ETISEO

Internal Technical Note

Data structure and output format

3 – Approved

IN_ETI_1_011

Document Title : Data structure and output format

Document version : 3

Document status : Approved

Date : 11 May 2006

Availability :

Authors : Silogic, Inria

Abstract This document presents the data terminology and structure used in Etiseo program.

Keyword List Metadata, bounding box, PETS, xml structure

DOCUMENT CHANGE LOG

Document Issue.	Date	Reasons for change
V1 Draft 1 V1 Draft 2 Version 2 – 0 Version 3	25 Oct. 2005 24 Nov 2005 6 January 2006 11 May 2006	Creation Inria revisions Apply changes after seminar feedback. Data set 2 delivery

APPLICABLE AND REFERENCE DOCUMENTS (A/R)

A/R	Reference	Title
Ref 1		Viper-GT Tutorial

Table of contents

1. INTRODUCTION

This document presents the data structures which are needed for the evaluation campaign and that should be provided by participants. The evaluation of algorithm performances is constructed according to several levels of difficulties and enables to get detailed information about the evaluated algorithms. The participant is not obliged to deliver all information presented below as it depends on what its system is able to do. Nevertheless, the more detailed data are given to the evaluator, the more detailed the evaluation results will be. Any complementary information to the information described below and delivered by participants is welcome but may not be evaluated.

2. DATA TERMINOLOGY

This section enumerates and defines all the vocabulary describing data used in a video understanding evaluation process.

Image: array of pixel generated at a time step by a video camera (e.g., composite, CCD, CMOS, PTZ, omni directional). An image is characterized by a timestamp (year, month, day, hour, minute, second, millisecond) and can correspond to a frame interleaved or not. An image can be of the following type: colour, black and white, infrared and with different compression levels.

Video sequence: temporal sequence of images which are generated by a video camera. A video sequence can be represented as a live stream (e.g., composite signal, MJPEG stream), as a file (e.g., a MPEG4 encoded file) or as a sequence of files (e.g., a sequence of JPEG files).

Video clip: a part of a video sequence, which corresponds to a particular situation to be evaluated.

Scene: the physical space where a real world event occurs and which can be observed by one or several video cameras. A scene without any physical object of interest is called an empty scene.

Blob: 2D image region that has been segmented based on regions (e.g., homogeneous in motion, colour, energy or texture information) or contours (e.g., using a shape model). This region can be defined as a set of pixels (not necessarily connected) or as a polygon delimiting its contour. It can be characterized by 2D features such as a colour histogram, a density, a 2D width and height.

Moving region: a blob that has been created following a motion criteria (e.g., either optical flow or reference image subtraction).

Physical object: a real world object in the scene. There are two types of physical objects: physical object of interest and contextual object.

Physical object of interest: a physical object evolving in the scene whose class (e.g., person, group, vehicle) has been predefined as interesting by end-users and whose motion cannot be foreseen using a priori information. It is usually characterized by a semantic class label, 2D or 3D features (e.g., 3D location, width and height, a posture, a trajectory, a direction, a speed), a list of blobs, an initial tracking time, a camera number for the camera which is the best seeing the object (in a multi camera configuration), an identifier. An identifier can either be defined locally to the current image, globally on the video sequence or globally on a scene (in a multi camera configuration).

Contextual object: a physical object attached to the scene. The contextual object is usually static and whenever in motion, its motion can be foreseen using a priori information. For instance, it can be in motion such as a door, an elevator, a fountain, a tree or displaceable (by a human being) such as a chair, a luggage.

Bounding Box (BBox): 2D Box including a physical object in an image. It is represented by the top left corner point coordinates (x,y), the width and the height. The system coordinate is for X a left to right axis and for Y a top to bottom axis.

Event: generic term to describe any event, action or activity happening in the scene and visually observable by cameras. Events of interest can either be predefined by end users or learned by the system. Events are characterized by the involved objects of interest (including contextual objects and zones of interest), their starting and ending time and by the cameras observing the events. Examples of events are “detection of a vehicle inside a zone”, “detection of an abandoned bag”, ”a meeting between two people”...

Ground truth data: data given by a human operator and which describe real world expected results (e.g., physical objects, events) at the output of a video understanding algorithm. These data are supposed to be unique and corresponding to end user requirements even if in many cases, this information can contains errors (annotation bias). These data can be written in a XML or MPEG7 format.

Annotation: information associated to a video clip including ground truth data plus other types of information about technical difficulties (e.g., shadows) and recording conditions (e.g., weather conditions) of the video clip under consideration. These annotations can provide several types for false or incorrect results (e.g., wrong classification, wrong detection).

