Semantic web technologies for video surveillance metadata

Chris Poppe, Gaëtan Martens, Pieterjan De Potter, Rik Van de Walle

Multimedia Tools and Applications, published online 14 September 2010

http://www.springerlink.com/content/n514j052014675p2/

Noname manuscript No.
(will be inserted by the editor)

Semantic Web Technologies for Video Surveillance Metadata

Chris Poppe · Gaëtan Martens · Pieterjan De Potter · Rik Van de Walle

Received: date / Accepted: date

Abstract Video surveillance systems are growing in size and complexity. Such systems typically consist of integrated modules of different vendors to cope with the increasing demands on network and storage capacity, intelligent video analytics, picture quality, and enhanced visual interfaces. Within a surveillance system, relevant information (like technical details on the video sequences, or analysis results of the monitored environment) is described using metadata standards. However, different modules typically use different standards, resulting in metadata interoperability problems. In this paper, we introduce the application of Semantic Web Technologies to overcome such problems. We present a semantic, layered metadata model and integrate it within a video surveillance system. Besides dealing with the metadata interoperability problem, the advantages of using Semantic Web Technologies and the inherent rule support are shown. A practical use case scenario is presented to illustrate the benefits of our novel approach.

Keywords Video Surveillance System · Semantic Web Technologies · Multimedia Standards · Reasoning · Video Analytics

1 Introduction

Video surveillance is proliferating worldwide, and, recently, distributed multi-camera surveillance systems have gained popularity. Typical surveillance systems start with the

C. Poppe · G. Martens · P. De Potter · R. Van de Walle
Department of Electronics and Information Systems - Multimedia Lab
Ghent University - IBBT
Gaston Crommenlaan 8 Bus 201, B-9050 Ledeberg-Ghent, Belgium
Tel.: +3293314959
Fax: +3293314896
E-mail: chris.poppe@ugent.be

G. Martens
E-mail: gaetan.martens@ugent.be

P. De Potter
E-mail: pieterjan.depotter@ugent.be

R. Van de Walle
E-mail: rik.vandewalle@ugent.be

detection and segmentation of objects of interest in images captured by each camera. The output of such object detection systems are pixel-wise segmentations of the image in foreground and background regions. Additional information that can be extracted from the images are the object sizes, colors, speeds, etc. This information forms the input for high-level analysis modules to make intelligent decisions on objects, classes, trajectories, and behaviours. Such information is generally called metadata and it has applications in a broad range of domains within computer science. When using a distributed video surveillance system (VSS), an interchange format is needed to structure this metadata. However, with the increasing complexity of large-scale distributed surveillance systems it becomes more and more difficult to use the same format in all modules of the system.

Generally, when defining an interchange format, a metadata standard is used that determines the structure of the format. In the context of video surveillance, different metadata standards have been proposed using the Extensible Markup Language (XML) as underlying language. XML allows to structure data according to an XML schema (following the XML Schema language [1]). The latter defines terms and constructs to represent the metadata and states the structure of the metadata. In this paper, we will show indeed that a number of different approaches exist in expressing the metadata associated with a video surveillance system. However, there is not one global metadata standard that is generally accepted for video surveillance, and most likely such a standard will not be introduced in the near future. Consequently, different modules describe their information according to different metadata formats. As a result, combining different metadata schemes with each other seems to be the only solution to create interoperability between different modules and systems. However, the standards generally use different XML constructs to denote the same concept. As such, it can be hard to find similarities between annotations using these different standards.

An additional disadvantage of XML is that it does not allow to explicitly define the semantics of the concepts that are described. Traditional metadata standards present an XML schema to define the structure and fields that can be used, and supply a textual description of the meaning of the different concepts. As such, the metadata is machine-readable but the semantics of the metadata fields are not.

In this paper, we introduce the application of Semantic Web Technologies to deal with the above-mentioned problems. We propose to create ontologies for the different metadata formats that are used in a surveillance system. Additionally, we create a global ontology specific for video surveillance systems, called the VSS ontology. This ontology represents all relevant information and acts as a uniform interface for the end-operator. The metadata ontologies are linked to the VSS ontology using rules and mappings, resulting in a layered metadata model. This model is consequently integrated in a semantic VSS to show the benefits.

The next section lists related work that discusses metadata in surveillance systems. In Sect. 3.1 we discuss the interoperability issues that occur when trying to incorporate existing metadata schemes with each other. Accordingly, in Sect. 3.2 the layered approach is presented that builds upon and combines formal representations of existing metadata schemes. Sect. 3.3 elaborates on the ontologies that represent the metadata standards and Sect. 3.4 shows the mappings and rules between the different ontologies. Sect. 4 presents a surveillance system that is built around our model and discusses the used technologies. To show the benefits of our approach, Sect. 5.1 discusses the semantic reasoning by using rules. Additionally, a use case scenario is shown in Sect. 5.2 to illustrate our system and, finally, conclusions are drawn in Sect. 6.

2 Related Work

Black et al. presented a framework for event detection and video content analysis within a multi-camera surveillance system [2]. They made a data model suited to describe images, objects (and their motion), and semantic aspects. This data model represents each of these aspects in a different layer and was modeled as tables in a database. Metadata is generated based on the layers to combine all the information and to increase the efficiency of querying. The actual metadata format was not reported, but their future work suggested using a common metadata standard for cooperation with other surveillance systems.

As the previous example suggests, to make metadata practically usable for information exchange between two or more modules, a common machine-readable metadata format is needed. This format describes which metadata can be used to describe the information of interest and how the metadata is structured. When using a common metadata format, software tools for automated manipulation can be created. One popular format for metadata is XML, which allows to structure the information so that it is machine-readable.

This approach can already be found in existing video surveillance systems (e.g., the CANDELA project [3] that uses MPEG-7 [4] to describe the features). Regarding to the used metadata standard many options are available. Next, we elaborate on related work using different metadata formats to describe video surveillance related metadata.

Zerzour et al. presented the VIGILANT system and created a semantic model for content and event based indexing of surveillance video [5]. It consists of a data model described using constructs from the KL-ONE language (used to explicitly represent conceptual information as a structured inheritance network). However, this language is not widely used for describing video surveillance metadata and disturbs the integration with different systems.

