Semantic Web 1 (2017) 1–5
IOS Press

Ontology for a Panoptes building: exploiting contextual information and smart camera network

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Roberto Marroquin *, Julien Dubois and Christophe Nicolle
Laboratoire Le2i - FRE 2005, CNRS, Arts & Métiers, Univ. Bourgogne Franche-Comté, Dijon, France
E-mail: {roberto-enrique.marroquin-cortez, julien.dubois, cnicolle}@u-bourgogne.fr

Abstract. The contextual information in the built environment is highly heterogeneous, it goes from static information (*e.g.,* information about the building structure) to dynamic information (*e.g.,* user's space-time information, sensors detections and events that occurred). This paper proposes to semantically fuse the building contextual information with data coming from a smart camera network by using ontologies and semantic web technologies. The ontology developed allows interoperability between the different contextual data and enables, without human interaction, real-time event detections to be performed and system reconfigurations. The use of semantic knowledge in multi-camera monitoring systems guarantees the protection of the user's privacy by not sending nor saving any image, just extracting the knowledge from them. This paper presents a new approach to develop a "all-seeing" smart building, where the global system is the first step to attempt to provide Artificial Intelligence (AI) to a building. More details of the system and future works can be found at the following website: `http://wisenet.checksem.fr/`.

Keywords: BIM, IFC, smart camera network, context-aware system, semantic interoperability, semantic gap, building AI

1. Introduction

In Greek mythology, Argus Panoptes was a giant with a hundred eyes. It was impossible to deceive his vigilance, for only some of his eyes slept while the rest were awake. Argus was the servant of Hera. At his death, Hera rewarded the giant's fidelity by transferring his eyes to the feathers of the peacock, his favorite animal. "To have the eyes of Argus" is a popular expression which means to be lucid and vigilant. The term Panoptes means "all-seeing". Within the built environment the term "all-seeing" is a quest in terms of access control, flow control and activities. In that context, a Panoptes building would characterize a smart building equipped with a network of cameras which could, in real-time, combine the different information seen and deduce the triggering of actions. In classical multi-camera based systems there is a monitor room with a central processing server where all the information is collected and analyzed in real-time by a human operator (or a set of them). However, as the size of the network increases, it becomes more difficult (or even impossible) for the human operator to monitor all the video streams at the same time and to identify events. Furthermore, having a large amount of information makes it infeasible to create a relation between actions that happened in the past and current actions.

Based on our experience, some issues and limitations of multi-camera based system deployed in built environments have been identified, such as:

*Corresponding author. E-mail: roberto-enrique.marroquin-cortez@u-bourgogne.fr.

- Selecting and filtering relevant information from the large amount of generated data.
- Dealing with missing information and non-detectable information. For example, a person may become *not-visible* to a camera due to its position or an obstruction.
- Reconfiguration of the cameras according to their context. The camera should adjust its configuration to ease the task of identifying a specific object or action. For example, the cameras should be able to adjust its parameters (*e.g.,* aperture and shutter speed) according to the light in the environment.
- Integration of data from different nodes and different domains. A multi-camera based system should be able to link the contextual information with the information coming from the different camera nodes to identify events and take decisions.
- Privacy protection. There are many privacy laws that restrict the monitoring of people, therefore, a multi-camera based system should be able to extract the useful information from an image/video while protecting the privacy of the individuals.

Many efforts have being devoted to deal with the aforesaid limitations of the multi-camera based system. The most prominent one is to rely on smart cameras to perform visual tasks semi-autonomously (with minimal human interaction). Smart cameras are specialized cameras that contain not only the image sensor but also a processing unit and some communication interfaces. In a few words, smart cameras are a self-contained vision systems [24,42]. The use of smart cameras in the built environment has become a growing trend due to the rich contextual data provided.

In the built environment, context is an essential factor since it provides information about the current status of users, places, objects, sensors and events. We assume that a smart building is a context-aware system because it extracts, interprets and uses the contextual information to automatically adapt its functionality according to the contextual changes.

A Panoptes building is a type of smart building that uses only smart cameras sensors and its main purpose is monitoring the different activities that occur in the built environment; in contrast to the smart building which uses different types of sensors and which mainly focuses on managing/monitoring the energy consumption. The creation of a Panoptes building is a complicate task due to the integration of data coming from different domains around the knowledge of the building composition. Many works have been done using semantic web standards such as Resource Description Framework (RDF) and Web Ontology Language (OWL) to represent contextual data [16]. On those systems, the ontology plays a crucial role in enabling the processing and sharing of information and knowledge, *i.e.,* the use of an ontology allows interoperability between different domains.