Reference data: data supposed to be constant and unique, corresponding to a functionality of a video understanding task and used to evaluate the output of a video understanding algorithm at a given task level. Reference data include ground truth data, data given by a video expert and data computed from all annotation and contextual information. For instance, the 3D position of a person is a reference data computed from the bounding box given by a video expert and the calibration matrix. In addition, rules should be given to video experts in order to define as objectively as possible particular data. For instance, for a partially occluded person, one can choose to draw the bounding box for the visible part only or for the full object (including its hidden part).

Crowd: group of at least 7 persons. A bounding box delineates all the involved persons.

Multiple vehicles: group of at least 7 vehicles. A bounding box delineates all the involved vehicles.

3. DATA STRUCTURE FOR ALGORITHM RESULTS

The metadata defines the data structure. Two kinds of data are represented in a video understanding system: physical objects and events. This section presents the different attributes characterizing these data at the output of a video understanding system. The following section presents the different attributes added to these data for the annotation and reference data creation. In the final section, we propose a xsd structure supporting these data structures. Each participant is supposed to provide a xml result file containing these data for each frame at the output of their whole video understanding system. These files will be used to evaluate the four video understanding tasks (i.e., the files used to evaluate the detection and the tracking task are identical, not separate ones).

3.1. PHYSICAL OBJECT

Please find below a list of physical object attributes, which characterize the output of a video understanding algorithm and which are required for the evaluation purpose. Mandatory attributes are marked with a star.

- ***Id**: Integer number associated to a physical object, unique for a video clip,
- ***Start time**: Integer representing the first frames of object apparition,
- ***Type**: Enumeration defining the type of object: for contextual object = door, chair…; for physical object of interest = person, vehicle, group_of_person, group_of_vehicle, crowd, multiple_vehicle…
- **Subtype**: Enumeration defining the sub-level of type. Example for type vehicle: car, truck, loader …
- ***Info2d**: List of integers for the 2D bounding box of the object in current image,
- **Info3d**: List of integers for the 3D position and size of the object,
- **Links**: Integers representing the Id of father objects in the video clip,
- **Speed**: Integer representing the 3D speed of the object.

3.2. EVENT

Please find below a list of event attributes that can be required for evaluation purposes:

- ***Id**: Integer characterizing the event, unique for a video clip,
- ***Name**: Name of the event (identification of the event). The list of event names to recognise in a sequence is delivered with the video,
- **Start and end time**: Integer corresponding to First and last frame of event detection,

- **Status**: Enumeration, 0-Previously tracked event; 1-New event Target; 2-Re-appeared event,
- **Physical objects**: List of physical objects ID involved in this event (Id of objects used in the tracking phase),
- **Contextual objects**: List of contextual objects ID involved in this event (Id of objects described in the context, provided with the video data set),

4. DATA STRUCTURE FOR GROUND TRUTH AND REFERENCE DATA

This section presents the different attributes characterizing the data in the annotation task.

4.1. PHYSICAL OBJECT

Please find below a list of physical object attributes, which characterize annotation and reference data task in addition to algorithm output attributes; Id, Start time, Type, Subtype, Info2d, Info3d, Links, Speed, +:

- **Nature**: Enumeration, contextual object or physical object of interest,
- **Status**: Enumeration, 0-Previously tracked object; 1-New Tracked Target; 2-Re-appeared Target, e.g. re-appeared from short or long term occlusion; 3-Merging with other tracked targets; 4-Splitting, e.g. one target is splitting into two; 5-Lost tracked target; 6-Out of Field of View,
- **Occluded**: Enumeration, 0-Not occluded; 1-Partially occluded; 2-Totally occluded,
- **Connected with**: Id of connected object (door 4 connected with vehicle 3)). The carried object (ex: a bag, a person, a door) has a connection to the "holder" (resp. person, group, vehicle).

4.2. EVENT

Event attributes in the annotation and reference data task are the same that the ones used at the output of a video understanding system (Id, Name, Start and end time, Status and involved physical and contextual objects).

The data structures and their attributes values of a video clip can be represented in two different ways:

- Frame-by-Frame: in each frame, the different attributes values of each data present in the scene at time t are given. This is a usual way for the output of a video understanding system: at each instant t we know the system state.

- Per descriptor instance: for each data structure, its different attributes values are declined along the sequence. This is more the case at the output of the annotation task.

