ViVA Project

Report

Ontology for video surveillance 1.0

M. C. Miguel Angel Palacios Alonso.

Dr. Pablo Hernández Leal.

Dr. Hugo Jair Escalante.

Dr. Enrique Sucar Succar

Instituto Nacional de Astrofísica, Óptica y electrónica

Content

List of Figures

List of tables

1. Introduction

Technological advances have facilitated the automation of several processes. In Mexico, a very important issue is related to security of public and private places. A closed-circuit camera system is the preferred preventive action. A camera system for surveillance can help to monitor different areas at the same time, so that, the area covered is larger and decreases the maintenance cost.

Currently IP cameras allow us to exploit the local enterprise network, which means, same wired infrastructure for data communication and internet access could be used for video video-surveillance.

However, although video surveillance systems have proved to be a very useful tool, in most of the cases they are used after that an event has happened. Preventive systems are preferred as they are critical for guaranteeing the safety of people and infrastructure. In order to provide to the ViVA project, preventive mechanisms, it is required to model all possible risk scenarios. Specifying potential threats and risk events.

Thus, in addition to vision algorithms for automatic event detection, the ViVA project proposes modeling the system based on an ontology. Ontologies are a formal way to specify knowledge of a particular domain. Specially, they allow us to define/describe all possible entities involved and their relationships. In this way, the knowledge can be exploited by the ViVA system to trigger alerts, whenever a potential threat is detected. This document describes the ViVA ontology and its most important entities for the video-surveillance domain.

1.1.1. Purpose of the document

This document describes the concepts and relations which are part of ViVA ontology. ViVA ontology has been designed to be used as a high-level layer in a video-surveillance system.

1.1.2. Scope

The ontology design process is considered to be as an iterative process. This characteristic is very important because the surveillance environments could be diverse (e.g., new events and entities can emerge with time). Thus, this document describes the 1.0 version of the ontology proposed.

2. Related work

ViVA Project proposes the development of an automated video surveillance system to assist a user (usually a security guard). The automated system should be able to understand what is happening in a video sequence being analyzed. Thus, in addition to the design and implementation of image processing and computer vision algorithms also a knowledge representation is required. All the low-level information produced by the algorithms should be organized in order to generate knowledge.

In the last years, ontologies have been used for representing contextual information [1]. Ontologies have expressiveness and reasoning capabilities that enable experts to understand the concepts used to describe the domain modeled. Flexibility, reasoning, information sharing, and knowledge representation make these models suitable to surveillance domains. A way to reduce the semantic gap in video processing (see Figure 1 Semantic gap between processing module and the semantic concept.Figure 1).

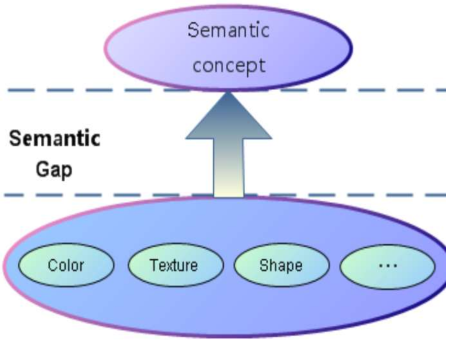

Figure 1 Semantic gap between processing module and the semantic concept.

An ontology for video event analysis was presented by [2]. The ontology is based on two levels of knowledge: application domain and the analysis system. The ontology and the case study is specialized for the Underground video surveillance domain. The framework designed by [2] is depicted in Figure 2. The knowledge is combined in a selection stage to determine the best visual analysis framework and system reactions to the analysis of the modeled domain. To test the ontology proposed events *GetObject, PutObject, AbandonedObject and StolenObject* were analized.

