SAM - An Interoperable Metadata Model for Multimodal Surveillance Applications

Peter Schallauer[1], Werner Bailer, Albert Hofmann, Roland Mörzinger
JOANNEUM RESEARCH Forschungsgesellschaft mbH, Steyrergasse 17, A-8010 Graz, AUSTRIA

ABSTRACT

Metadata interoperability is crucial for various kinds of surveillance applications and systems, e.g. metadata mining in multi-sensor environments, metadata exchange in networked camera systems or information fusion in multi-sensor and multi-detector environments. Different metadata formats have been proposed to foster metadata interoperability, but they show significant limitations. ViPER, CVML and MPEG Visual Surveillance MAF support only the visual modality, CVML's frame based approach leads to inefficient representation, and MPEG-7's comprehensiveness handicaps its efficient usage for a specific application. To overcome these limitations we propose the Surveillance Application Metadata (SAM) model, capable of describing online and offline analysis results as a set of time lines containing events. A set of sensors, detectors, recorded media items and object instances is described centrally and linked from the event descriptions. The time lines can be related to a subset of sensors and detectors for any modality and different levels of abstraction. Hierarchical classification schemes are used for many purposes, such as types of properties and their values, event types, object classes, coordinate systems etc. in order to allow for application specific adaptations without modifying the data model while ensuring the controlled use of terms. The model supports efficient representation of dense spatio-temporal information such as object trajectories. SAM is not bound to a specific serialization but can be mapped to different existing formats within the limitations evoked by the target format. SAM specifications and examples have been made available.

Keywords: Metadata, interoperability, surveillance, multi-modal, event, object, sensor, description.

1. INTRODUCTION

Classical surveillance systems are mainly based on human attention for detecting critical events directly from video streams. There is a limited need for exchanging metadata in such a scenario. The larger modern surveillance systems get and the more automatization of analysis, fusion and mining of critical events across sensors and networks is applied; the more metadata needs to be exchanged between components and users in such systems. Metadata interoperability becomes a crucial question for leveraging the full capabilities. The aim of this work is to create a metadata model that can be used to represent metadata produced in a wide range of surveillance applications. Section 2 provides an overview of requirements for a metadata model capable of dealing with a wide range of applications. In recent years different metadata formats have been utilized in the surveillance domain. Section 3 presents a review of the metadata formats ViPER, CVML, MPEG-7 and MPEG Visual Surveillance MAF with respect to our formulated application requirements. They show significant limitations: Either they have been developed for metadata interoperability within a different field of application, for representing metadata of a single modality or provide inefficient metadata modeling and representation. To overcome these limitations we propose in section 4 the Surveillance Application Metadata (SAM) model with its design criteria, its model classes and relations and a description of how to deal with controlled vocabularies. Section 5 describes how the SAM model can be represented in a persistent metadata format, how an API implementation has been done and how SAM has been used in two example applications.

[1] peter.schallauer@joanneum.at; phone +43 316 876 1202; fax +43 316 876 1191; http://www.joanneum.at/iis

2. REQUIREMENTS

We intend to define a model that can be used to represent the metadata produced in a wide range of surveillance applications. These applications differ in a number of aspects that imply certain requirements for the metadata model. In the following, we discuss these requirements in detail.

2.1 Description of events and their properties

The central concept in surveillance applications is the *event*. This concept is usually interpreted quite broadly and goes beyond the colloquial use of the word. We use the term event to denote any spatiotemporal event in the scene that is of interest in a certain application, such as objects/persons entering/leaving, performing any kind of interaction among them or with the scene or any behavior they might show. The description of such events and their properties – spatio-temporal extent, identification of objects/persons involved and event-specific properties – is thus the main requirement for the metadata model.

2.2 Time-based description

These events could be ordered by any of their properties, e.g. by object. However the most general and important dimension is time. It thus makes sense to require a time-based description of events. This should consist of a single or several timelines in order to support multiple sensors, detectors, abstractions etc. as discussed below.

2.3 Distributed systems

Surveillance systems are increasingly distributed and analysis results have to be exchanged across components of the system. Fragments of the description may be synchronously created in different components. Thus the metadata model needs to provide means for identification of entities across descriptions or description fragments and support for both local and system-wide frames of reference, such as coordinate systems.

2.4 Online and offline applications

In the applications of interest both offline and online scenarios exist. Online scenarios are characterized by the need to incrementally fill the data model with small units of information, to efficiently access information in the current time window and to efficiently serialize fragments of the description. In offline applications handling larger descriptions, that are well structured and contain little redundancy, is the main issue.

2.5 Flexible in terms of modalities and sensor types

Advanced surveillance systems use a large number of sensors capturing various modalities. The metadata model has to provide means of dealing with the specific properties of each of these modalities as well as providing an abstraction layer to describe integrated detection and analysis results across the different modalities.

