is not concerned with the internal representation of information. This intermediate level supports the cooperation between several 3DAF frameworks independently from the specific 3DSEAM instances representation format (MPEG-7, RDF, OWL, DBMS, etc.). The insertion into the framework of annotations and indexing results is performed by the *Annotation Management* module. Each insertion is translated into a specific *add* operation applied onto the specific repository. The *remove* and *update* operations are executed in a similar manner.

The *Query Management* module treats requests sent by external modules or applications. SQL-like queries sent to the *Query Management* module are formulated using 3DSEAM concepts (3DObject, Entity, Semantics, Media Profile, etc.) and not implementation-specific representation convention. This module translates 3DSEAM concept-based queries into storage-dependent queries.

In order to support flexible description, we adopt the following notation. The ‘.’ concerns the navigation through the *static* structure of the model. For example, if *mf* is a *MultimediaFragment*, *mf.Entity* denotes the *Entity* associated with the *mf*. The ‘:’ allows the navigation through the flexible part of the model (concept properties: semantic profiles, media profile attributes, appearance properties, etc.). The left part

denotes a concept and the right part the name of a property. Hence, if *e* is an *Entity* then *e:ID* denotes the *ID* property of the entity *e*. In some cases the right-hand property can be structured (e.g. a semantic profile). The ‘:’ can then be used to access subsequent properties. For instance, if a semantic profile called *Classification* which contains information about categories of object is created, the notation *s:Classification:class* is used in order to access the *class(es)* – in the *Classification* profile – associated with a specific *Semantics* instance *s*. In Figure 5, two 3DSEAM queries are shown. The first one asks for the semantic class(es) associated with a specific 3DObject. The second one asks for some location information of a 3DObject which belongs to the *building* and *floor* semantic classes.

```

1) Retrieve the classification for the 3D object whom id is $FRAG_ID

SELECT s:Classification:class FROM Semantics
WHERE s.Entity.3DObject:ID=$FRAG_ID

2) Retrieve media locators of object that belongs to the "building" and "floor" class

SELECT f.MediaLocator FROM 3DObject
WHERE f.Entity.Semantics:Classification:class="building"
AND f.Entity.Semantics:Classification:class="floor"

```

Figure 5: SQL-like 3DSEAM queries

The results (semantics, entities description or fragments descriptions) are transformed into an auto-descriptive XML format. Specific tools can then be used to analyze the semantics and to reconstruct/reassemble 3D models.

In the next section, we propose an MPEG-7 instantiation of the 3DSEAM model and its associated 3DAF framework. We have chosen an MPEG-7 approach to represent the instances of a 3DSEAM model since MPEG-7 is known for having a great potential for describing and annotating multimedia documents.

## 5 An MPEG-7 annotation framework

### 5.1 MPEG-7 overview

Stemming from research efforts of the Moving Picture Experts Group (MPEG) working group, MPEG-7 is a standard that addresses the semantic description of media resources. Even if MPEG-7 has been proposed in the context of digital audio and video data, it is highly extensible and could cover other areas. Due to its high capability of evolution, we consider MPEG-7 as a valuable candidate for fulfilling the requirements needed for associating semantic annotations with a 3D scene.

MPEG-7 provides multimedia content description utilities for audio and visual contents: Descriptors (D), Descriptors Schemes (DS) and a Description Definition Language (DDL). Descriptors are indexing units describing the visual, audio and semantic features of objects. Description Schemes group several D and other DS into structured, semantic units. The DDL defines the syntax for creating new DS. Derived from the XMLSchema, it ensures the extensibility of the standard.

The existing DS cover the following areas: visual description (VDS), audio description (ADS) and multimedia content description (MDS). The VDS and ADS are linked to the physical, logical or semantic organisations of the document described by MDS. MDS offers DS for characterising both the physical and the logical structures of a multimedia content. It ensures also the semantic description. The basic structural element is called a *segment*. A segment corresponds to a spatial, temporal or spatio-temporal decomposition of the content. A segment (Audio Segment, Visual Segment ...) can be decomposed into smaller segments leading to a hierarchical segmentation of the media content. Then, each segment is indexed using the available set of (visual/audio/...) descriptors and description schemes.

In order to integrate 3D content description into MPEG-7, we have analysed the MPEG-7 predefined locators and we have proposed some extensions in order to localize objects in 3D

scenes [Bilasco et al. 2005a] which are briefly recalled in the next section.