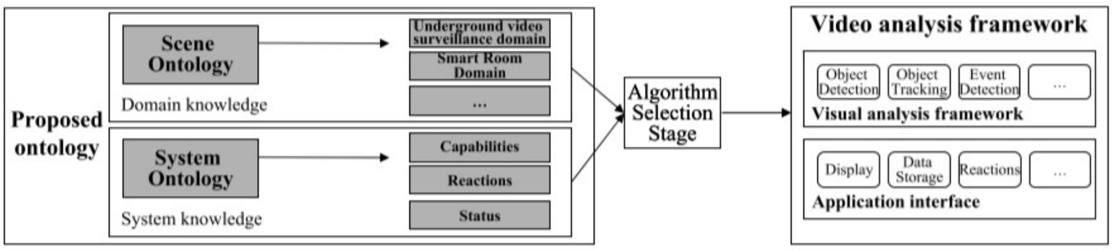

Figure 2 Overview of the proposed framework by [San Miguel et al, 2009]

In order to detect events and objects [3] proposes to use an ontology together with a rule detection system. The system uses probabilities to estimate certain events as well as its initial and final times. The system considers a manual segmentation of static elements (see Figure

3) to participate in the Performance Evaluation of Tracking and Surveillance (PETS) challenge [4].

Figure 3 Scene representation from PETS 2012 challenge camera view.

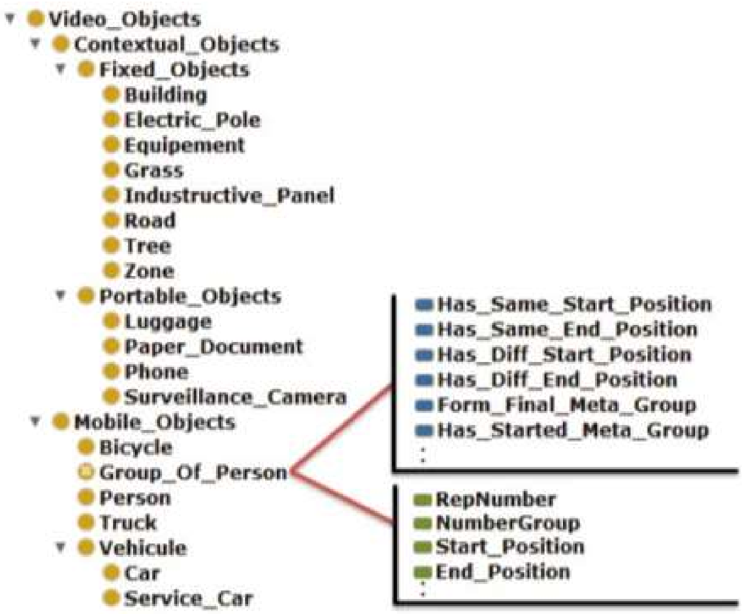

Figure 4 Video Objects class hierarchy sample from [3]

The representation is based on [2] and complete it by adding new concepts. The ontology is organized in four categories: Video Events, Video Objects, Video Sequences and Bounding boxes. Figure 4 shows a sample of the Video_Objects class hierarchy proposed. Protégé [5] is used for inference rule definitions in SWRL language.

In [6] a behavior ontology is proposed. The ontology is evaluated in the PETS [4] 2006 and 2007 challenge. This approach sets that a specific behavior can be acted in various ways but they still share a *general plot.* The Behavior model (general plot) is based in the scene concept as shown in Figure 5. An object model also is designed (see Figure 6) where low level data is specified.

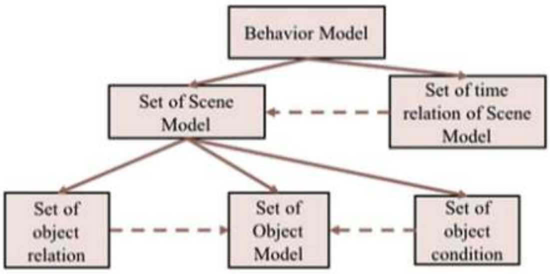

Figure 5 The modeling of a specific behavior in [6]

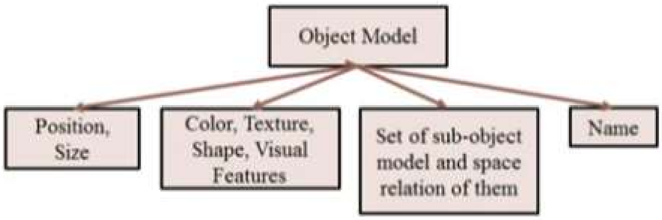

Figure 6 The modeling of an object in [6]

Three kinds of relation are defined: Space, Space-Time and Action. The Figure 7 shows an example of this kind of relations (in frame, near, his, far, out frame) in left luggage behavior.