2.6 Abstraction levels

The information created by the processing of the captured sensor data exists on different abstraction levels: some information is quite low-level and tightly related to the sensor type, while the application of detection and analysis algorithms creates increasingly higher-level and abstract information. For example, from the captured video the change of a region can be detected, which can be described as an event of an object entering an area the scene. The region can then be identified as a human, and analysis of the trajectory will indicate the walking direction. Integrating knowledge about the scene and from other sensors an event of a person entering a prohibited area can be inferred.

2.7 Description granularity

The needs for the granularity (e.g. temporal density) vary depending on the application and the type of information. While some information might be required on a very detailed level for later analysis, other information can be represented quite coarsely. This requirement might also be different within different components of a system, e.g. an online analysis component might require very detailed information for a certain time range, while for later processing only a coarser representation of the metadata is kept.

2.8 Efficient representation of spatio-temporally sampled data

In many applications there are some data with a very dense granularity, such as trajectories of objects or persons. The data model has to provide efficient means for representing such data.

2.9 Support for controlled vocabularies

The consistency of metadata values, especially those that are exchanged between systems or entered/modified by users has to be ensured. Following experience from the library, archive and production communities this can be done by controlled vocabularies. The use of controlled vocabularies for identifying types of events and entities, properties etc. can also help to keep the metadata model more general and thus easier to adapt to new applications.

2.10 Representation suitable as internal data model of application for interchange between components

Especially in an online scenario fragments of metadata are typically exchanged between software components of the system. The metadata model thus needs to be represented as object model.

2.11 Support for different serializations

For persistent storage as well as for exchange between systems serialization of the data model is necessary. Depending on the application, for example efficient binary storage or a human readable format might be required. Certain applications require serialization conforming to a certain standard, which might not be able to capture the functionality of the data model.

3. SURVEILLANCE METADATA FORMATS REVIEW

The majority of surveillance systems still use proprietary metadata representations. However the industry and research community have proposed a number of formats and standards that can be used for representing the metadata of surveillance applications. In the following we review these formats in the light of the requirements defined above.

3.1 ViPER

The Video Performance Evaluation Resource (ViPER) [2] is a framework for the evaluation of video analysis tools. The ViPER XML format [8] has been defined based using XML Schema as an exchange format for manually produced ground truth and results from automatic analysis tools. A ViPER XML description consists of two main parts: configuration information (such as types of events, information about the setup and capture) and the actual data. The data is organized by source files. A list of events is described for each source file, with the time span, spatial attributes and additional properties such as detection confidence for each of the events.

The format is sufficiently flexible to support media types other than video; however, the time representation is frame oriented which makes it difficult to combine sensor data sampled at different rates. The format is source file oriented; properties of sensor and detectors have to be associated with a source file or a set of them. The concept of source file is flexible and in a recent version a ‘virtual’ file can be defined as a sequence of separate files. Descriptions on different abstraction layers are difficult due to the file orientation and would require some workarounds with application defined semantics. The format is highly flexible for defining new types of objects, with a set of attributes (using the basic ViPER data types) and having static and dynamic values for these attributes. Such definitions can be made in any description instance. Thus the semantics and units of the attribute values are not formally specified. Object instances are not defined in advance; however, all occurrences of an instance in the current video can be collected in one element XML. Identifying object instances across several documents requires application logic. In the current set of basic data types only image but no world coordinates are supported, which is a problem for multi-sensor systems. Trajectories of arbitrary shapes can be modeled; however a matrix type does not exist. Extensions of the basic data types are possible, but will not be compatible with existing ViPER based tools.

3.2 CVML

The CVML language (Computer Vision Markup Language) was presented at IPCR 2004 [5]. The language defines a common data interface specifically designed for computer vision. Its focus is on content description of video and image sequence data. The structure of CVML allows only the definition of one dataset or sequence per XML file. CVML does not differentiate between sensors and dataset and it does not facilitate the description of (physical) media locations. Further the concept of non-visual (e.g. audio) features is not supported at all.

Although CVML comes with open-source cross-platform libraries, such as a C++ based CoreLibrary[2] which supports reading and writing the information in XML, the documentation of the large amount of available language tags is weak and incomplete. CVML is described in more detail at mindmakers.org[3].

3.3 MPEG-7

MPEG-7 [7] is a comprehensive standard for the description of multimedia content, including structuring the content as well as describing a number of low-, mid- and high-level features for each of the segments in the structure. Due to the flexibility of the spatial, temporal and spatio-temporal structuring capabilities it is well suited for the description of events in different kinds of audiovisual modalities.