The need for describing video analytics with a common metadata format also arises when considering the evaluation (and comparison) of different video surveillance systems. Young and Ferryman presented a dataset of video sequences for evaluation of different algorithms and defined a common XML-based format to describe the detection results [6]. It contains information on objects and trajectories. The use of the common format allows to automatically and objectively analyze the performance of different algorithms.

An XML-based Computer Vision Markup Language (CVML) was presented by List et al. [7]. Additionally, they offered a free software library called CoreLibrary that assists people in handling the language. It has been used to describe hand-labelled ground truth datasets as part of the CAVIAR project [1].

Annesley et al. gave an interesting overview of the usage of MPEG-7 for video surveillance in general [8]. They presented examples on how MPEG-7 descriptors can be used. Since MPEG-7 is a large (and complex) metadata standard, they proposed a video surveillance specific profile to limit the amount of descriptors that need to be supported. Additionally, they created a Visual Surveillance XML schema (VS7) that uses some of the MPEG-7 descriptors and contains new types.

Recent efforts create languages that allow to explicitly define semantic information. An example of the latter, within the context of video surveillance systems, is the Video Event Representation Language (VERL), suggested by Nevatia et al. [9, 10]. It is used to

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

describe events and relations in video sequences using an ontology. Additionally, a Video Event Markup Language (VEML) was created that allows to annotate instances of the events described in VERL. Initially, VEML was a proprietary language constructed in XML, but in the final version, the base format used is the Web Ontology Language (OWL)[11], designed by the W3C Web Ontology Working Group. However, Nevatia et al. reported problems for describing the entire VERL ontology with OWL, so not all constructs are available as OWL instances. Integration with the MPEG-7 standard was proposed as future work but no information was given on how this could be done.

As this overview shows, most related work focusses on the usage of one single metadata format within a VSS. However, with the growing size and modularity of these VSSs, the possibility to include and work with different metadata formats is a prerequisite for the creation of a practically useful VSS. Different metadata standards exist and instances of these metadata schemes are everywhere. Therefore, to work with these formats, we need to address the interoperability between the different standards and cannot rely on one metadata standard alone (like MPEG-7). In the next section, we give more details about these interoperability problems and present a semantic layered metadata model tailored for use in a VSS.

3 Video Surveillance Metadata

3.1 Interoperability Issues

When trying to match the XML schemas of different standards we face interoperability problems. These problems were already signalled by the W3C Multimedia Semantics Incubator Group in which the authors of this paper have actively participated [12]. Although each of the standardized formats introduces interoperability amongst applications that use that standardized metadata scheme, issues occur when using different metadata schemes together. Listing 1 shows an XML fragment that describes the event of a detected person using CVML constructs [7]. The orientation and position are denoted and high level information about the specific action of the person is described. Similarly, Listing 2 shows an XML description of a detected person using the Visual Surveillance XML schema (VS7)[8]. This fragment holds metadata on the camera, the captured video sequence and includes temporal and spatial information on a bounding box that represents the detected person.

These examples illustrate the issues of interoperability created when using multiple metadata standards. The same concepts are described but in a totally different format. There are mismatches in the names of the XML elements, the structure, and the semantics. Mapping such XML fragments on each other is obviously a cumbersome task. The usage of eXtensible Stylesheet Language Transformations (XSLT) stylesheets [13], which were specifically created to transform XML instances, cannot always encompass the differences between different metadata standards. Additionally, when using XML to describe metadata, it is hard to describe semantic aspects. XML was mainly created to structure information and in many cases a metadata standard consists of an XML schema to denote the structure and the metadata fields that can be used, and a complementary textual description of the actual semantics of the metadata fields. Note that even when using one single standard (e.g., MPEG-7) to describe a resource, issues in interoperability can exist due to a lack of precise semantics [14]. As these examples show, using XML Schema is not sufficient. Consequently, we propose the use

```
<frame number="50">
   <objectlist>
      <object id="0">
         <orientation>148</orientation>
         <box xc="77" yc="73" w="21" h="16"/>
         <appearance>visible</appearance>
         <hypothesislist>
            <hypothesis id="1" prev="1.0" evaluation="1.0">
               <movement evaluation="1.0">
                   walking
               </movement>
               <role evaluation="1.0">walker</role>
               <context evaluation="1.0">walking</context>
               <situation evaluation="1.0">
                   moving
               </situation>
            </hypothesis>
         </hypothesislist>
      </object>
   </objectlist>
</frame>
```

Listing 1 Example of CVML metadata fragment describing a moving person.

of Semantic Web Technologies to deal with these issues by creating a layered semantic metadata model, discussed in the next section.

3.2 Semantic Metadata Model

Semantic Web Technologies allow to alleviate the interoperability issues within one metadata standard. For example, efforts have been undertaken to translate MPEG-7 into an OWL ontology and to enable its integration with other ontologies through appropriate frameworks, thus enhancing interoperability [14–17]. In the same way it is possible to create ontologies for each metadata standard that is used for video surveillance. These ontologies, called metadata ontologies, allow to structure the data and incorporate the semantic meaning of and relations between the different elements of the metadata standard. Such metadata ontologies form the first part of our metadata model.

To solve interoperability issues inherent to the use of several different metadata schemes, the ideal scenario would be to create one commonly accepted (metadata) ontology that encompasses all relevant concepts and that would be used in every module of the VSS. However, this is not feasible in practice as can be seen by the plethora of existing metadata standards. Different standards are used, whether they are small and simple (CVML) or broad and complex (MPEG-7). Conceptually, these metadata formats are on the same level, i.e. they all describe content. Consequently, we regard each metadata format as equally important and will handle the ontologies representing them as such.