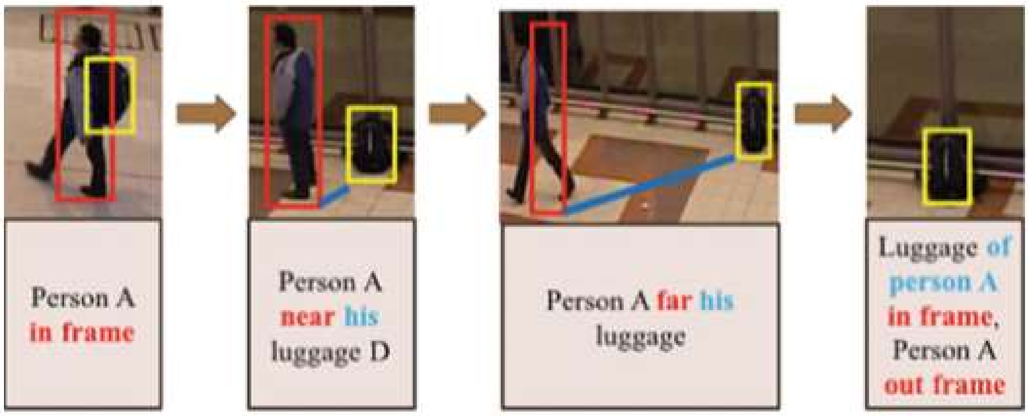

Figure 7 The example interested objects and their relations, conditions in left luggage behavior in [6]

Finally, Video Surveillance Online Repository (VISOR) [7] is a support tool for different projects, it contains a large set of multimedia data and the corresponding metadata annotations. Visor provides a list of video surveillance concepts used in the Visor system. The ViSOR project has defined several video surveillance concepts. A general overview of the hierarchy of concepts proposed by ViSOR is depicted in Figure 8. Concepts are organized in two main branches, Content and Context.

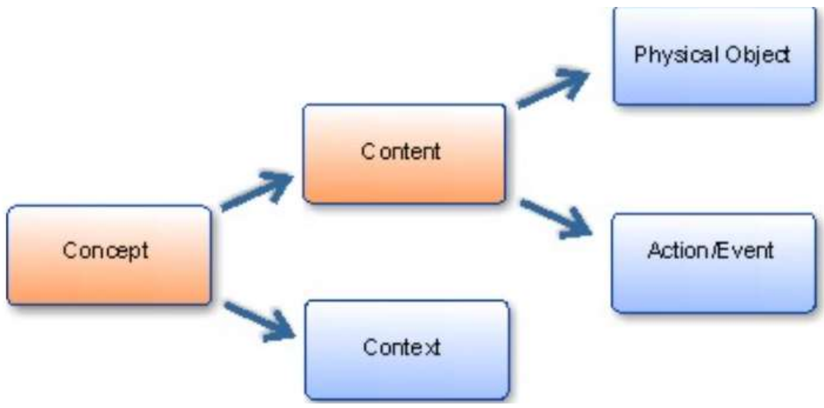

Figure 8 General overview of the surveillance concepts proposed by the ViSOR project

Ontologies are a knowledge representation technique that facilitates the reduction of semantic gaps between low level descriptors and the domain of interest. Reviewed literature shows that ontologies have been widely used and proved to be useful for the video-surveillance domain.

3. An ontology for video-surveillance

The ontology concept was introduced into the engineering field through the IA field where it was defined as the real-world representation in computer programs. Thus, an ontology could be defined as a way to organize related concepts in some domain [8]. The final objective is to share the knowledge captured by the ontology under a common specification.