The description approach of MPEG-7 is content-centric so that multiple streams and non-continuous recordings can be easily handled. However the standard has no support for data from non-audiovisual sensors. The descriptions can be organized along the time lines of the content streams; however this time is designed as a media time and not as a real-world time. As the media time of a recorded sensor data may be different than the real-world time it corresponds to a workaround for such issues would be needed. Similar problems occur with spatial description in real-world coordinates, they can be given for a content segment, but the description of e.g. object trajectories is not possible in world-coordinate system that integrates the view of several video streams. MPEG-7 has a very generic way of describing tools used for media creation, so that the semantics of specific information (e.g. camera calibration) are not captured. MPEG-7 has a very comprehensive support for controlled vocabularies called Classification Schemes that could be easily adopted for a surveillance description.

3.4 MPEG Visual Surveillance Application Format

MPEG has started working on Multimedia Application Formats (MAF) [4], which provide the framework for integration of elements from several MPEG standards into a single specification that is suitable for specific but widely usable applications. Typically, MAFs specify how to combine metadata with timed media information for a presentation in a well-defined format that facilitates interchange, management, editing, and presentation of the media.

One of the MAFs under development is the Video Surveillance Application Format (VSAF) [1]. This MAF includes MPEG-4 AVC for video coding, the AVC file format, the ISO Base Media File Format supporting global and time synchronized metadata and subsets of MPEG-7 part 5 (Multimedia Description Schemes) and part 3 (Visual). The focus in the current draft is on the data encoding and wrapping functionality, while we are interested in the MPEG-7 based metadata fragments that can be attached to the container or to individual tracks (each representing video from one camera).

The VSAF has a focus on visual data thus only description of the visual modality is supported. The video stream may be decomposed temporally into segments or spatially into regions and grids. For a segment, identifiers, time, tool settings as well as comments and text annotation can be annotated; the latter can make use of classification schemes. Objects related to events can be referenced by IDs. Two types of visual descriptors can be used for a segment.

A drawback is that the format requires separate MPEG-7 documents for global and clip level information, as it is assumed that related files are packed together into an MAF file. Sensors can only be specified with the clip, if there exists more than one clip captured with a sensor the sensor description has to be repeated. Some of the MPEG-7 description elements are used in a way that does not follow the intended semantics of the MPEG-7 standard.

3.5 Conclusion

This review shows that the existing formats are based on quite different requirements. Some of the formats' focus is outside the surveillance domain (e.g. MPEG-7) while others are specifically designed for a specific use case (such as CVML). We can get relevant input for the metadata model design from each of these formats, however, there is no single format covering all the requirements defined above. Also the models of some of these formats are strongly influenced by the serialization method (e.g. XML) supported by this format.

[2] http://corelibrary.sourceforge.net/
[3] http://www.mindmakers.org/projects/cvml

4. SAM MODEL

4.1 Design criteria

We intend to define a metadata model that fulfils the requirements listed in Section 2 and is compatible to modeling approaches of the existing formats discussed in Section 3 as far as possible. The basic design decisions for the metadata model are thus the following.

Decouple model and serialization. As we have found that no existing format fulfils our requirements and given the need to support different serializations, the model design and the serialization method are decoupled. This avoids constraints on the model imposed by limitations of certain persistent formats.

Multiple timelines of events. Considering several requirements and the representation used in MPEG-7 and ViPER a time-based description of events is the main structure of the model. Taking the requirements for multimodal and multi-sensor systems as well as several abstraction levels into account several such timelines can exist. They can be linked by time and the reference to sensors.

Lightweight and flexible. The metadata model should be lightweight and easy to understand. At the same time flexibility for a wide range of surveillance applications is needed, which seems to contradict the lightweightness requirements. However, analysis of the requirements shows that the basic structures are suitable across all target applications, while the variability is in the details of types of events, objects, sensors etc. supported. This can be solved by including the basic types and structures into the metadata model while specific types and properties are created by qualifiers coming from controlled vocabularies (cf. Section 4.3).

4.2 SAM classes and their relations

The classes of the SAM model and their relations are shown in Figure 1. The main class of the model is *SAM* which directly links to the core concepts *Sensor*, *SensorData*, *Detector*, *Object* and *Description*, where each may occur multiple times.

A *Sensor* may be a hardware component (e.g. video camera, microphone) or a virtual sensor like an accumulation of other physical sensors. A *Sensor* is described by its type and unique id, and may optionally specify a list of *Locations*, *CoordinateSystems* and additional *Properties*. The data captured by one or a group of sensors (e.g. videos, audio files) can be described with the *SensorData* concept. The *RecordingInformation* of the *SensorData* describes the temporal and spatial metadata of the recorded data.