### 5.2 Locating 3D contents in MPEG-7

MPEG-7 supports two main types of localization descriptors: the `MediaLocatorType` (a pointer-like locator) and the `RegionLocatorType` (a geometric-like locator).

The `MediaLocatorType` could be used directly in order to structurally localize 3D objects. However, the fact that the `MediaLocator` can only contain a unique URL, does not meet the requirements of the structural localization in the case of 3D where multiple URIs are necessary. For instance, in order to create large 3D scenes, a common technique consists in decomposing it into several X3D files. Objects can then be represented over several files. Several URIs are then needed to point at this kind of objects. Thus, we introduce the `StructuralLocatorType` as an extension of the `MediaLocatorType` in order to support the localization of this kind of objects. The complete DDL definitions are available in [Bilasco et al. 2005a].

The MPEG-7 `RegionLocatorType` building blocs are assembled from rectangular boxes and polygons. We propose a `_3DRegionLocatorType` as a 3D specific region definition. The localization of 3D objects can be obtained by integrating 3D region locators (this guarantees the geometric localization) with structural locators (selection of content units) into a generic `_3DLocatorType`.

In order to support the annotation and the indexing of 3D media objects, we extend the predefined `MultimediaSegmentType` DS. The `_3DObjectType` corresponds to a segment in a 3D scene and is defined by at least one `_3DLocator`.

In the next section, we illustrate the choices made in order to instantiate the 3DSEAM model using MPEG-7.

### 5.3 Instantiating the 3DSEAM model using MPEG-7

The MPEG-7 instantiation aims at maintaining as much as possible the independence between the different dimensions of an entity: its semantics and its multimedia materializations. In order to guarantee this independence, for each 3D scene or object, we use two independent MDS description schemas: the Semantic

```

1:<Mpeg7 ...>
2: <Description xsi:type="SemanticDescriptionType">
3: <Semantics>...
4: <SemanticBase xsi:type="ObjectType">...
5: <Object id="White_building">
6: <Label href="onto://imag.fr/Classification">
7: <Name>Classification</Name>
8: <Term>
9: <Name>class</Name>
10: <Definition>building</Definition>
11: </Term>
12: </Label>
13: <ObjectRef refid="White_building_1st_floor"/>
14: <ObjectRef refid="White_building_2nd_floor"/>
15: </Object>
16: <Object id="White_building_1st_floor">
17: <Label href="onto://imag.fr/Classification">
18: <Name>Classification</Name>
19: <Term>
20: <Name>class</Name>
21: <Definition>floor</Definition>
22: </Term>
23: </Label>
24: </Object>
25: <Object id="White_building_2nd_floor">
26: ...</Object>
27: <Object id="Dark_building">
28: ...</Object>
29: <Object id="Aerial_passages">
30: <Label href="onto://imag.fr/Classification">
31: <Name>Classification</Name>
32: <Term>
33: <Name>class</Name>
34: <Definition>passage</Definition>
35: </Term>
36: </Label>
37: <ObjectRef refid="passage_1st_level"/>
38: <ObjectRef refid=" passage_2nd_level"/>
39: </Object>
40: ...

```

Figure 6: Example of 3DSEAM entity description using MPEG-7

```

1: <Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
2: <Description xsi:type="ContentEntityType">
3: <MultimediaContent xsi:type="MultimediaType">
4: <Multimedia xsi:type="_3DObjectType" id="White_building_segm">
5: <Locator xsi:type="StructuralLocatorType">
6: <MediaLocator>
7:   xpath://document("buildings.x3d")/child::*[@id='WHITE']
8: </MediaLocator>
9: </Locator>
10: <MediaProfile>...</MediaProfile>
11: <Geometry>...</Geometry>
12: <Appearance>...</Appearance>
13: <EntityRef idref="White_building" />
14: <MediaSourceDecomposition>
15: <MultimediaRef idref="White_1st_floor_segment" />
16: <MultimediaRef idref="White_2nd_floor_segment" />
17: </MediaSourceDecomposition>
18: </Multimedia>
19: <Multimedia xsi:type="_3DObjectType" id="White_1st_floor_segm">
20: <Locator xsi:type="_3DLocatorType">
21: <MediaLocator>
22:   xpath://document("buildings.x3d")/child::*[@id='WHITE']
23: </MediaLocator>
24: <SpatialLocator xsi:type="Box3D" x="0" y="-3"
25:   z="0" l="12" h="1" w="2" />
26: </Locator>
27: <EntityRef idref="White_building_1st_floor" />
28: </Multimedia>
29: <Multimedia xsi:type="_3DObjectType"
30:   id="White_2nd_floor_segm">
31: ...
32: </Multimedia>
33: </MultimediaContent>
34: </Description>
35: </Mpeg7>

```

Figure 7: Example of an MPEG-7 fragment description inside the 3DSEAM model.