3.1.1. ViVA ontology

ViVA ontology is based on [2], [3] and [7] with improvements and modifications tailored for the ViVA project and the considered scenarios. ViVA ontology proposes three main classes (see Figure 9):

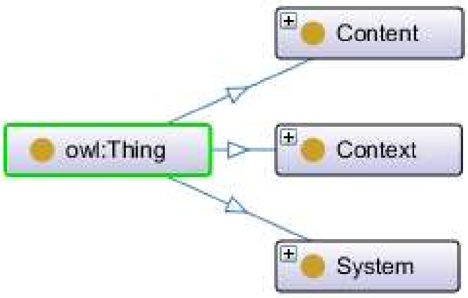

Figure 9 ViVA ontology contracted. Content, System and Context are disjoint classes that define the basic ontology hierarchy.

- Context. It contains all elements that provide information about the real context when the recording process is carried out.
- System. It defines the main hardware and software elements that belong to the system.
- Content. Visual elements that are found in online/offline videos that are captured by the cameras of the system.

These three main concepts and the ontology structure (with the most basic relation among concepts, the *hasSubclass* relation) defines a hierarchy. The *Content* and *Context* are the *superclassses* core of the ViVA ontology. Content represents from visual events, all the events of interest. Context complements the knowledge that the branch Content produces. On the other hand, System will provide information about the software and hardware capabilities of the surveillance system.

3.1.2. System

The class System defines the main hardware and software elements that belong to the real system, a hierarchical overview can be seen in the Figure 10.

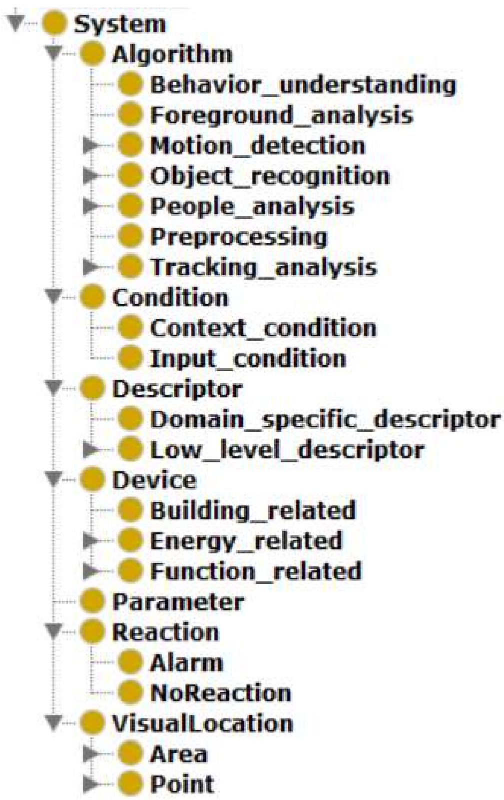

Figure 10 Hierarchical overview of the System class and its subclasses.

Main concepts of the System class are described in the Table 1.

Table 1 Main concepts of the System class

Concept	Description
Algorithm	In the surveillance context, a process followed to obtain visual features from video.
Condition	A requirement necessary to complete a task o calculation.
Descriptor	Visual feature of the content in images and videos.
Device	Represents a generic apparatus (hardware + software + firmware) intended to accomplish a particular task. It is a producer and/or consumer of data.
Parameter	A measurable factor that defines a system.
Reaction	A logical task that the system should do in response of some event.
VisualPosition	A position on image.

Processing capabilities of the system depends on the current weather conditions, so, a *hasWeather* relation is required (see Figure 11).