As a second part of our model, we create an ontology specific for usage within a VSS that encompasses both system- and analytics-related metadata. System-centric metadata includes technical information on the captured images or video sequences, which is generally less interesting for the end-operator. Secondly, the analytics-centric metadata describes the actual content, meaning what is happening in the captured video. This can include concepts to describe detected objects (like persons and vehicles).

```
<VS7:VS7 xmlns:VS7="xsdVS7" xmlns:mp7="urn:mpeg:mpeg7:schema:2001"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="xsdVS7 vs7.xsd">
  <DescriptionMetadata>
     <mp7:Comment>
        <mp7:FreeTextAnnotation>
           A Visual Surveillance Schema (VS7) document
        </mp7:FreeTextAnnotation>
     </mp7:Comment>
  </DescriptionMetadata>
  <Media id="A">
     <MediaInstance>
        <mp7:InstanceIdentifier>
           Camera1
        </mp7:InstanceIdentifier>
        <mp7:MediaLocator>
           <mp7:MediaUri>
              file:/K:/camera1.avi
           </mp7:MediaUri>
        </mp7:MediaLocator>
     </MediaInstance>
  </Media>
  <LLID id="LLID1">
     <TemporalMask>
        <mp7:SubInterval>
           <mp7:MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
               mediaTimeBase="../../../Media[0]">
              1058
           </mp7:MediaRelIncrTimePoint>
           <mp7:MediaIncrDuration>1</mp7:MediaIncrDuration>
        </mp7:SubInterval>
     </TemporalMask>
     <Mask>
        <BB mp7:dim="4">187 162 282 409</BB>
        <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
           <mp7:Coeff>
              -202 59 27 42 5 11 19 14 6 13 11 22 6 11 16 7
           </mp7:Coeff>
        </ScalableColor>
     </Mask>
  </LLID>
</VS7>
```

Listing 2 Example of VS7 metadata fragment describing a detected object.

The VSS ontology is shown in Fig. 1, as can be seen, only a limited number of classes are created. In video surveillance, a bounding box is typically used to denote the location of a detected object or event. In the VSS ontology this is represented by the *BoundingBox* class which is a subclass of the more generic *Segment* class. The box (or segment) has properties to denote the coordinates of the lower left corner, width and height. Additionally, temporal information like the exact time and sample time (frame number) of the occurence of the segment can be stored. Lastly, it holds a reference to an actual image that contains this segment. Note that the latter is system-related metadata and will in most cases not be available in the used metadata standards (CVML has no way to define this for example). The properties that are internal to the classes are modeled as *DatatypeProperties* in OWL. To relate a bounding box with an object we introduce the *represents* property which is modeled as an *ObjectProperty*.

Note that the current ontology is kept very simple. However, since it is created with basic OWL constructs, it can easily be extended in the future. Possible extensions include the encapsulation of technical information about the video sequences, cameras,

viewpoints, more elaborate descriptions of detected events and trajects, and so on. This is future work, the focus of this paper lies not on ontology engineering but on using ontologies to overcome typical issues inherent to XML-based metadata in video surveillance.

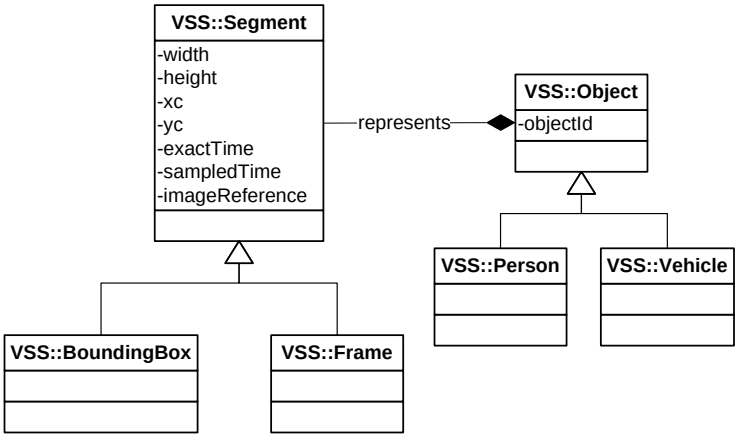

Fig. 1 Schematic overview of the VSS ontology.

To combine the VSS ontology with the metadata ontologies, we create a layered metadata model that combines the different metadata ontologies. Different metadata ontologies (representing the different metadata standards) constitute the lower layer and are linked to the VSS ontology, which acts as an upper layer. As such, we create a hierarchical system of two layers. The upper layer contains the VSS ontology with concepts suited for video surveillance systems. The lower layer exists of several metadata ontologies which can be used by different modules within a VSS.

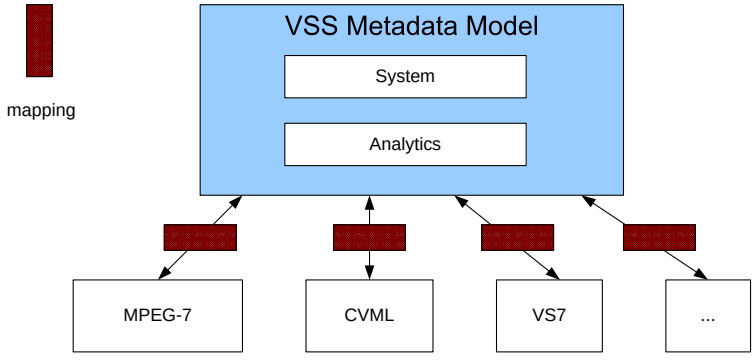

Fig. 2 Layered metadata model; the arrows denote mappings between the ontologies.

The layered metadata model is shown in Fig. 2 and consists of the created VSS model and the underlying metadata standards. Between the different ontologies there are so called mappings (represented as arrows in the figure) which consist of mapping ontologies and inference rules. These define the relations between classes, properties, and instances of involved ontologies and are discussed in Sect. 3.4.

The way we organize the different metadata schemes allows intelligent reasoning on different levels. The combination of the formal representation of different metadata standards in the lower layer allows broadening the search space when looking for content based on certain fields of a specific metadata scheme. The upper ontology allows the application of the semantic knowledge to make intelligent decisions on system level (e.g., make thumbnails of image regions with detected persons if the size is larger than a certain value).

3.3 Lower layer

The layered approach allows including new and existing ontologies. To show the extensibility of the system, we create a metadata ontology based on the CVML format. The XML-based CVML language is entirely defined by a textual specification (no XML schema has been created), however, a free software library has been made available that assists people in handling the language. Although the language defines the structure of the actual metadata, it cannot sufficiently describe the underlying semantics. When semantic reasoning is the goal, a formal representation of this metadata schema is needed.