A *Location* is described by one or multiple specific coordinates, or by one or multiple *Spatiotemporal Trajectories*. A trajectory specifies a list of coordinates, each having a specific time point assigned. The interval between the coordinates can either be fixed or variable, depending on the use case (e.g. analyze offline videos with a fixed frame rate of 25fps, or the use in an online detector that may drop frames in the analysis process due to variable computation time and external influences). Coordinates and *Trajectories* may explicitly refer to a *CoordinateSystem*, either one defined by a sensor (e.g. an image coordinate system of a video camera), or a common coordinate system referenced from a controlled vocabulary (e.g. a common Groundplane coordinate system).

A *Detector* is a software or hardware component extracting information from sensor data. It is described by a type and unique id and may optionally have additional *Properties*.

The *Object* class describes a specific object instance (e.g. a specific vehicle or person), the occurrences of the object is modeled by an *Events*. An object is identified by its type and unique id and may specify additional *Properties*.

An *Event* describes an incident involving zero or more *Objects*. The confidence of the event can either be specified as single value for the whole event time range or as a *ConfidenceTrajectory* with different confidence values over time. Also part of the *Event* description are references to the involved *Sensors* and *Detectors*, the *Location* (e.g. the *Trajectory* of the object movement) and a list of *Properties*.

A *Description* represents a timeline of *Events*.