A bridge is developed to pass from one representation to the other.

5. DATA STRUCTURE FOR CONTEXT ANNOTATION

Please find below a list of contextual object attributes:

- **Id**: Integer characterizing the contextual object, unique for a scene,
- **Type**: Enumeration defining the type of the contextual object (ex: zone, door, equipment…),
- **Name**: Name of the contextual object (identification of the object),
- **Polygon**: List of 2D or 3D points defining the contextual object.

The context information file is communicated to participant in a Viper-Gt file format described afterwards.

6. OUTPUT FORMAT

For each of the above video clip annotation composition, we associate a xml format:

- Frame-by-Frame structure: an extension of PETS xml file format (included event recognition) is proposed for compatibility and time saving reason.
- Per descriptor instance: a metadata configuration is created with Viper GT tool for ground truth creation and can be used by participants.

Every participant is free to use its own file format but we warmly recommend to use one of the two file formats described above.

Bridges between PETS xml format and Viper xml format and vice versa will be produced. In addition, in case the participant output format is different, a bridge to one of the ETISEO format will be created according to its video clip annotation composition.

The dataset contains an example of both file formats.

6.1. PETS XML EXTENDED FORMAT

6.1.1. PROPOSED FORMAT

The header contains different information about the conditions of the output creation. It is very useful that each participant fill in this information for the evaluator.

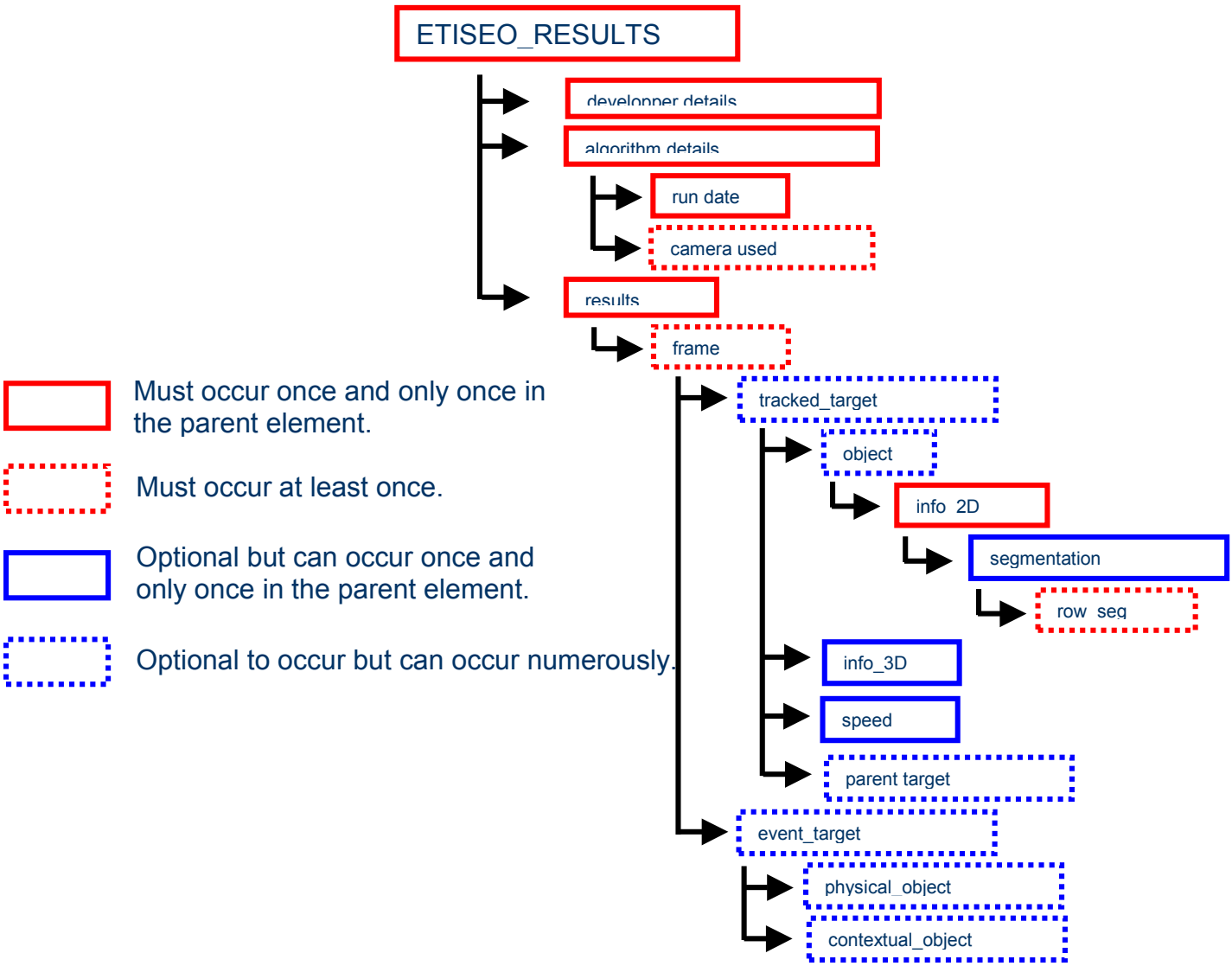
The Xml format is shown below. Extensions from PETS format have been **marked in blue.**