For the development of the formal representation we cannot use an automatic conversion like Garcia et al. [16], since no XML schema is available for CVML. Consequently, we manually created classes and properties, which allow us to define semantic relationships that more closely resemble the actual meaning of the different fields. The final ontology is a compact OWL ontology with about 10 classes [2].

3.4 Mapping and rules

A mapping ontology typically consists of basic OWL and RDFS [18] constructs (e.g., *owl:equivalentClass* and *rdfs:subPropertyOf*) between concepts of different ontologies. Listing 3 shows an excerpt of a mapping ontology between the CVML ontology and the VSS ontology. The mapping ontology links properties of the different ontologies to eachother through the *rdfs:subPropertyOf* constructs (lines 6 and 10). The Listing also shows how the standard OWL constructs are used to map a CVML *Box* class on the conceptually equivalent VSS *BoundingBox* class using *owl:equivalentClass* (line 15).

Note that, for practical implementations, a mapping ontology as presented above is not sufficient. Rules are needed to create advanced conditional relationships, for example to declare instance equivalence when certain properties match, or to calculate new values for certain elements. Within a VSS the actual instances of the data (e.g., information on a specific image, or a detected person) are not known beforehand. Hence, we cannot define relations on them in the pre-determined mapping ontologies. However, by defining rules, new instances can automatically be linked to those that

[2] The CVML ontology can be found online at
http://multimedialab.elis.ugent.be/users/chpoppe/ontologies/surveillance/CVML.owl

```
<rdf:Description rdf:about="./CVML.owl#Object">
    <owl:equivalentClass rdf:resource="./Surveillance.owl#Object"/>
</rdf:Description>

<rdf:Description rdf:about="./CVML.owl#frame">
   <rdfs:subPropertyOf rdf:resource="./Surveillance.owl#sampledTime"/>
</rdf:Description>

<rdf:Description rdf:about="./CVML.owl#describes">
   <rdfs:subPropertyOf rdf:resource="./Surveillance.owl#represents"/>
</rdf:Description>

<rdf:Description rdf:about="./Surveillance.owl#BoundingBox">
   <owl:equivalentClass>
     <rdf:Description rdf:about="./surveillance/CVML.owl#Box"/>
   </owl:equivalentClass>
</rdf:Description>
```

Listing 3 Example of mapping using OWL constructs within the CVML to VSS mapping (the namespaces were abbreviated for layout purposes).

are stored within the system. As such, a dynamic mapping is created since the rules are triggered when new (instance) data becomes available.

Listing 4 shows such a rule to calculate the values of certain properties (we adopt the informal notation declared in SWRL (Semantic Web Rule Language) to give a human readable form of the rules [19]). In CVML a bounding box is described by the coordinates of the centre (xc and yc), the *width*, and the *height*. The VSS ontology also uses properties xc and yc, but these represent the coordinates of the lower left corner, so the CVML values need to be converted. The rule first looks for an instance in CVML that has values for the centre, width and height (this is an instance of the CVML *Box* class and is stored in variable $o1$). Next it calculates the coordinates of the lower left corner. Finally, the new properties in the VSS namespace are added to this instance. Note that the mapping (as defined above in Listing 3) states that an instance of the class *Box* in the CVML ontology is also an instance of the class *BoundingBox* in the VSS ontology. So this instance indeed can get the properties xc, yc, *width* and *height* which are defined in the VSS ontology.

```
@prefix vss: <http://multimedialab.elis.ugent.be/users/chpoppe/ontologies
     /surveillance/Surveillance.owl#>.
@prefix cvml:<http://multimedialab.elis.ugent.be/users/chpoppe/ontologies
     /surveillance/CVML.owl#>.

[r1: (?o1 cvml:xc ?xc1) (?o1 cvml:yc ?yc1)
      (?o1 cvml:width ?width1) (?o1 cvml:height ?height1)
      quotient(?height1 2 ?halveHeight) quotient(?width1 2 ?halveWidth)
      difference(?yc1 ?halveHeight ?ycorner)
      difference(?xc1 ?halveWidth ?xcorner)
      -> (?o1 vss:xc ?xcorner) (?o1 vss:yc ?ycorner)
         (?o1 vss:width ?width1) (?o1 vsss:height ?height1)]
```

Listing 4 Rule for mapping CVML properties of a *Box* instance to VSS properties of *BoundingBox* instances.

Next to the calculation or conversion of actual values of elements in the ontologies, rules are also needed to relate certain constructs in different ontologies. For example, to denote in the MPEG-7 ontology that a bounding box represents a person, several properties are needed. MPEG-7 is a multimedia metadata standard targeted for different domains. As such a bounding box, or a region in an image can be used to represent different things. If one wants to use MPEG-7 for video surveillance, more specific to denote that an object is detected, the semantics of the bounding box need to be described, resulting in additional properties. In contrast, only one property is needed to describe this in the VSS ontology. In OWL it is not possible to state that one property is equal to a cascade of other properties, so a rule is needed to convert such information. Such a rule first looks for the appearance of a combination of properties according to one ontology and then creates new properties in the VSS ontology.

The rule, shown in Listing 5, searches for an instance (stored in variable *segment*) that is linked through the *mpeg7:regionLocator* property to another instance (which is consequently stored in variable *box*). If this segment is related to an instance of the MPEG-7 *Person* class (stored in variable *o1*) through the *mpeg7:semantics* and *mpeg7:agent* object properties, the rule infers that *box* is the subject and *o1* is the object of the property *represents* from the VSS ontology (line 5).

```
1  @prefix vss: <http://multimedialab.elis.ugent.be/users/chpoppe/ontologies
       /surveillance/Surveillance.owl#>.
   @prefix mpeg7: <http://multimedialab.elis.ugent.be/users/chpoppe/
       ontologies/surveillance/mpeg7.owl#>.

   [r1: (?segment mpeg7:regionLocator ?box)
5       (?segment mpeg7:semantics ?sem) (?sem mpeg7:agent ?o1)
        (?o1 rdf:type mpeg7:Person)
        -> (?box vss:represents ?o1) ]
```

Listing 5 Rule to relate a cascade of properties in the MPEG-7 ontology to one specific property in the VSS ontology.