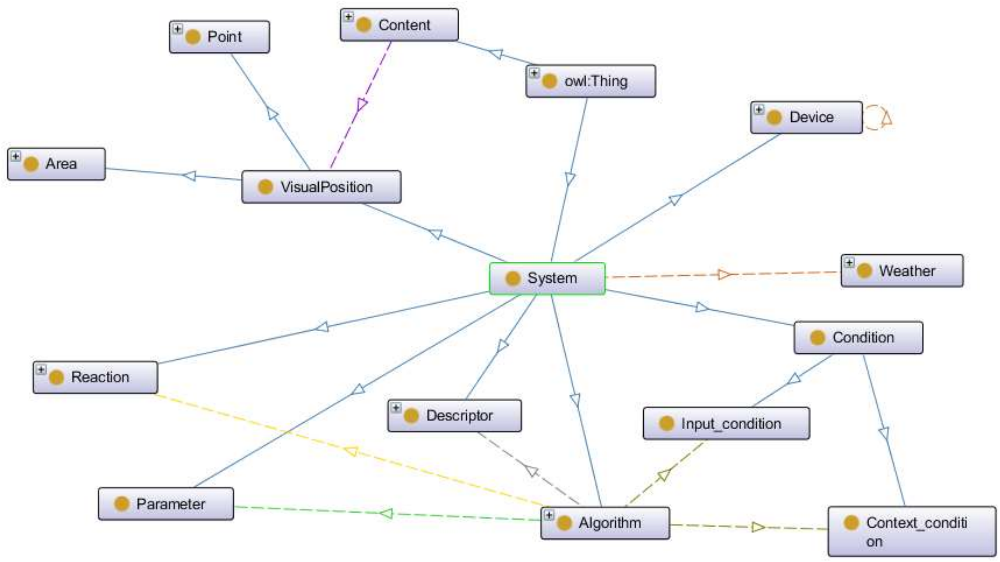

Figure 11 System class relations

Device is one of the main elements, is based on the SAREF ontology [9] and it represents different kinds of compound devices as smartphones or more simple devices as accelerometers or cameras. The second important element is the class *Algorithm*, that represents the vision algorithms that are implemented in the system. *HasInput_Condition* and *hasContext_condition* are required by the class in order to specify algorithms' preconditions as well as a relation with the subclass *Parameter*. If the algorithm has a response when something is detected the *hasReaction* relation is used.

The subclass *VisualPosition* is used to specify the position of the visual objects in the video. As the position of an object or event could be an area (as the box of one blob) or a centroid the subclasses *Area* and *Point* have been defined. *VisualPosition* is a subclass of *System*, that has two subclasses *Area* and *Point*. *Area* has two subclasses, Rectangle and Polygon. These subclasses have a *consistOf* relation with the class Point. Low level information is represented in the *Algorithm* class through the *hasDescriptor* relation.

3.1.3. Context

Information related to the context is contained in the *Context* class. It contains all elements that provide information about the real context when the recording process is carried out. For example, weather (Rainy, Sunny, Windy), place where the event is taking place (Church, School, Parking, Bank, etc.) and information about the video.

The Figure 12 shows the overview the hierarchy defined by the subclass relation of *Context*.

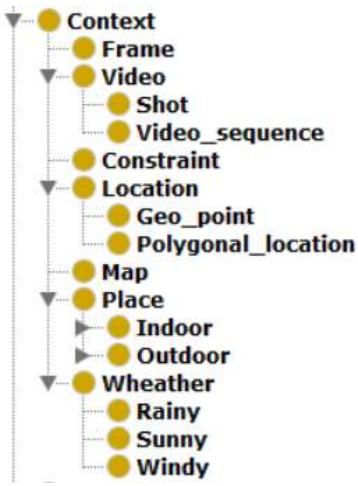

Figure 12 Hierarchical overview of the Context class. Frame, Video, Location, Map, Place and Weather are the main subclasses that provide information about the real context.

Frame, Video, Constraint, Location, Map, Place and Weather are the main concepts in the *Context* branch and are described in Table 2

Table 2 Main concepts of the Context class

Concept	Description
Frame	One image which compose a complete video.
Video	A recording of moving visual images made digitally or on videotape.
Constraint	A limitation or restriction.
Location	Represents a geospatial location

Map	A representation of an area of land or sea showing physical features, cities, roads,etc.
Place	A particular place with physical extension.
Weather	The state of the atmosphere at a place and time.

Figure 13 shows the main relations among the *Context* subclasses. The class video is defined by two relations *hasVideo_sequence* and *hasDevice*. The *hasDevice* relation defines the camera in which the video was recorded. The *Device* class is defined in the System superclass.