This paper proposes an ontology for a Panoptes building that re-purposes and integrates information about different domains composing the built context. The proposed ontology is the kernel of the WiseNET (Wise NETwork) system, which is a context-aware system whose main function is to perform reasoning about heterogeneous sources of information. Explicitly, the WiseNET system enhances the information of a smart camera network (SCN) with contextual information to allow autonomously real-time event/anomalies detection and system reconfiguration. The main contribution of this paper is the semantic fusion of Industry Foundations Classes (IFC) data with sensor information and other domain information in the Panoptes context, by using semantic web technologies. A semantic-based system is presented as well, which allows to overcome the multi-camera based system limitations and some computer vision problems, specially the privacy protection which nowadays is an important factor to consider.

The rest of the paper is organized as follows. Section 2 introduces the terminology used in the paper as well as summarizes related works. Section 3 describes the development process of the WiseNET ontology and its links to existing ontologies. Section 4 discusses the operational model, specifically the WiseNET system architecture. Section 5 presents the population of the ontology from the IFC, this is done once at the system initialization and is considered as `static population`. Section 6 presents the population of the ontology from smart cameras which consists of a `static population`, to initialize the sensors, and a `dynamic population` which occurs each time the smart cameras detect a person. Finally, Sections 7 and 8 present Discussion and Conclusions respectively.

2. Background and related work

Nowadays, multi-camera based systems have become a part of our daily life. They can be found in cities, commercial centers, supermarkets, offices, and even in houses. The advances in image sensor technology allow us to have smart cameras, which are low-cost and low-power systems that capture high-level description of a scene and analyze it in real-time [42]. These smart cameras can extract necessary/pertinent information from different images/video by employing different image processing algorithms such as face detection [47], person detection [8], people tracking [13], fall detection [35], object detection [33], etc.

Smart camera networks have been used in the built environment for a long time. The main applications focus on the following problematics:

- Study of space-use. Space-use is a research that aims at analyzing the relation between built space and its use. An example of this research is presented by Tomé *et al.* which studied the space-use by using computer vision-based tracking and Radio Frequency Identification (RFID) [38].
- Monitoring of elderly people. The context of this type of applications is health care and ambient assisted living. Some notable works in this field are: Nawaz *et al.* which used smart cameras to monitor the activities of elderly people while protecting their privacy [26]; and Crispim-Junior *et al.* which developed a multi-camera based framework to recognize the activities of elderly people during a clinical protocol in a hospital [7].
- Security, monitoring and surveillance. These are the most well-known applications in the built environment; they consists on using the visual information to monitor the activities of the building users. Yu *et al.* developed a robust image processing algorithm to perform multi-camera multi-object tracking in a built environment [46]; other examples in this field can be found in the survey of Winkler *et al.* [41].

Most of the previous applications use a SCN deployed in a built environment to obtain and analyze different type of information. Therefore, they might be considered as Panoptes building applications. The main function of a Panoptes building is to combine the different information obtained by the SCN and to deduce the triggering of actions/events in real-time. In that context, Panoptes building applications should understand the static building information as well as perceive (accurately) the dynamic and evolving data, *i.e.,* they should be aware of their context. A context-aware system in the built environment is a complex task; it requires information from different domains such as environment data, sensing devices, spatio-temporal facts and details about the different events that may occur. For example, the required event information could be a set of concepts and relations concerning the different events that may occur in a built environment, their location, the time they occurred, the agents involved, the relation to other events and their consequences. In the case of the sensor information, the required data could be the description of the different sensing devices, the process implemented on them and their results. Regarding the environment, the required data could be the structure of the building, its topology and the different elements contained in a space.

The built environment data can be obtained using the Building Information Modeling (BIM) of a building. BIM becomes a general term designing the set of numerical data, objects and processes appended during the life-cycle of a building [11]. From the designing, construction and facility management steps, the BIM allows practitioners and managers to exchange data in a uniform way using the IFC standard [40]. The IFC standard gives the base of description, both semantic and graphic of all elements making the building [20]. This allows to aggregate all heterogeneous software dedicated to the built environment on an interoperability way. In the domain of interoperability three levels are described: technical, organizational and semantics [21]. The IFC aims the technical interoperability level [9]. The organizational level is in charge of the practitioners according to the law of each country and the rules of each enterprise. The semantics level aims to clearly specify the meaning of each element making the BIM.

An important work was made to bring the IFC EXPRESS schema into the semantic web world using OWL as the schema modeling language. Dibley *et al.* compared different frameworks that tried to achieve this goal [10]. The result is the `ifcowl` ontology whose main objective is to convert the IFC concepts and instances data into equivalent RDF data. The conversion procedure from EXPRESS to OWL can be found in [30].