Description Schema (*Semantics*) and the Multimedia Content Description Schema (*MultimediaFragment*).

The *Entity* class and the *Semantics* class of the 3DSEAM model are instantiated using the *Semantic Description Schemas*. The semantics associated with the entity is introduced by MPEG-7 semantic elements. The semantic elements available in the Semantic Base Description Schema are *labels*, *definitions*, *properties* and *relations*.

The Multimedia Content Description Schema ensures the decomposition of the scene into independent 3D objects characterised using the *\_3DObjectType*.

Figure 6 and Figure 7 show an XML instantiation of the model using the extension we propose for MPEG-7. The semantic description is shown in Figure 6, whereas the multimedia fragment description is shown in Figure 7. The example focuses on the indexing of the scene described in Figure 1a. A first (semantic) description introduces in Figure 6 the following main entities: *White building* (lines 5-15), *Dark building* (lines 27-28) and *Aerial passages* (lines 29-38). The *white building* and the *Aerial passages* entities are further on divided in sub-sequent entities. Only, the definitions of the *White building floors* are presented in Figure 6 : the *White building 1<sup>st</sup> floor* (lines 16-24), the *White building 2<sup>nd</sup> floor*

(lines 25-26), etc.

We use the Label DS to introduce semantic profiles (the Classification profile lines: 6,17 and 30 in Figure 6). For each semantic profile a *Label* element is instantiated. The *href* attribute of a *Label* (lines 6, 17 and 30) is overloaded so that it corresponds to an ontology entry that defines the profile. In order to facilitate the usage (especially the query formulation), an alias is defined by the *Name* child element of *Label*. The semantic properties of a profile are introduced by *Term* elements (lines: 8,19 and 32 in Figure 6) . The name of the property (specific to the defined semantic profile) is defined by the *Name* element (lines 9, 20, 33). The value of the property is specified by the *Definition* element (lines 10, 21, 34). Although not illustrated here, semantic relations can be introduced using the *Term* *termId* and *relation* attributes. The *termId* points at the related property and the *relation* attribute indicate the nature of the relation.

Some multimedia fragments (*\_3DObjectType* instances) are presented in Figure 7. We focus on some fragments related to the *White building* object: the *white building segment* (lines 4-18) and the *White building 1st floor segment* (lines 15-28). The entire building is pointed at using a simple structural locator (lines 6-8), while the first floor is pointed at with a composite locator: a structural one (lines 21-23) and a geometric

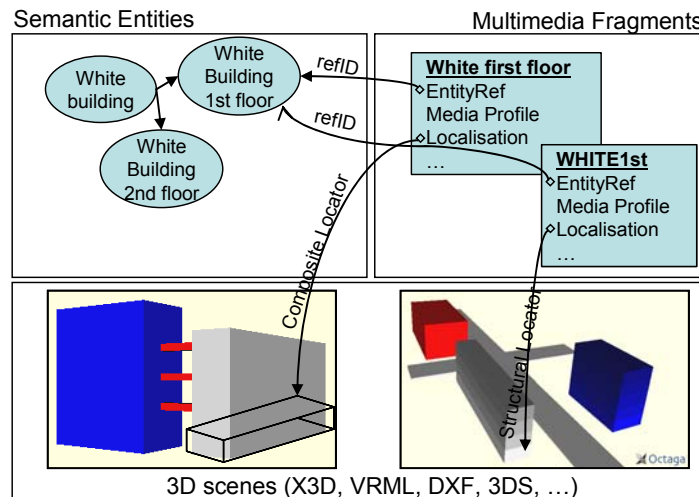


Figure 8: Global organization schema for the MPEG-7 based Annotation Repository in 3DAF

one (lines 24-25). The LogicalStructure of the scene is introduced using the `MediaSourceDecomposition` element (line 14). The `Appearance`, the `Geometry`, the `Topology` and the `MediaProfile` are directly stored within the `_3DObjectType` description. The link between the entities and the corresponding 3D units is done using the `id` of the entity through the `EntityRef` element associated with each 3D object (line 13 for the `White building` entity). Semantics can then be associated with each of the multimedia segments.