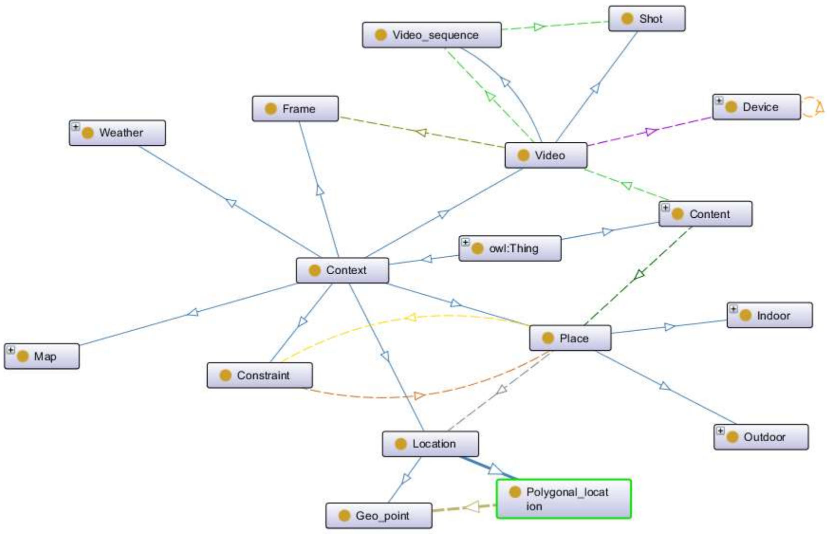

Figure 13 Relation overview of the Content class

The *Place* class is intended to represent the place where the event took place (indoor or outdoor). As can be seen, the *Place* class has a relation *hasConstraint* to indicate some constraints about the use of the place. For example, an instance of a public parking can have a constraint about the opening and closing times or in an airport some areas are forbidden to public access. This kind of information is intended to be captured by the *Constraint* class and the relation *hasConstraint* in Place.

Location class represents a geospatial location. In general, a location could be a point (a Geo_point instance, for example a person location) or polygonal for an area as parking (an instance of the Polygonal_Location). The Polygonal_location class has an object restriction to indicate that at least three points are necessary. Thus, the *Place* class also has a *hasLocation* relation with this class.

3.1.4. Content

All visual elements that can be captured by one camera are represented by this class. It contains two subclasses (see Table 3): Event and Object.

Table 3 Main concepts of the Content class.

Concept	Description
Event	Visual behavior or interaction of objects in the scene.
Object	All physical elements that can be seen in video.

Figure 14 shows a hierarchical overview of the main subclasses of the Content class.

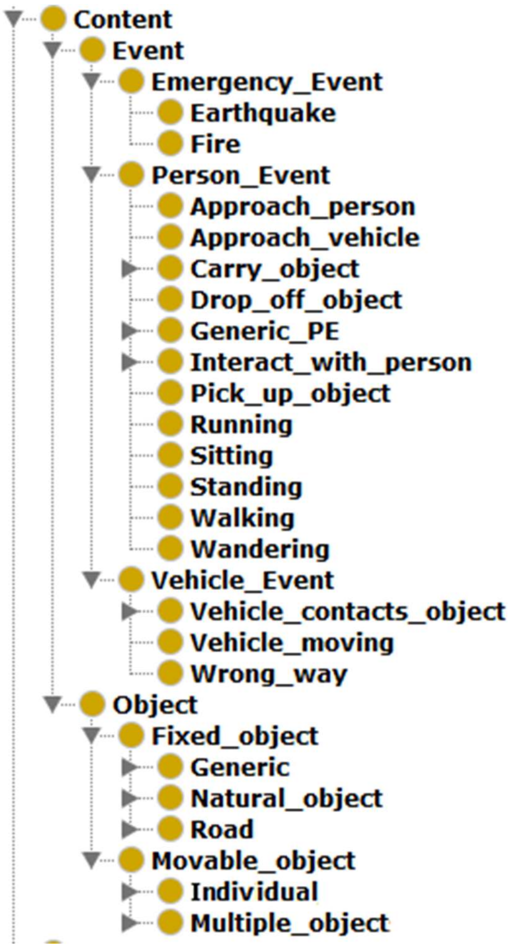

Figure 14 General hierarchical overview of the main subclasses of Content.

First, as all the Content are visual elements (Object or Event), they have a position in the video frame. Thus, as can be seen in Figure 15, Content class has a *hasVisualPosition relation* with the class *VisualPosition* to specify it. All the subclasses of *Content*, i.e., *Object* and *Event* have also the *hasPlace* and *hasVideo* relations.