<?xml version="1.0" encoding="UTF-8"?>

```
<ETISEO_RESULTS xmlns="http://www.silogic.fr/etiseo/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.silogic.fr/etiseo/ETISEO_RESULTS.xsd">
        <developer_details author="" email="" institution=""/>
        <algorithm_details dataset_name="VS1" is_testing_results="true" dataset_capture_number="11"
task_developed_for="" version="">
                <run_date day="0" month="0" year="0"/>
                <cameras_used camera_id="1"/>
        </algorithm_details>
    <results>
        <frame frame_id="0">
        </frame>
        <…>
        <frame frame_id="151">
            <tracked_target target_id="1"  start_frame_track="150"  end_frame_track="151"  status="0">
                <info_3d x="-8.8" y="16.7"  z="0"  width="0.8"  length="0.4"  height="1.8">
                </info_3d>
                <speed=”3.25” />
                <object object_id="1"  classification="4">
                    <info_2d xmin="234"  xmax="285"  x_center="259"  ymin="574"  ymax="507"
y_center="540"  viewed_in_camera_id="1">
                        <segmentation>
                            <row_seg row_id="0" col_seg="0…1111…0"/>
                            <row_seg row_id="1" col_seg="0…1111…0"/>
                            <…>
                        </segmentation>
                    </info_2d>
                </object>
                <parent_target target_id="1" frame_id="150">
                </parent_target>
            </tracked_target>
            <tracked_target target_id="2"  start_frame_track="150"  end_frame_track="151"  status="0">
            <…>
            </tracked_target>
            <event_target target_id="10"  name="inside" start_frame="150"  end_frame="350"
status=”0” >
                <physical_object id="4" viewed_in_camera_id="1"/>
                <…>
                <contextual_object id="10" />
                <…>
            </event_target>

        </frame>
    </results>
</ETISEO_RESULTS>
```

6.1.2. SCHEMA DEFINITION



A detailed schema of the result file is also available on the dataset DVDs in the documentation folder (ETISEO_RESULTS.png).

6.1.3. REMARKS

a. Pets format contains object segmentation information that will not be used in the ETISEO program. It is recommended to eliminate this from your output for capacity reason.

b. Pets format supports multiple camera in the same xml but participant could provide one file per camera.

c. One Pets tag (an integer) supports classification information. Two levels of classification are presents in our data structure: Type and Sub-type. The code on 4 numbers is defined: the thousand and the hundred for Type attribute (100 possibilities) and the ten and unity for the Sub-type attribute (100 possibilities). The nature of object is deduced from is type. Number 0 is reserved for unknown category (0000 = Type is unknown and Sub-type is unknown). The table below give an example of code declination:

type		subtype		Classification code
0	unknown			0
1	person			100
		1	child	101
		2	baby	102
2	vehicle	0	unknown	200
		1	car	201
		2	truck	202
		3	bike	203
		4	ground_power_unit	204
		5	tanker	205
		6	aircraft	206
		7	push_back	207
		8	conveyor_belt	208
		9	loader	209
		10	jet_bridge	210
		11	van	211
		12	bus	212
		13	motorcycle	213
3	person_group			300
4	vehicle_group			400
5	door			500
6	bag			600
7	box			700
8	baggage			800
9	pushchair			900
10	trolley			1000
11	dustbin			1100
12	briefcase			1200
13	baby_car_seat			1300

Table 1. Object classification codes

You can provide any other code for categorisation purpose but you must communicate them to the evaluator. In this case, the translation will be made in the bridge.

d. One of the metrics for event evaluation evaluates the position of tracked targets involved in the scenario with its Info2D. This is why we precise in the xml the viewed_in_camera_id field for involved objects. This field is optional and if not mentioned the first camera and Info2D (between <object> tags) will be used in the evaluation.