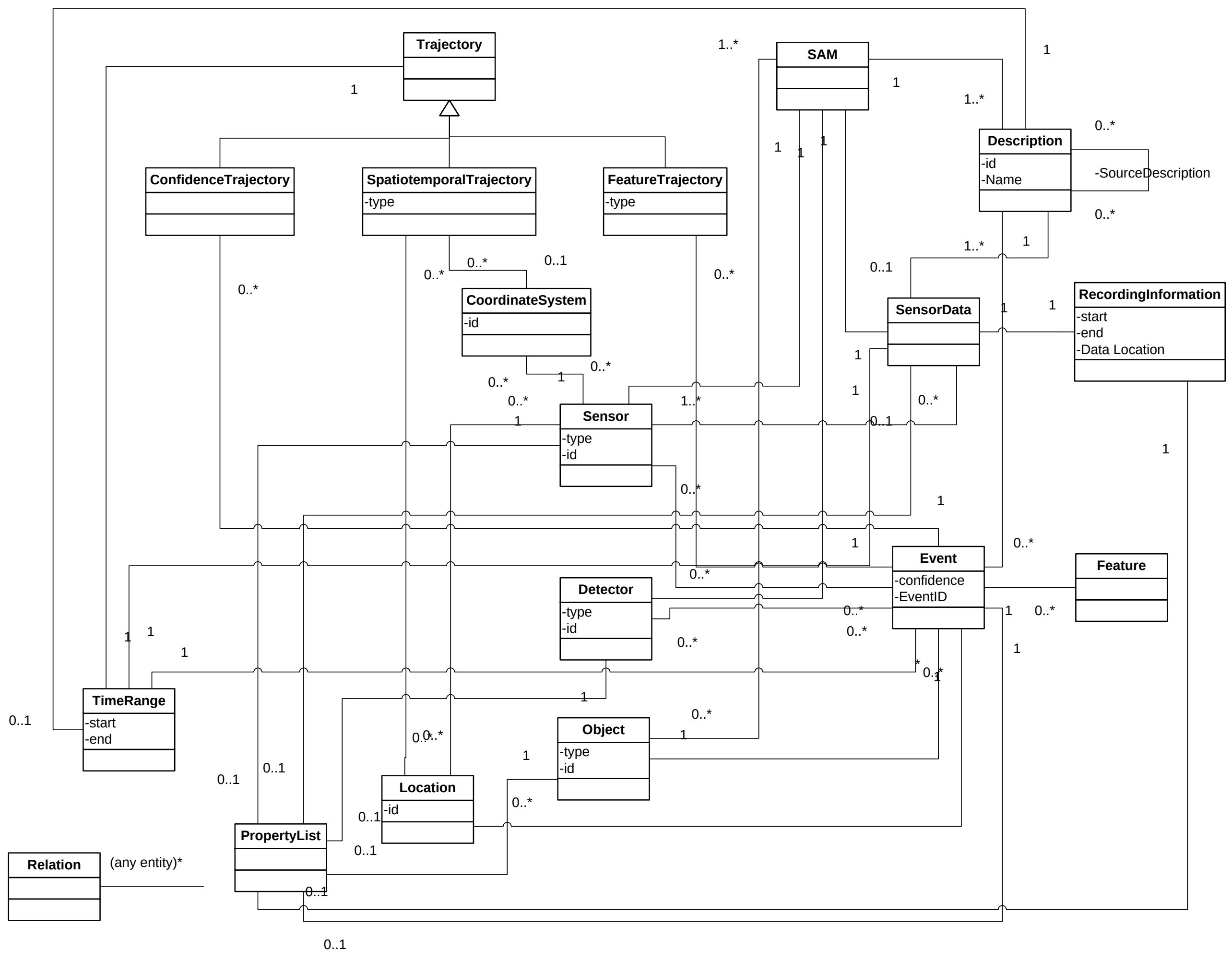


Figure 1: The SAM Model

4.3 Controlled use of terms (Classification Schemes)

As has been mentioned before controlled vocabularies have crucial functionalities in SAM. A controlled vocabulary is any set of controlled terms, with or without relations between them. This can be as simple as a list of values, a classification scheme, a thesaurus or ontology. The only requirement is that such a controlled vocabulary and any term in it can be referenced using a URI. SAM does not define or require the use of a specific format for the controlled vocabulary. For example, formats such as MPEG-7 Classification Schemes [7], RDF [6] or SKOS [3] can be used.

The use for controlled vocabularies in SAM is twofold: One is the rather classical application for controlling values of properties such as names of places, object instances, sets of values for tool settings etc. The other use is for qualifiers of general data types and keys in property lists. This adds flexibility to the model as no specific types of events, objects etc. have to be defined for a certain application but more general types can be qualified. Thus only an application specific controlled vocabulary has to be defined while basic tools for handling the data model can be reused.

5. USING SAM

5.1 Serialization

As outlined above we have decoupled the metadata model design from the serialization. However, in many cases serializations of metadata descriptions are necessary for exchange between systems. We have defined a native serialization of the data model called *SAM.F* (SAM Format). This format has been defined by representing the metadata model described in Section 4 using XML Schema and is capable of representing all aspects of the model so that a lossless exchange of descriptions is possible. The intended use of this format is mainly for development and for the serialization of intermediate results in systems. Different implementations of SAM.F are possible. Section 5.2 describes exemplarily a C++ implementation, other languages can quite easily be implemented, e.g. in order to generate Java bindings based on the SAM.F XML schema XMLBeans could be applied.

An example of a resulting XML document conforming to SAM.F is shown below. This description fragment contains an event describing the movement of a person detected in Camera2. The time range of the person movement is stated directly inside the Event node, as well as in the Trajectory node. If the event describes only one trajectory, the event time range and trajectory time range are equal. Date and Time values are serialized according to ISO 8601, optionally including the fractions after the second. If no UTC offset is used for date and time values, UTC will be assumed and not the current local time. The confidence value may range from 0.0 (absolutely not confident) to 1.0 (completely confident). In the example we describe the confidence for the whole event time range, if the confidence of detection differs over time, the ConfidenceTrajectory element can be used (which has the same form as the Trajectory element in the example, except that the Values element only contains pairs of the time and the respective confidence value). The Trajectory element describes the time related coordinates of the person on a common ground plane. The values of the coordSystem and type attribute are referenced from a classification scheme (in the form ClassificationSchemeName:value). The Values element contains the time related coordinates. In the example we use a variable interval, thus we specify the time value for each coordinate.

```xml
<SAM>
  <Description id=”Description1”>
    <Event id="Event1">
      <TimeRange>
        <Start>2008-09-19T16:19:35.5130000+01:00</Start>
        <End>2008-09-19T16:19:36.1930000+01:00</End>
      </TimeRange>
      <Confidence>1.000000</Confidence>
      <SensorRef>Camera2</SensorRef>
      <Location id="Event1Location">
        <Trajectory
             coordSystem="CoordinateSystemCS:CoordinateSystem.Cartesian.GroundplaneMidOfGroundline2D"
             dim="2" type="ShapeCS:Shape.Point2D">
          <TimeRange>
            <Start>2008-09-19T16:19:35.5130000+01:00</Start>
            <End>2008-09-19T16:19:36.1930000+01:00</End>
          </TimeRange>
          <Values>
            16:19:35.5130000 1.88442 26.6065
            16:19:35.5930000 1.88681 26.4618
            16:19:35.7130000 1.84043 26.4612
            16:19:36.1930000 1.88681 26.4618
          </Values>
        </Trajectory>
      </Location>
      <ObjectRef>Person1</ObjectRef>
    </Event>
  </Description>
</SAM>
```

For the exchange of descriptions the SAM data model can be mapped to any existing format or standard. Necessarily such a mapping will be lossy in some cases, depending on the appropriateness and expressivity of the target format. We have investigated mappings to the formats discussed in Section 3. The limitations (and resulting loss of information) encountered corresponds to the not fulfilled requirements of these formats. For some constructs workarounds can be found which require application specific interpretations of these elements. However, many applications will not have all the requirements considered by our data model. In such cases some of the existing formats may be sufficient to represent all metadata elements needed in these applications.