In the next section, we present a semantic VSS that uses the layered semantic metadata model proposed in this section. First, we briefly elaborate on the analysis modules that are used in our system. Next, we show how the metadata in RDF format can be obtained. Lastly, we show how the semantic metadata model is integrated in the system.

4 Semantic Video Surveillance System

A conceptual overview of our semantic VSS is shown in Fig. 3. In this paper we restrict ourselves to surveillance systems with only one camera to monitor the scene. Since the beginnings of digital video surveillance, compression is used to reduce the bandwidth and storage costs. An encoder is used to remove the redundancy in and between the video frames and a compressed video stream or bit stream is created. The analysis of the video stream can occur on a decoded version, or directly on the compressed video stream. In the next section, we will elaborate on these two analysis methods. The results of the analysis modules are represented as (XML-based) metadata and sent through a webservice to the semantic unit. The XML-based metadata is first

converted to RDF, explained in Sect. 4.2. The RDF triples are then processed and linked to the VSS ontology in the RDF triple store within the semantic unit. Sect. 4.3 will discuss this unit and the used technologies. Lastly, end-users can communicate with a second webservice which offers query access to the metadata.

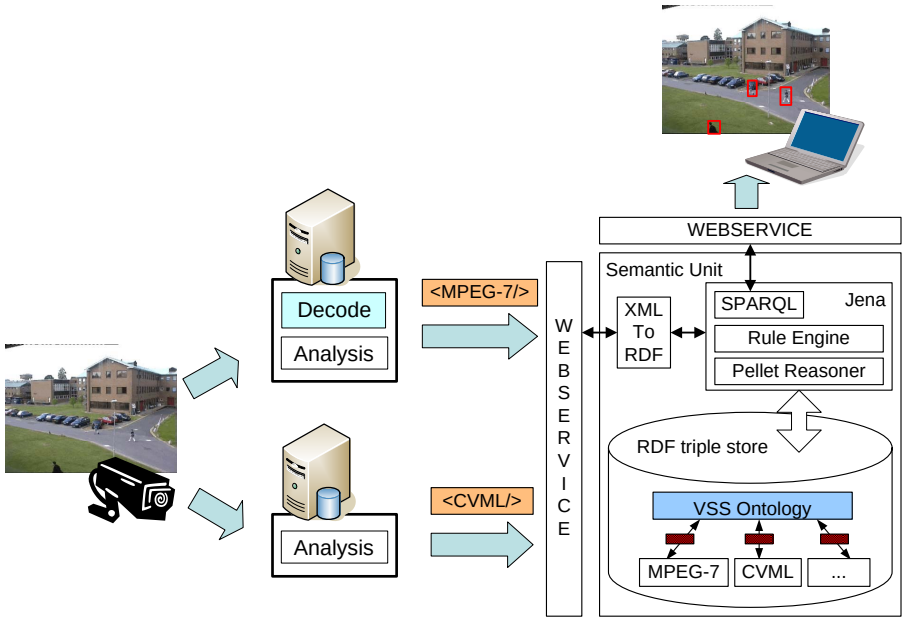

Fig. 3 Architecture of the Semantic Video Surveillance System.

4.1 Video Analytics

Within an intelligent VSS, one or more video analytics modules are used that analyze the captured images to retrieve relevant information in an automated manner. When looking at video analytics systems, we distinguish two main approaches. A first one analyzes the original video sequences on a pixel-level. In a second approach, the analysis happens in the compressed domain, meaning that the sequences are first encoded with a video codec and the compressed bit stream is analyzed.

In most cases a first form of processing is applied on the camera, including contrast enhancement, noise reduction, etc. In case of a smart camera the processing also includes video analytics, like motion detection. This processing can be used to reduce the amount of data that is sent from the camera. Such cameras will, for example, only produce video streams if a certain amount of motion is detected in the scene. Since the analysis on the camera can occur on the original captured sequences, we can apply typical pixel-based video analysis methods. A pixel-based moving object detection technique, presented in our previous work [20], is integrated in our VSS. This technique

detects those pixels that are likely to correspond with moving objects (like people and vehicles) in the sequence and represents the objects using MPEG-7 metadata.

Besides the embedded analysis that occurs on the camera, surveillance systems can have independent analysis modules that analyze the video sequences after compression. Consequently, if one wants to analyze the captured images, a decoding step is needed before pixel-based algorithms can be applied. To avoid the decoding step and to reuse the work done during the encoding, the literature holds several efforts to perform the analytics directly upon the compressed video stream. In this case, the compressed video stream is analyzed and the specific coding constructs that are available in the stream are the main information sources. Since the compressed video is a more compact representation of the original video stream, analytics working in the compressed domain can be faster than the pixel-domain approaches. Moreover, it is not necessary to fully decode the video stream before the analysis can be done, resulting in additional gains in time.

For the analysis of the compressed video, we use a moving object detection technique working on H.264/AVC-compressed video sequences. More information on this detection technique can be found in [21]. Since this module works in the compressed domain, no color or texture information is available, so there is no need to use a detailled and advanced metadata standard like MPEG-7. Consequently, CVML is used to represent the bounding boxes of the objects that are detected in the compressed streams.

To summarize, in our implementation we analyze the same video sequence with two distinct algorithms. The algorithms are implemented in C++ and detect moving objects in video surveillance sequences. Information on these objects (e.g., bounding boxes for spatial location) are represented using XML-based metadata standards (MPEG-7 for the pixel-based approach and CVML for the compressed-domain approach).

The next section discusses how the generated XML instances can be converted to RDF instances which can be linked with the semantic data model.

4.2 XML to RDF Conversion

We have chosen the generic XML to RDF conversion presented by Van Deursen et al. [22]. This uses an XML document as mapping document that defines specific mapping rules between an XML instance document and the resulting RDF document. A generic $XMLtoRDF$ tool takes this mapping document and the used ontology as input and then automatically transforms corresponding XML documents to RDF instances, as shown in Fig. 4.

The XML-based mapping document is built from specific elements which form a mapping language and can be interpreted by the $XMLtoRDF$ tool. This language allows to create a simple mapping of XML nodes to corresponding OWL classes or properties. Conditional mappings are available in case a mapping not always holds. In that case, a condition can be made of XPATH (XML Path Language) or SPARQL ASK expressions. Finally, value processing is included which specifies different ways to infer the value of a resulting OWL property. These specific language constructs ease the development of such XML to RDF mappings.