#### 5.4 A reuse-oriented MPEG-7 instance of 3DAF

In this section, we discuss the instantiation of 3DAF modules using an MPEG-7 annotation representation.

We start by illustrating the general organization of the Annotations Repository. In Figure 8, three independent data repositories are shown. Each repository is concerned with a particular topic: the semantic annotations (on the left of the figure), the multimedia fragments description (on the right of the figure) and the 3D models (at the bottom of the figure). Multimedia fragments are linked to semantic entities. The fragments are defined by some locators that point at the materialization of the multimedia fragment inside a specific multimedia document or 3D scene.

In the example presented here, the `White first floor` segment (from the scene in Figure 1a) and the `WHITE1st` segment (from the scene in Figure 1b) are associated with the same semantic entity `White building 1st floor`. Each segment is linked to the initial scene through a locator. Some reuse scenarios can be imagined from this architecture. For instance, querying all the materializations of the entity `White building 1st floor` can be performed by filtering the multimedia segments registered into the segment repository in order to get those associated with the corresponding entity. The location links attached to the segments allow the retrieval of the

specific pieces of X3D code modelling the entity.

```

1) Retrieve the classification for the 3D object
whom id is $FRAG_ID

for $class in doc("entities.mpeg7.xml")//Object/Label
  [Classification='Name/text()']/Term[class='Name/text()']/Definition
let $frag:=doc("fragments.mpeg7.xml")//Multimedia[$FRAG_ID=@id]
where $frag/EntityRef/@idref=$class/ancestor::Object/@id
return <item type="Semantics:Classification.class">{$class}</item>

2) Retrieve media locators of object that belongs
to the "building" and "floor" class

for $locator in doc("fragments.mpeg7.xml")//Multimedia
  [ "_3DObjectType" = @xsi:type ]/Locator/MediaLocator
let $class:=doc("entities.mpeg7.xml")//Object
  [ @id=$locator/ancestor::Multimedia/EntityRef/@idref, @id ]/
  Label[contains(Name/text(), "Classification")]/
  Term[contains(Name/text(), "class")]
where (contains($class/Definition/text(), 'building')
  and contains($class/Definition/text(), 'floor'))
return
<item type="Object3D.MediaLocator ">{$locator}</item>

```

Figure 9: Querying MPEG-7 annotation repositories using XQuery

In order to be able to query the MPEG-7 repositories, the Query Management and the Annotation Management modules have to translate the 3DSEAM (select, respectively insert/update) queries into MPEG-7 queries. MPEG-7 query languages exists [Fatemi et al. 2003] but they are specific to audio-video dimension of multimedia content and to our knowledge they are not extensible. As MPEG-7 is an XML language we intend to use XQuery in order to extract information. 3DSEAM concepts and properties have to be translated into XPath expressions that point at their corresponding location in the MPEG-7 instantiation files. For instance, the expression `SEMANTICS:Classification:class` (the `class` property of the `Classification` semantic profile) is translated into `//Object/Label[Name/text()='Classification']/Term`