5.2 API

We have implemented a C++ library for SAM to allow for flexible integration into C++ applications. The architecture of the SAM package is shown in Figure 2.

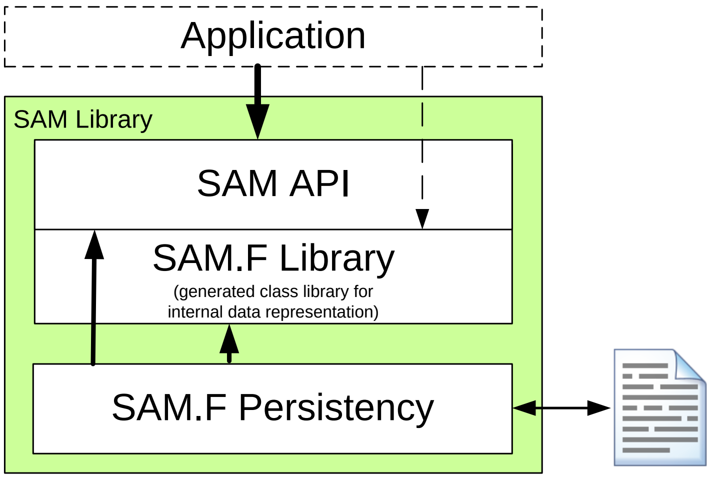


Figure 2: SAM package structure.

Based on the SAM.F XML schema, the SAM.F library was generated with the JRS API Generator (same technology as used for the JRS MPEG-7 Library, see http://mpeg7.joanneum.at). This SAM.F library serves as the basic internal data representation. Since the generated API is completely based on the SAM.F XML schema, classes for abstraction are provided (SAM API) that hides the SAM.F native XML specific representation of the original concepts of the metadata model.

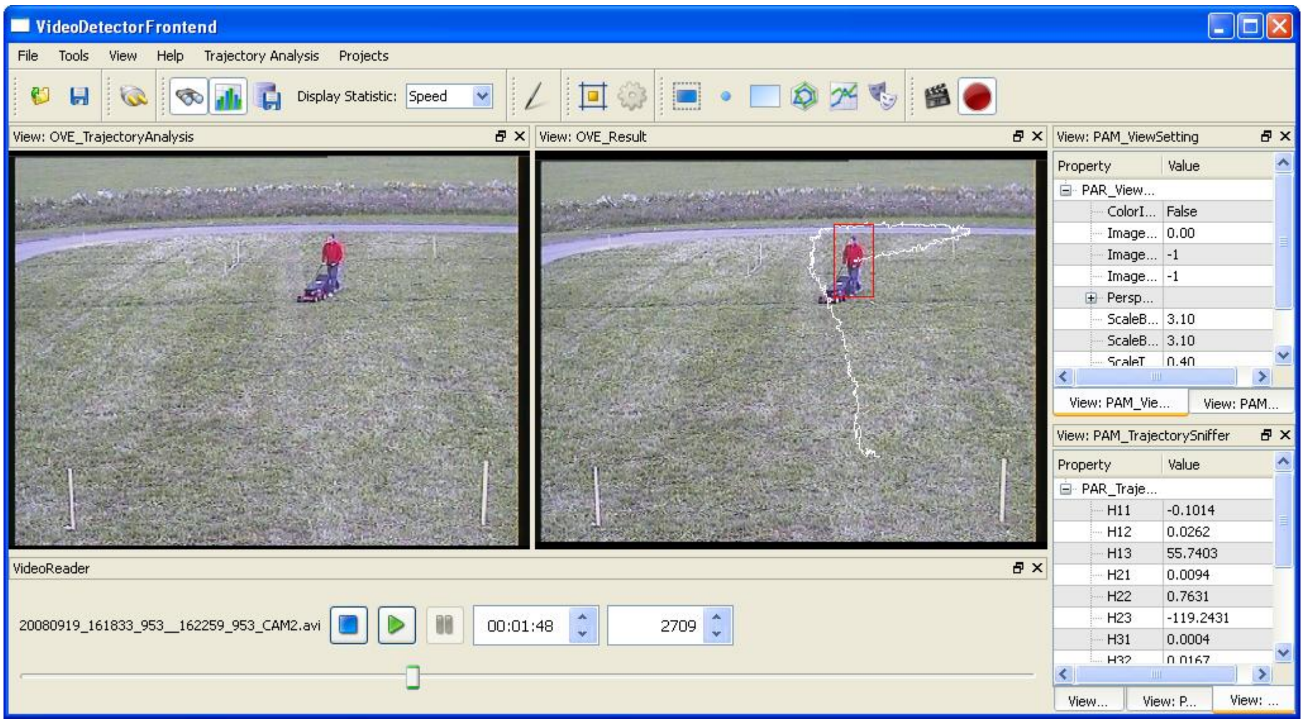
Persistency functionality is implemented via plug-ins. A persistency plug-in implements a slim interface for serializing and parsing a SAM document. The SAM.F persistency plug-in is built-in, so the SAM library is self-contained with serializing and parsing functionality. The library can handle multiple different persistency formats / plug-ins at the same time (e.g. in order to load a document from format A and store it to format B).