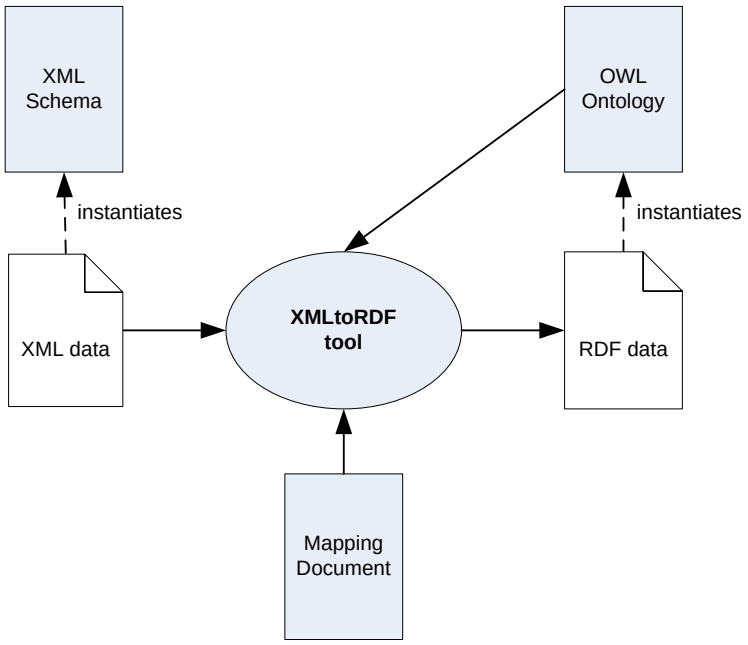

Fig. 4 Working of the $XMLtoRDF$ tool.

4.3 Semantic Unit

The different analysis modules create metadata for each frame that is analyzed. For instance, when a moving object is detected, a bounding box is created and represented in the used metadata standard. In the surveillance system, the same moving object can be detected by different analysis modules, so different metadata representations can exist. The detected bounding boxes can differ in size and position, depending on the used analysis algorithm, but the object that they describe is conceptually the same. If an operator wants to retrieve images with a specific moving object, the detected objects should be regarded as one object. So the information expressed in the different metadata standards need to be linked to each other. Hence, we can use the layered semantic metadata model presented in Sect. 3.2 to perform the appropriate mappings.

For this purpose, a semantic service is created that uses Jena [3] as underlying platform. The VSS ontology and the ontologies that represent the metadata formats are imported in the Jena platform at start-up of the system. For the reasoning we use Pellet [4] and rules are described in SWRL and interpreted by the Jena platform. When the analysis modules send new metadata to the web service, these are added to the RDF triple store in the Jena platform and, if appropriate, rules are triggered. All metadata is now present in the RDF triple store so standard approaches can be used for querying this information. For this purpose SPARQL is used to perform the queries on the metadata [23].

[3] http://www.jena.sourceforge.net
[4] http://pellet.owldl.com

Finally, a web interface is provided to the end-operator that offers specific methods for querying the VSS. Internally these methods use SPARQL queries which are resolved by the Jena framework. The results of these queries are then interpreted by the web service and presented to the end-operator in a suitable way.

Note that the querying occurs with respect to the VSS ontology, and not to the metadata ontologies. This prevents that the SPARQL queries have to incorporate knowledge about each metadata format that is used. Since all metadata ontologies in the system are linked to the VSS ontology, we can retrieve information by only querying the latter. As such, when a new analysis module is incorporated in the system, using a different metadata standard, it is enough to provide the mappings to the VSS ontology to make it fully integrated with the semantic metadata model.

5 Evaluation

The way our VSS is built gives some advantages compared to traditional approaches. Firstly, since we represent the metadata (including the ontologies and mappings) in OWL, we can make use of the existing Semantic Web Technologies to perform reasoning or to create rules. Some practical examples are given in the next section. Secondly, our system allows to combine metadata formatted according to different metadata standards. In Sect. 5.2, a practical use case scenario is given to show how the proposed system deals with different metadata formats.

5.1 Semantic reasoning

As discussed before, a moving object can be detected by several analysis modules, possibly from different vendors. It is not feasible that all analysis modules have knowledge on the detected objects in other analysis modules. Hence, each module separately creates metadata for the detected object and both CVML and MPEG-7 representations of the object are entered in the RDF triple store. Following the rules and mappings, shown in Sect. 3.4, this metadata is automatically mapped upon the VSS ontology, creating instances of the VSS *Object* class. However, there should only be one instance of this class to represent the detected object.

Since the metadata is represented using Semantic Web Technologies, we can take advantage of the standard rule support to deduce that different detected objects are conceptually the same. The rule shown in Listing 6 states that if two bounding boxes in a frame largely overlap, they represent the same object. As a result, the instances of the VSS *Object* class are now considered to be equal but represented by several bounding boxes.

By using the same principle, we can introduce a tracking system that finds identical objects in consecutive frames. An example of a very simple tracker can be found in Listing 7. This rule states that if two bounding boxes largely overlap in consecutive frames, they denote the same object.

In traditional surveillance systems, an analysis module usually does both the detection and tracking of moving objects. Information on the trajectories of moving objects can be interesting to an end-operator, so, accordingly, some metadata standards have provided constructs to represent these. For example, in CVML the tracking is represented by giving a unique identifier to an object. This way, all objects with the same

```
@prefix vss: <http://multimedialab.elis.ugent.be/users/chpoppe/ontologies
     /surveillance/Surveillance.owl#>.

[r1: (?o1 rdf:type vss:Object)(?o2 rdf:type vss:Object) notEqual(?o1 ?o2)
     (?x1 vss:represents ?o1) (?x2 vss:represents ?o2)  notEqual(?x1,?x2)
     (?x1 vss:xc ?xc1) (?x1 vss:yc ?yc1)
     (?x2 vss:xc ?xc2) (?x2 vss:yc ?yc2)
     (?x1 vss:width ?width1) (?x2 vss:width ?width2)
     (?x1 vss:height ?height1) (?x2 vss:height ?height2)
     le(?xc1 ?xc2) le(?yc1 ?yc2)
     sum(?xc1 ?width1 ?sumw1)  le(?xc2 ?sumw1)
     sum(?yc1 ?height1 ?sumh1) le(?yc2 ?sumh1)
     -> (?o1 owl:sameAs ?o2)]
```