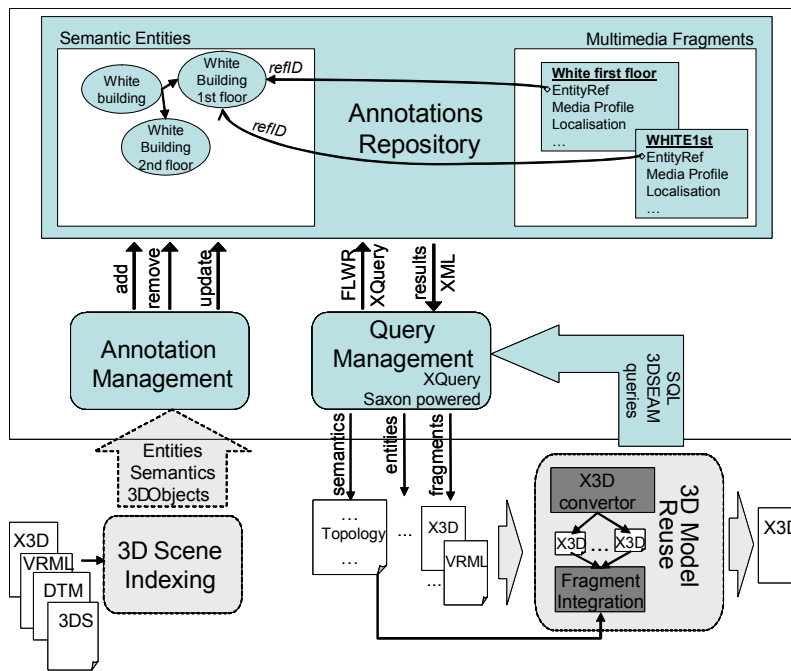


Figure 10: A 3DSEAM-based indexing framework for 3D model reuse

[Name/text()='class']. Further on, the SELECT queries can be translated into FLWR expressions. In Figure 9 we show the XQuery equivalent of the queries formulated in Figure 5.

The process of translation is not automatic since the translation from XQuery to SQL and the inverse translation are not trivial. Some recent works [Krishnamurthy et al. 2004] in the field of XML databases offer partial answers to the translation from XQuery to SQL. However, no universal (that treats all queries categories) and reversible solution exists. In order to solve this problem we plan to build predefined set of template queries. This solution is more restrictive than a free query approach. However, generally, users prefer to fill in forms and retrieve results. A typical user is not very keen on formulating by itself complex query. Research engines on the Web are good examples of this observation, as users restrain their researches to a sequence of keywords without using the logical operators available.

Besides the core modules, two external modules (see Figure 10): 3D Scene Indexation and 3D Model Reuse, are proposed. Their role is, respectively, to feed the annotation framework with relevant information and to reuse and assemble existing pieces of 3D data into new 3D object and scenes.

The *3D Model Reuse* module assembles the raw information issued from the *Query Management* module into coherent 3D models. The module consists of two components. A first component is used for retrieving fragments (in the initial format: X3D, VRML, 3DS...) corresponding to the description extracted from the *Annotations Repository*. These fragments are all converted into X3D. The resulting X3D fragments are then assembled by the *Fragment Integration* component using indexation information (topology, real world localization of objects, etc.) extracted from the *Annotation Repository*.

## 6 Conclusion

In this paper, we have presented 3DAF a generic annotation framework. This framework is built on the basis of an extensible annotation model (3DSEAM) that supports the description of 3D object semantic features. The semantic added to the pure X3D geometric modelling of the scene enhances the management process of 3D objects.

3DAF provides a module (Annotation Management module) that deals with the organization of the semantic annotations. Two other modules are proposed assuring the independence between specific implementation of the 3DSEAM model and the 3DSEAM queries formulated by external agents (users or applications). Queries, that take into account the variability that can appear in terms of semantic profiles properties, can be built using the naming convention considered.

An instantiation of the 3DSEAM model using MPEG-7 is achieved in order to build a 3DAF-like framework that supports the reuse of 3D objects. Some issues related to the translation of 3DSEAM queries into XQuery expression are discussed.

Our future research efforts will be directed towards the implementation of the framework in charge of the indexing, the annotation and the reuse of 3D objects. Two points must be addressed in order to construct this framework: 1) the instantiation of the 3DSEAM for the objects contained in a 3D scene by manual, semi-automatic or automatic localization and

annotation, and 2) the use of a formatter linked to the query language that will specify the type of presentation for the result items (raw information, dynamically generated 3D scenes). The automatic translation of 3DSEAM SQL-like queries into MPEG-7 XQuery will also be explored. The fact that we query a specific model that, partially, has a static structure (the main 3DSEAM concepts and their relations) could help in developing a tool that enables automatic translation.