5.3 Applications

This section shows two examples of software applications where SAM has been utilized prototypically.

VideoDetectorFrontend

The VideoDetectorFrontend is an application for online analysis of a single video camera. We can plug-in various detectors and trackers in the analysis chain and visualize their results in separate views. In the example in Figure 3 we have included a person detector and a tracker. The results of the tracker are continuous trajectories of the detected persons. A separate plug-in receives notifications for the detected trajectories and stores these results in a SAM file. By using a homography we can also map the coordinates within the image into coordinates on a common ground plane and store these trajectories in the SAM file along with the image coordinates.



Property	Value
PAR_View...	
ColorI...	False
Image...	0.00
Image...	-1
Image...	-1
Persp...	
ScaleB...	3.10
ScaleB...	3.10
ScaleT	0.40

Property	Value
PAR_Traje...	
H11	-0.1014
H12	0.0262
H13	55.7403
H21	0.0094
H22	0.7631
H23	-119.2431
H31	0.0004
H32	0.0167

Figure 3: VideoDetectorFrontend user interface.

MultiView Player

The MultiView player is an application for offline multi-view visualization, navigation and appraisal of multi-modal content analysis results. The user interface as shown in Figure 4 is divided into three main parts: Timeline components for efficient navigation in the world timeline (an overview and just below an temporally zoomed timescale, also visualizing time ranges where videos are available for a certain camera), a video player area for time synchronized playback of visual and acoustic recordings for each camera or microphone, and a ground plane visualization with overlay functionality of the multimodal detection results. On the ground plane we can visualize object locations provided by an acoustic location detector using a microphone array sensor and the results from the visual person detection and tracking provided by the VideoDetectorFrontend application. Both detection results are provided to the MultiView Player in the SAM.F XML format.

For the data input in the MultiView player we have implemented a SAM FileLoader plug-in that extracts the information from multiple SAM files into the internal data representation of the tool.