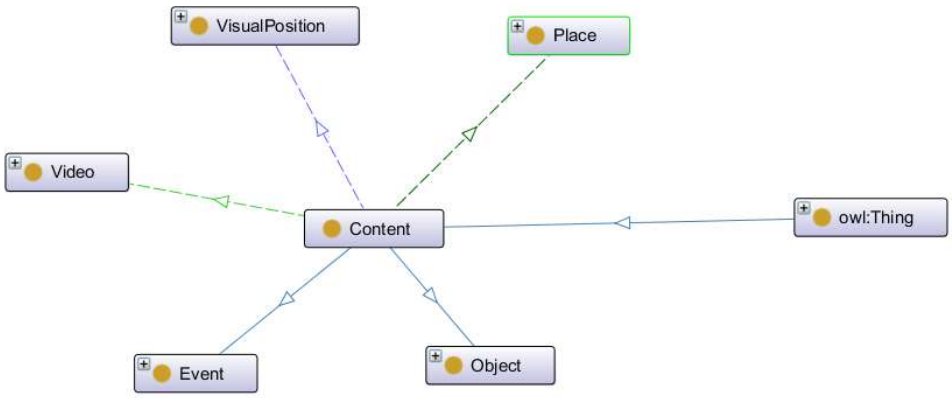

Figure 15 Relations of the Content class.

On the other hand, the main objective of the ontology is to capture the knowledge about security events. This is modeled through the Event entity. For ViVA ontology, an event is generated because of the behavior of one object or the interaction among objects. For example, a person walking or a person leaving an object. Thus, Event and Object are defined as subclasses of Content.

Event class defines three types (see Figure 14): *Person_Event*, *Vehicle_Event* and *Emergeny_Event*. *Person_Event* contains simple activities (*Standing, Running, Sitting, walking*), interaction with objects like in *Carry_Object* and interaction with other people (Fighting, Greeting, *Holding*, etc). Other complementary generic activities are contained in *PE_Generic* as *Enters_an_area*, *Exits_an_Area* and *Moving*.

Object is a subclass of *Content* because in a video is possible to detect all type of objects like backpacks, watches, smartphones, trees, etc. It is worth to noting that in this context, in the first stage of the visual processing a person also is an object. In the Object class, we can find two subclasses *Movable_Object* and *Fixed_Object (see* Figure 16*).*

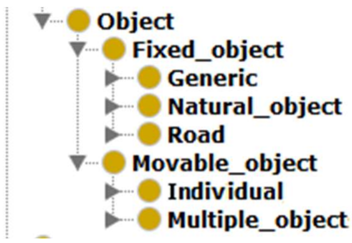

Figure 16 The Object branch of the ViVA ontology. Two subclasses are identified: Movable_object and Fixed_Object

A Movable_object represents a visual object that has capabilities of movement or which can be moved by a third party. Inside the *Movable_object* are two subclasses, *Individual* and *Multiple_Object. Multiple_object* (see Figure 17) defines a group of several individual objects. These individuals can be persons, animals and vehicles.

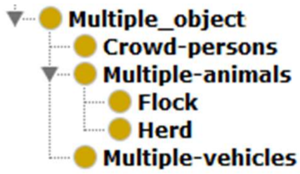

Figure 17 Movable_Object defined by several individual objects that could be persons, animals and vehicles.

Individual and *Movable_object* could be of several types of animals (*Bird* and *Mammals*), *Person* (*Child, Adult, Baby, Man, Woman*), *Vehicle* and *Man-made_object* (see Figure 18). A complementary *Bodypart* subclass is defined as individual because of their relevance as attribute to detect events or person based in this classes.