Listing 6 Rule to find identical objects in one frame.

```
@prefix vss: <http://multimedialab.elis.ugent.be/users/chpoppe/ontologies
     /surveillance/Surveillance.owl#>.

[r1: (?o1 rdf:type vss:Object) (?o2 rdf:type vss:Object) notEqual(?o1 ?o2
     )
     (?x1 vss:represents ?o1) (?x2 vss:represents ?o2)   notEqual(?x1,?x2
          )
     (?x1 vss:xc ?xc1) (?x1 vss:yc ?yc1)
     (?x2 vss:xc ?xc2) (?x2 vss:yc ?yc2)
     (?x1 vss:width ?width1) (?x2 vss:width ?width2)
     (?x1 vss:height ?height1) (?x2 vss:height ?height2)
     le(?xc1 ?xc2) le(?yc1 ?yc2)
     sum(?xc1 ?width1 ?sumw1) le(?xc2 ?sumw1)
     sum(?yc1 ?height1 ?sumh1) le(?yc2 ?sumh1)
     (?x1 vss:sampledTime ?t1) (?x2 vss:sampledTime ?t2)
     max(?t1 ?t2 ?max) min(?t1 ?t2 ?min) difference(?max ?min ?diff)
     lessThan(?diff 3)
     -> (?o1 owl:sameAs ?o2) ]
```

Listing 7 Rule to find identical objects over several frames.

identifier are conceptually the same, so the bounding boxes in the different frames can be found. In the CVML ontology, we have chosen to represent the identifier as a property that is an *owl:inverseFunctionalProperty*. This is a standard construct in OWL, stating that if two instances have the same value for this property, they are considered to be equal. This way, if two instances of the CVML *Object* class have the same identifier, they are automatically set to be equal by the reasoner. So, regardless whether the tracking occurs by the analysis module (in software) or by the reasoning engine (through rules), the result is that only one instance of the VSS *Object* class will represent the tracked object.

5.2 Use Case Scenario

This section presents a walk-through of a specific use case scenario in which an end-operator wants to see all images with a moving object.

The first analysis module uses MPEG-7 to describe the detected objects. An example of such a metadata instance is shown in Listing 8. As mentioned before, MPEG-7 is a complex metadata standard that is used in different domains. Therefore, the

Semantic element and *SpatialDecomposition* element need to be combined to denote that a bounding box represents an object (in this case a person).

When this MPEG-7 XML-based annotation is uploaded to the system, RDF triples are created that correspond to the MPEG-7 ontology using the $XMLtoRDF$ tool (we use the ontology of Garcia et al.[16]).

The second analysis module uses CVML to describe the moving objects. Note that the specification of CVML explicitely defines a bounding box to represent a moving object. As such, it does not need additional constructs like the MPEG-7 example to denote that the box corresponds to a detected person. Hence, the XML annotation in CVML is much simpler (already shown in Listing 1).

Within the metadata service, this XML annotation is converted to RDF triples, again by the $XMLtoRDF$ tool, and stored for future retrieval.

Finally, our VSS offers the end-operator the possibility to search for images containing moving objects through a web interface. When this search is requested, the web service constructs a SPARQL query solely based on VSS metadata as shown in Listing 10. This query searches for image references (stored in variable Z) that are linked to segments which represent an object. As shown, the query only uses concepts of the VSS ontology to retrieve the desired images.

Since the semantic representations of our VSS metadata model and the underlying MPEG-7 and CVML standards are linked together, through the ontology mapping and rules as explained in Sect. 3.2, the system retrieves references to all images that contain a moving object (or more precisely, that contain a bounding box assumed to represent a moving object). Using the references, the actual pictures can be retrieved and shown to the operator. The web service can also retrieve the coordinates of the box that represents the object and draw the box on the shown image for better interpretation.

A traditional XML-based VSS would have to create queries that interpret all the XML metadata formats. So for each format, a specific query has to be made that follows the structure of the standard. In our case, only one query is needed since all information is linked together through the use of the semantic metadata model.

6 Conclusions

The main contribution of this paper is the introduction of Semantic Web Technologies for the creation of a layered metadata model to augment the capacities of video surveillance systems. The layered metadata model has been created to deal with current interoperability problems induced by the application of different metadata formats in the various modules of current video surveillance systems. An upper layer consists of an ontology specifically created for video surveillance systems and includes technical and analytics metadata. This ontology is linked to a lower layer containing a pool of metadata ontologies, commonly used in surveillance. We introduced the application of Semantic Web Technologies consisting of mapping ontologies and inference rules to integrate the different ontologies in the layered metadata model. Additionally, we presented a video surveillance system that integrates this metadata model. To show the advantages of our approach, an object tracking system has been created with rules inherent to the Semantic Web Technologies. Lastly, we have shown that the system can deal with the information management of multiple analytics modules each using different metadata standards.

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2004" xmlns:xsi="http://www.w3.org
    /2001/XMLSchema-instance" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
    xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 Mpeg7-v2.xsd">
  <!-- Describes the Media Segments -->
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="SegmentCollectionType">

        <!-- a segment of video -->
        <Segment xsi:type="VideoSegmentType">
        <!-- Assigns IDs and shows the hierarchy -->
          <Semantic>
          <Label>
            <Name>A detected Person</Name>
          </Label>
          <MediaOccurrence>
            <MediaInformationRef xpath="SpatialDecomposition[0]"/>
          </MediaOccurrence>
          <SemanticBase xsi:type="AgentObjectType" id="AO1">
            <AbstractionLevel dimension="1"/>
            <Label href="urn:ipcorp:obj:move:person ">
              <Name>Person1</Name>
            </Label>
            <MediaOccurrence>
              <MediaInformationRef xpath="SpatialTemporalDecomposition[1]
                  "/>
            </MediaOccurrence>
          </SemanticBase>
        </Semantic>