6.2. GROUND TRUTH AND REFERENCE DATA XML FORMAT

The annotation tool chosen to generate the ground truth file is Viper-GT. Viper-Gt allows its own metadata definition and is adapted to our annotation needs. The xml viper format for descriptors and attributes is conformed to metadata structure define in section 3. The ground truth and reference data header xml format contains this data structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<viper xmlns="http://lamp.cfar.umd.edu/viper#" xmlns:data="http://lamp.cfar.umd.edu/viperdata#">
    <config>
        <descriptor name="Information" type="FILE">
            <attribute dynamic="false" name="SOURCETYPE" type="lvalue">
                <data:lvalue-possibles>
                    <data:lvalue-enum value="SEQUENCE"/>
                    <data:lvalue-enum value="FRAMES"/>
                </data:lvalue-possibles>
            </attribute>
            <attribute dynamic="false" name="NUMFRAMES" type="dvalue"/>
            <attribute dynamic="false" name="FRAMERATE" type="fvalue"/>
            <attribute dynamic="false" name="H-FRAME-SIZE" type="dvalue"/>
            <attribute dynamic="false" name="V-FRAME-SIZE" type="dvalue"/>
        </descriptor>
        <descriptor name="physical_object" type="OBJECT">
            <attribute dynamic="false" name="nature" type="svalue"/>
            <attribute dynamic="true" name="status" type="dvalue"/>
            <attribute dynamic="true" name="type" type="svalue"/>
            <attribute dynamic="true" name="subtype" type="svalue"/>
            <attribute dynamic="true" name="info2D" type="bbox"/>
            <attribute dynamic="true" name="info3D" type="info3d"/>
            <attribute dynamic="true" name="connected_with" type="dvalue"/>
            <attribute dynamic="true" name="links" type="svalue"/>
            <attribute dynamic="true" name="speed" type="dvalue"/>
            <attribute dynamic="true" name="occluded" type="dvalue"/>
        </descriptor>
        <descriptor name="event" type="OBJECT">
            <attribute dynamic="false" name="name" type="svalue"/>
            <attribute dynamic="true" name="status" type="dvalue"/>
            <attribute dynamic="false" name="physical_object" type="svalue"/>
            <attribute dynamic="false" name="contextual_object" type="svalue"/>
        </descriptor>
    </config>
</viper>
```

This "header" is included in all xml ground truth and reference data files. The rest of the file contains the physical objects and events instances and their attributes values defined for a particular sequence. Example for sequence ETI-VS1-CR-4-C1.avi:

```
<?xml version="1.0" encoding="UTF-8"?>
<viper xmlns="http://lamp.cfar.umd.edu/viper#" xmlns:data="http://lamp.cfar.umd.edu/viperdata#">
```

```
<config> <…>
</config>
<data>
    <sourcefile filename="file:ETI-VS1-CR-4-C1.avi">
        <file id="0" name="Information">
            <attribute name="SOURCETYPE">
                <data:lvalue value="SEQUENCE"/>
            </attribute>
            <attribute name="NUMFRAMES">
                <data:dvalue value="1000"/>
            </attribute>
            <attribute name="FRAMERATE">
                <data:fvalue value="12.0"/>
            </attribute>
            <attribute name="H-FRAME-SIZE">
                <data:dvalue value="640"/>
            </attribute>
            <attribute name="V-FRAME-SIZE">
                <data:dvalue value="480"/>
            </attribute>
        </file>
        <object framespan="1:20 40:100" id="4" name="physical_object">
            <attribute name="nature">
                <data:svalue value="physical_object_of_interest"/>
            </attribute>
            <attribute name="status">
                <data:dvalue framespan="1:1" value="1"/>
                <data:dvalue framespan="2:20" value="0"/>
                <data:dvalue framespan="40:40" value="2"/>
                <data:dvalue framespan="40:100" value="0"/>
            </attribute>
            <attribute name="type">
                <data:svalue framespan="1:20 40:100" value="person"/>
            </attribute>
            <attribute name="subtype"/>
            <attribute name="info2D">
                <data:bbox framespan="1:1" height="182" width="86" x="138" y="59"/>
                <data:bbox framespan="2:2" height="183" width="86" x="139" y="59"/>
                <data:bbox framespan="3:3" height="185" width="86" x="141" y="60"/>
                <data:bbox framespan="4:17" height="187" width="87" x="143" y="60"/>
                <data:bbox framespan="17:17" height="210" width="91" x="168" y="69"/>
                <data:bbox framespan="18:18" height="212" width="92" x="170" y="69"/>
                <…>
                <data:bbox framespan="40:100" height="212" width="100" x="300" y="100"/>
            </attribute>
            <attribute name="info3D">
                <data:bbox framespan="1:1" height="2" width="0.4" length="0.2" x="13.8" y="5.9" z="0"/>
                <data:bbox framespan="2:2" height="2" width="0.4" length="0.3" x="14.9" y="6" z="0"/>
                <data:bbox framespan="3:3" height="1.8" width="0.5" length="0.4" x="15" y="6.2" z="0"/>
                <data:bbox framespan="4:17" height="2" width="0.5" length="0.5" x="15.5" y="6.5" z="0"/>
                <data:bbox framespan="17:17" height="2" width="0.4" length="0.5" x="16" y="7" z="0"/>
                <data:bbox framespan="18:18" height="2" width="0.4" length="0.5" x="17" y="7.5" z="0"/>
                <…>
                <data:bbox framespan="40:100" height="2" width="0.5" length="0.4" x="30" y="10" z="0"/>
            </attribute>
            <attribute name="connected_with">
                <data:svalue framespan="2:20" value="4"/>
                <data:svalue framespan="40:100" value="4"/>
```