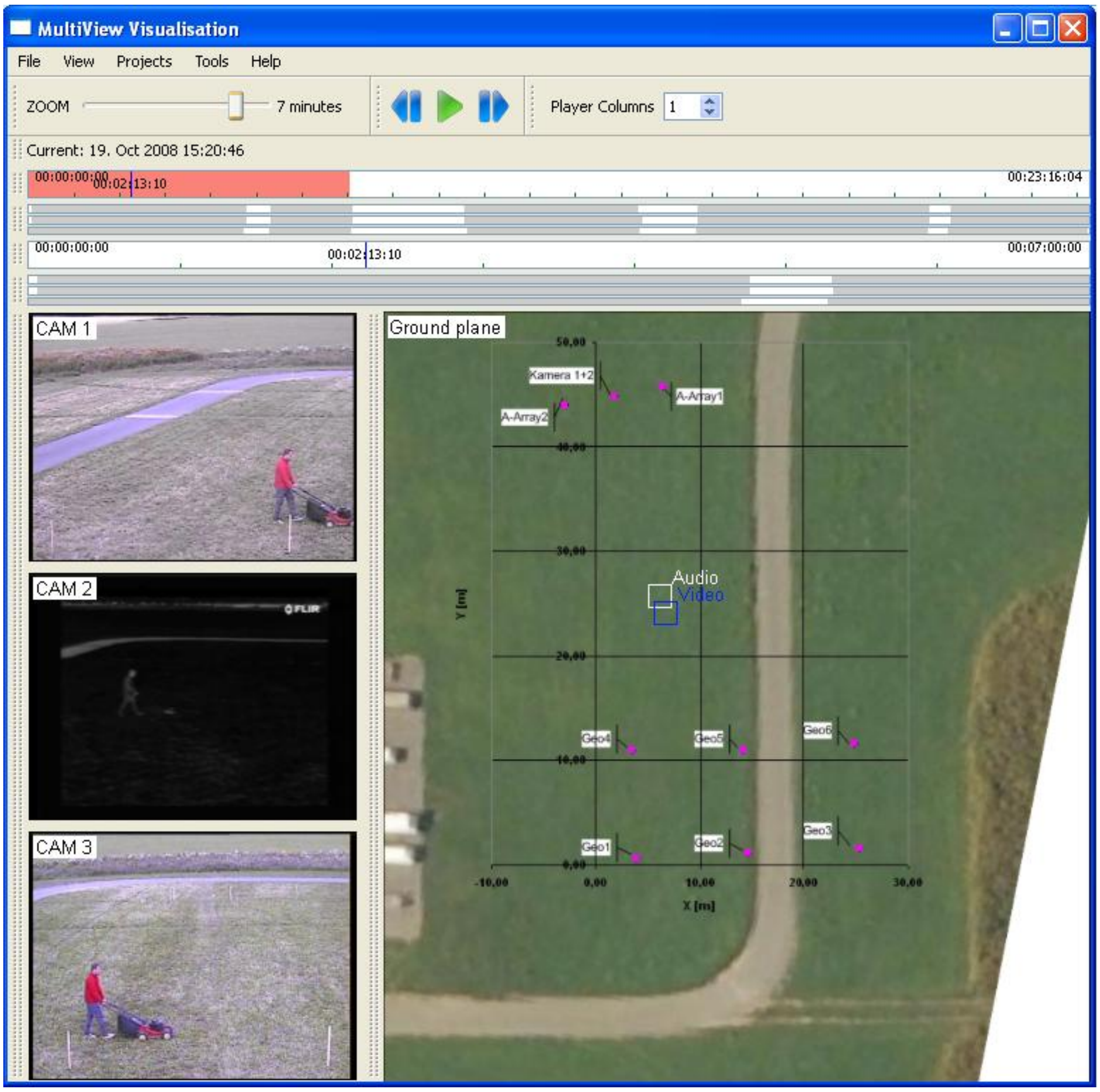


Figure 4: MultiView visualization user interface.

6. CONCLUSION

We have defined a metadata model for use in different types of surveillance applications. A review of existing metadata formats used in the surveillance domain shows that none of these formats fulfils all the requirements coming from this broad range of applications, mainly due to restrictions in terms of supported modalities, structure and serialization format. We have thus defined SAM as metadata model following the design principles (i) decoupling of model and serialization, (ii) support of multiple timelines of events and (iii) lightweightness and flexibility. The SAM model is based on a time-based event description, making heavy use of controlled vocabularies to allow for customization to application specific needs. There is a native XML based serialization called SAM.F, however, other serializations (e.g. binary) and (lossy) mappings to other metadata formats are possible.

SAM has been prototypically used in two applications and has proven to be suitable for different application scenarios. We have shown that SAM is not only limited to visual metadata description but also for other sensor types like acoustic sensors. Based on the SAM.F XML schema a C++ library has been created in order to allow for efficient application development. Bindings to other languages are easily possible by the same approach. Some additional SAM resources are available at sam.joanneum.at, e.g. the SAM-F XML schema and example file, several sample classification schemes and a set of explanatory slides.

7. ACKNOWLEDGEMENTS

The authors would like to thank Werner Haas, Georg Thallinger, Neuschmied Helmut, Herwig Zeiner, Maria Fellner, Franz Graf, Harald Rainer, Bernhard Rettenbacher as well as several other colleagues at JOANNEUM RESEARCH, who contributed valuable input to the work. This work has been funded partially under the 7th Framework Programme of the European Uni**o**n within the IST project "SCOVIS" (IST FP7 216465, http://www.scovis.eu) and the national research projects AREA MUMOSIS, iObserve, 3D-Inside and additional research activities funded by the Austrian Federal Ministry for Transport, Innovation and Technology.

REFERENCES

[1] Bäse G. and Rathgen, T. (eds.), “Text of ISO/IEC FCD 23000-10 Video surveillance application format,” ISO/IEC JTC1/SC29/WG11MPEG2008/N9706 (2008).

[2] Doermann, D. and Mihalcik, D., “Tools and techniques for video performance evaluation,” Proc. 15th Intl. Conference on Pattern Recognition, 4, 167-170 (2000).

[3] Isaac, A. and Summers, E., “SKOS Simple Knowledge Organization System Primer”, W3C Working Draft, URL: http://www.w3.org/TR/skos-primer (2008).

[4] Kim K., Schreiner F. and Diepold K. (eds.), “MAF Overview,” ISO/IEC JTC1/SC29/WG11 N10233 (2008).

[5] T. List and R. B. Fisher, “CVML—an XML-based computer vision markup language,” Proc. ICPR, vol. 1, pp. 789–792, Cambridge, UK, August 2004.

[6] Manola F. and Miller E., “RDF Primer”, W3C Recommendation, URL: http://www.w3.org/TR/rdf-primer (2004).

[7] MPEG-7 – Multimedia Content Description Interface, ISO/IEC 15938:2001.

[8] ViPER XML: A video description format. URL: http://viper-toolkit.sourceforge.net/docs/file/ (2003).