## 7 References

- AUTODESK LTD. 1997. *3D-Studio File Format (.3ds) – revision 0.93*, online version <http://www.dcs.ed.ac.uk/home/mxr/gfx/3d/3DS.spec> (checked on 8th February 2006).
- AUTODESK LTD. 2006. *Autocad 2006 DXF Reference*, online version [http://images.autodesk.com/adsk/files/dxf\\_format.pdf](http://images.autodesk.com/adsk/files/dxf_format.pdf) (checked on 8th February 2006).
- BILASCO I.M., GENSEL J., VILLANOVA-OLIVER M. AND MARTIN H. 2005. On indexing of 3D scenes using MPEG-7, in *Proceedings of ACM Multimedia 05*, Singapore, 471-474.
- BILASCO I.M., GENSEL J., VILLANOVA-OLIVER M. AND MARTIN H. 2005. 3DSEAM: a model for annotating 3D scenes using MPEG-7, in *Proceedings of IEEE International Symposium on Multimedia '05*, Irvine, California, 310-319.
- BOOCH G., RUMBAUGH J., AND JACOBSON I. 1999. *The Unified Modeling Language User Guide*. Addison-Wesley Longman Inc.
- Caligari Corporation. 2002. Caligari trueSpace file format, online version <http://www.caligari.com/download/zip/scncob65.zip> (checked on 8th February 2006).
- DEL BIMBO A. AND PALA P. 2004. Content Based Retrieval of 3D Data, in *Lecture Notes in Computer Science*, Volume 3287, 13-24.
- DINESH K. P., DOEL K., DOUG J. L., LANG J., LLOYD J. E., RICHMOND J. L. AND YAU S. H. 2001. Scanning physical interaction behavior of 3D objects, in *Proceedings of SIGGRAPH'01*, ACM Press/ACM SIGGRAPH, Los Angeles, Computer Graphics Proceedings, Annual Conference Series, ACM, 87-96.
- FATEMI N., KHALED O. A. AND CORAY G. 2003. An XQuery adaptation for MPEG-7 documents retrieval. in *Proceedings of XML Conference and Exposition*, Philadelphia.
- FOLEY P.G., MAMAGHANI F., AND BIRKEL P.A. 1998. *The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) Development Project*, online version <http://sedris.org/download/documentation/it103a60.pdf> (checked on 8 February 2006).
- GRUBER T. 1995, Towards principles for the design of ontologies used for knowledge sharing, in *International Journal of Human-Computer Studies*, Volume 43 , Issue 5-6 Nov./Dec., 907-928.
- HASTING A. AND FERGUSON S. 1994. *Lightwave 3D Object File Format*, online version <http://www.dcs.ed.ac.uk/home/mxr/gfx/3d/LWOB.txt> (checked on 8th February 2006).

- HETHERINGTON R.E, SCOTT J.P. 2004. Adding a fourth dimension to three dimensional virtual spaces, in *Proceedings of the 9th International Conference on 3D Web Technology*, Monterey, California, 163-172.
- KIM D.H., YUN I.D. AND LEE S.U. 2004. A comparative study on attributed relational graph matching algorithms for perceptual 3-D shape descriptor in MPEG-7, in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, New York, USA, 700-707.
- KRISHNAMURTHY R., KAUSHIK R. AND NAUGHTON J.F. 2004. Efficient XML-to-SQL Query Translation: Where to Add the Intelligence ?, in *Proceedings of VLDB 2004*, Toronto, Canada, 144-155.
- MARTINEZ J. M. AND KOENEN R. 2002. MPEG-7: The Generic Multimedia Content Description Standard, Part 1, *IEEE Multimedia*, IEEE Computer Society Press, Volume 9, Issue 2, 78-87.
- OHBUCI R., NAKAZAWA M. AND TAKEI T. 2003. Retrieving 3D Shapes based on Their Appearance, in *Proceedings of MIR'03*, Berkeley, CA, USA, 39-46.
- PFUND M. 2001. Topologic data structures for a 3D GIS, in *Proceedings of ISPRS'01*, vol. 34, Bangkok, Thailand, 233-237.
- SCHAEFER J. GRANIOLA B, HEARD P. AND DE LA CRUZ J. 2002. *Web-Based Repository for STRICOM SNE Objects*, online at [http://www.sedris.org/download/documentation/web\\_paper.pdf](http://www.sedris.org/download/documentation/web_paper.pdf) (checked on 8 February 2006).
- ZLATANOVA S., RAHMAN A. A. AND SHI W. 2004. Topological models and frameworks for 3D spatial objects, *Journal of Computers & Geosciences*, May, Vol 30, Issue 4, 419-428.
- WEB3D CONSORTIUM. 2004. *Information technology — Computer graphics and image processing — Extensible 3D (X3D) — Part 1: Architecture and base components*, ISO/IEC 19775-1:2004, online version [www.web3d.org/x3d/specifications/](http://www.web3d.org/x3d/specifications/) (checked on 8 February 2006).