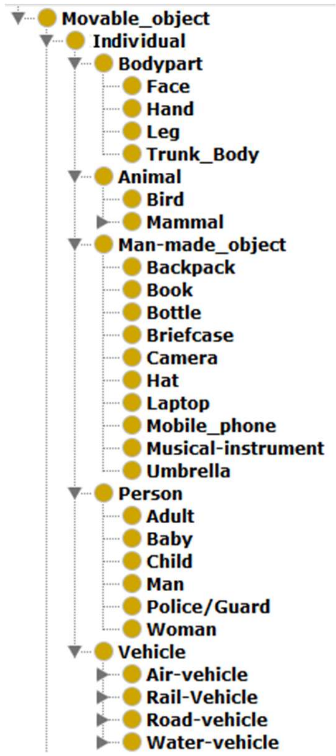

Figure 18 Individual branch defined on ViVA ontology.

Fixed_object are those objects considered that is no possible that it changes its position in the scene. Three types of fixed objects are defined: Generic, Natural and Road. Figure 19 shows these categories.

Figure 19 Fixed_Object branch defined in the ViVA ontology.

4. Discussion

First of all, an ontology is a way to formally represent knowledge, often specialized to some domain or subject matter. More precisely, it is not the vocabulary or concepts as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture. Thus, translating the terms in an ontology from one language to another, for example from English to French, does not change the ontology conceptually. The previous one, is an immediate benefit of use ontologies, a common vocabulary to get scalability in the system development. New components have been able of use the knowledge captured by the ontology. Thus, other processes as data modeling and data base implementation are simplified.

ViVA ontology can be extended in any of their three main branches in order add more devices or new capabilities of processing video to the surveillance system (branch System), and new surveillance domains (branch Content and Context).

ViVA ontology can be used as in [4] as a tool for selection of algorithms when the system starts and depending of the events that the user wants to detect. In other hand, If ViVA ontology is complemented with an inference system, also can be used as an intelligent module for automatic event detection in adaptive way.

5. Conclusions

The design process related to an ontology helps to achieve a deeper knowledge of the domain under analysis. Thus, in terms of software development the process will help for closing the semantic gap. More specifically, for the video surveillance domain the semantic gap between events of interest and a set of low level descriptors obtained by image processing or computer vision algorithms.

Thus, a novel ontology based on three main concepts has been described. ViVA ontology in addition to describe the concepts and their relations also propose a concept hierarchy defined by System, Content and Context branches. The ontology was designed with owl format and using Protégé.

As an iterative process, the ViVA ontology 1.0 will be extended for next stages and inference capabilities will be explored.

A description of all the concepts are listed in the html documentation related to the ViVA ontology.

6. References

[1] N. D. Rodríguez, M. P. Cuellar, J. Lilius and M. D. Clavo-Flores, "A Survey on Ontologies for Human Behavior Recognition," *Journal ACM computing surveys,* vol. 46, no. 4, p. 33, 2014.

[2] J. M. M. Á. G. Juan Carlos San Miguel, "An ontology for Event Detection and its Application in Surveillance Video," in *Sixth IEE International Conference on Advanced Video and Signal Based Surveillance*, 2009.

[3] A. L. A. G. I. M. B. Mohammed Yassine Kazi Tani, "Events Detection Using a Video-Surveillance Ontology and a Ruled-Based Approach," in *Computer Vision-ECCV Workshops*, 2014.

[4] "Performance Evaluation of Tracking and Surveillance," 2012. [Online]. Available: http://www.cvg.reading.ac.uk/PETS2012/. [Accessed 05 2017].

[5] "Protégé," The protégé project, 2012. [Online]. Available: http://protege.stanford.edu/. [Accessed 5 2017].

[6] A. M. T. H. V. N. Ngoc Q. Ly, "Specifc Behavior Recognition Based on Behavior Ontology," in *Recent Developments in Intelligent Information and Database Systems*, vol. 642, L. M. N. T. N. Dariusz Król, Ed., Springer, 2016, pp. 99-109.

[7] "Video Surveillance Online Repository," 2017. [Online]. Available: http://www.openvisor.org/config_schema.asp. [Accessed 5 2017].

[8] G. Jakus, V. Milutinovic, S. Omerovic and T. Saso, Concepts, ontologies and knowledge representation, New York Heidelberg Dordrecht London: Springer, 2013, p. 71.

[9] "Saref Ontology," [Online]. Available: http://ontology.tno.nl/saref/. [Accessed 05 2017].