        <!-- a description of a moving region within a video -->
        <SpatialDecomposition>
          <MovingRegion id="Segment1">
            <!-- Region location -->
            <SpatioTemporalLocator>
              <ParameterTrajectory motionModel="still">
                <!-- Time point  -->
                <MediaTime>
                  <MediaTimePoint>2006-12-04T14:09:17</MediaTimePoint>
                </MediaTime>
                <InitialRegion>
                  <Box xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" mpeg7:dim
                      ="2 2">
                    179 109 53 178
                                                </Box>
                </InitialRegion>
              </ParameterTrajectory>
            </SpatioTemporalLocator>
            <!-- Colour description -->
            <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="
                Average">
              <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="6">
                <Coeff>
                    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                  </Coeff>
              </ScalableColor>
            </VisualDescriptor>
          </MovingRegion>
        </SpatialDecomposition>
        </Segment>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Listing 8 Example of metadata expressed in MPEG-7 format.

```
<?xml version="1.0" encoding="UTF-8"?>
    <frame number="36">
        <objectlist>
            <object id="0">
                <orientation>143</orientation>
                <box xc="78" yc="64" w="19" h="13"/>
                <appearance>visible</appearance>
                <hypothesislist>
                    <hypothesis id="1" prev="1.0" evaluation="1.0">
                        <movement evaluation="1.0">walking</movement>
                        <role evaluation="1.0">walker</role>
                        <context evaluation="1.0">walking</context>
                        <situation evaluation="1.0">moving</situation>
                    </hypothesis>
                 </hypothesislist>
            </object>
        </objectlist>
        <grouplist/>
    </frame>
```

Listing 9 Example of CVML metadata in XML format. It expresses that a person has been detected in frame 36.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vss: <http://multimedialab.elis.ugent.be/users/chpoppe/ontologies/
     surveillance/Surveillance.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?Z
WHERE
{ ?X rdf:type vss:Object.
  ?Y vss:represents ?X.
  ?Y vss:imageReference ?Z.
}
ORDER BY ?Z
```

Listing 10 Example of SPARQL query to retrieve.

Currently, the VSS ontology contains a restricted set of concepts for video surveillance, but it can easily be extended since it is created in OWL. As such, in future work, other relevant aspects can be added like the usage of different cameras and viewpoints, means to describe behavior or contextual information like prohibited areas and so on.

Acknowledgements The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

1. D.C. Fallside and P. Walmsley. XML Schema part 0: Primer (second edition). W3C Recommendation, W3C, October 2004.
2. J. Black, D. Makris, and T. Ellis. Hierarchical Database for a Multi-Camera Surveillance System. *Pattern Analysis and Applications*, pages 430–446, 2005.

3. R.G.J. Wijnhoven, E.G.T. Jaspersand, and P.H.N. de With. Flexible surveillance system architecture for prototyping video content analysis algorithms. In *Proc. SPIE Electronic Imaging*, volume 6073, 2006.
4. MPEG-7 Overview, International Organization for Standardisation, Klagenfurt ISO/IEC JTC1/SC29/WG11, July 2002.
5. K. Zerzour, G. Frazier, and F. Marir. VIGILANT: A Semantic Model for Content and Event Based Indexing and Retrieval of Surveillance Video. In *Proceedings of International Workshop onKnowledge Representation meets Databases*, pages 143–154, 2000.
6. D.P. Young and J.M. Ferryman. PETS Metrics: On-Line Performance Evaluation Service. In *Proceedings of Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 317– 324, 2005.
7. T. List and R.B. Fisher. CVML - An XML-Based Computer Vision Markup Language. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 789–792, 2004.
8. J. Annesley, A. Colombo, J. Orwell, and S. Velastin. A Profile of MPEG-7 for Visual Surveillance. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 482–487, 2007.
9. R. Nevatia, J. Hobbs, and R. Bolles. An Ontology for Video Event Representation. In *Proceedings of Computer Vision and Pattern Recognition Workshop*, pages 119–129, 2004.
10. A.R.J. Francois, R.Nevatia, J. Hobbs, and R. Bolles. VERL: An Ontology Framework for Representing and Annotating Video Events. *IEEE Multimedia*, pages 76–78, 2005.
11. P. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, 10 february 2004. Available on http://www.w3.org/TR/2004/REC-owl-semantics-20040210/.
12. W3C Multimedia Semantics Incubator Group.
Available on http://www.w3.org/2005/Incubator/mmsem/.
13. J. Clark. XSL Transformations: XSLT (Version 1.0). W3C Recommendation, W3C, November 1999.
14. R. Troncy, W. Bailer, M. Hausenblas, P. Hofmair, and R. Schlatte. Enabling Multimedia Metadata Interoperability by Defining Formal Semantics of MPEG-7 Profiles. In *Lecture Notes in Computer Science*, volume 4306, pages 41–55, 2006.
15. J. Hunter. Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In *Proceedings of the First Semantic Web Working Symposium (SWWS)*, pages 261–281, 2001.
16. R. Garcia and O. Celma. Semantic Integration and Retrieval of Multimedia Metadata. In *Proc. 5th Knowledge Markup and Semantic Annotation Workshop*, pages 69–80, 2006.
17. R. Arndt, R. Troncy, S. Staab, and M. Vacura L. Hardman. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *Lecture Notes of Computer Science: The Semantic Web*, volume 4825, pages 30–43, 2007.
18. F. Manola and E. Miller. RDF Primer. W3C Recommendation, W3C, February 2004.
19. I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language - Combining OWL and RuleML, W3C Member Submission, 21 May 2004. 2004. Available on http://www.w3.org/Submission/SWRL/.
20. C. Poppe, S. De Bruyne, G. Martens, P. Lambert, and R. Van de Walle. Intelligent Preprocessing for Fast Moving Object Detection. In *Proceedings of SPIE Security and Defense*, volume 6978, page 69780S, 2008.
21. C. Poppe, S. De Bruyne, T. Paridaens, P. Lambert, and R. Van de Walle. Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. *Visual Communication and Image Representation*, 20(6):428–437, 2009.
22. D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. XML to RDF conversion: a Generic Approach. In *Proceedings the 4th International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, 2008.
23. SPARQL Query Language for RDF, W3C Recommendation 15 January 2008. Available on http://www.w3.org/TR/rdf-sparql-query/.