```
            </attribute>
            <attribute name="links"/>
            <attribute name="speed">
                <data:dvalue framespan="1:3" value="0.5"/>
                <data:dvalue framespan="4:17" value="1.3"/>
                <data:dvalue framespan="18:20" value="0.7"/>
                <data:dvalue framespan="40:100" value="2"/>
            </attribute>
            <attribute name="occluded">
                <data:dvalue framespan="1:20" value="0"/>
                <data:dvalue framespan="40:60" value="1"/>
                <data:dvalue framespan="60:80" value="2"/>
                <data:dvalue framespan="60:100" value="0"/>
            </attribute>
        </object>
        <object framespan="10:20" id="0" name="event">
            <attribute name="name">
                <data:svalue value="person_enter_corridor"/>
            </attribute>
            <attribute name="status">
                <data:dvalue framespan="10:10" value="1"/>
                <data:dvalue framespan="11:20" value="0"/>
            </attribute>
            <attribute name="physical_object">
                <data:svalue value="4"/>
            </attribute>
            <attribute name="contextual_object">
                <data:svalue value="1"/>
            </attribute>
        </object>
    </sourcefile>
</data>
</viper>
```

6.3. CONTEXT INFORMATION XML FORMAT

The context information is generated with Viper-Gt and uses the following metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<viper xmlns="http://lamp.cfar.umd.edu/viper#" xmlns:data="http://lamp.cfar.umd.edu/viperdata#">
    <config>
        <descriptor name="Information" type="FILE">
            <attribute dynamic="false" name="SOURCETYPE" type="lvalue">
                <data:lvalue-possibles>
                    <data:lvalue-enum value="SEQUENCE"/>
                    <data:lvalue-enum value="FRAMES"/>
                </data:lvalue-possibles>
            </attribute>
            <attribute dynamic="false" name="NUMFRAMES" type="dvalue"/>
            <attribute dynamic="false" name="FRAMERATE" type="fvalue"/>
            <attribute dynamic="false" name="H-FRAME-SIZE" type="dvalue"/>
            <attribute dynamic="false" name="V-FRAME-SIZE" type="dvalue"/>
        </descriptor>
        <descriptor name="context" type="OBJECT">
            <attribute dynamic="false" name="type" type="svalue"/>
            <attribute dynamic="false" name="name" type="svalue"/>
            <attribute dynamic="false" name="polygon2D" type="polygon">
```

```
                <data:polygon-type open="false"/>
            </attribute>
        </descriptor>
    </config>
</viper>
```

As Viper doesn't support yet 3D annotations, we replace for the 3D context the attributes polygon2D by a string (svalue) named polygon3D containing a list of 3D points

Example of a square zone:

```
<attribute name="polygon3D">
    <data:svalue value="(2.5 -1 0) (2.5 -2 0) (3.5 -2 0) (3.5 -1 0)"/>
</attribute>
```

6.4. OTHER OUTPUT FORMAT

Other output formats will be transformed in PETS xml structure if the organisation is Frame-by-Frame or in Viper xml format in the other case.

7. CONCLUSION

Data structure and organisation is defined in this reference document. Annotations rules and metrics are based on the metadata defined above. Please, refer to this paper for any question regarding output structure.