According to Studer, an ontology is a formal, explicit specification of a shared conceptualization [37]. In other words, an ontology is a set of concepts and relations used to describe and represent an area of concern. Currently, ontologies are represented using OWL-2 language, which is the recommendation of the World Wide Web Consortium (W3C) [43]. Other recommended technologies/languages in the semantic web field are: RDF, used for representing information in the form of a graph composed of triples [45]; RDF Schema (RDFS), which provides a vocabulary for creating a hierarchy of classes and properties [3]; SPARQL[1], used to query RDF data [44]; and the Semantic Web Rule Language (SWRL[2]), which is used for extending the OWL model with rule axioms [18].

One important application of ontology is for semantic fusion, which consists on integrating and organizing data and knowledge coming from multiple heterogeneous sources and to unified them into a consistent representation. There have being many works that combine semantic information coming from different sources in an ontology. Hong *et al.* presented some context-aware systems were the ontology plays a central role for enabling interoperability between devices and agents which are not designed to work together [16]. Dibley *et al.* developed an ontology framework that combines a sensor ontology with a building ontology and others supporting ontologies [10]. Other works have focused on using ontologies for combining computer vision with different kinds of information such as SanMiguel *et al.* which created an ontology composed mainly of knowledge about objects, events and image processing algorithms [34]; Chaochaisit *et al.* presented a semantic connection between sensor specification, localization methods and contextual information [5]; and Town which presented an ontology that fusion multiple computer vision stages with context information for image retrieval and event detections [39]. The use of ontologies as an interoperability agent warranties the information fusion. Consequently, we propose the creation of an ontology to combine and re-purpose the different types of information required by a Panoptes building.

Our approach differs from classical computer vision which deals with algorithm improvements [46] and signal processing problems [35], by dealing with a meaning problem in computer vision [39], where the observation of the smart camera is improved by semantically fusing it with contextual information (*e.g.,* position of the sensors, position of the users, spaces in the environment and events that have occurred).

[1]SPARQL is a recursive acronym for SPARQL Protocol and RDF Query Language.

[2]Currently (February, 2017) SWRL is not a W3C recommendation yet.

3. Formal modeling

The WiseNET ontology is an OWL-2 ontology that incorporates a vast corpus of concepts in the domain of a Panoptes building. The ontology provides a vocabulary for combining, analysing and re-purposing the information coming from the smart camera network (SCN) deployed in a built environment. The main function of the WiseNET ontology is to perform real-time event/anomalies detection and initiate system reconfiguration.

3.1. Ontology development

The goal for developing the WiseNET ontology was to create a shared understanding of the structure of information for a Panoptes building. The WiseNET developing process follows the Noy and McGuinness methodology for ontology development [27]. This methodology consists of seven steps which are: determine scope of the ontology, consider reuse, enumerate classes, define classes and properties, define constrains and create instances (ontology population shown on Sections 5 and 6).

3.1.1. Scope of the ontology

To determine the scope of the ontology we need to think about the kind of knowledge that should be covered by the ontology and its use, *i.e.,* its domain.

Some competency questions were formulated to determine the focus of the ontology (Table 1). Those competency questions should be answered by the ontology and from them it can be extracted the different kind of knowledge that should be contained in the WiseNET ontology. Roughly, is knowledge about the environment, events, person, sensors and time.

3.1.2. Links to existing ontologies

When developing a new ontology it is recommended to reuse existing ontologies as much as possible. In this way, one can focus on defining the specific knowledge of the application. The reuse of external ontologies, not only saves time but also gives the advantage of using mature and proved ontological resources that have been validated by their applications and (some) by the W3C.

The WiseNET ontology reuses resources from many different ontologies (see Table 2). However, there are six key ontologies that cover most of the required concepts of the different domains, those are:

Table 1
Selected competency questions for the WiseNET ontology

Competency questions	
How many people are in the space X?	Which doors does the camera Z observes?
Where is the person Y located?	What are the nearest cameras of camera Z?
What is the position of all the persons?	Which building elements does the camera Z observes?
Where was the person Y in the last T minutes?	What image processing algorithms are implemented in the camera Z?
How many rooms does the storey X has?	Is somebody in a restricted area?
Is the space X empty/occupied?	At what time does the person Y enter/left the space Z?
Is the space X1 connected to the space X2?	How long does the person Y stayed in the space X?
What are all the spaces connected to the space X?	Where were all the people at time T?
Which types of sensors are in the system?	At what time does the event E occured?
What is the position of all the sensors?	Who was involved in the event E?
Where is the camera Z located?	Which events happened in the last T minutes?

E, X, X1, X2, Y, Z are id's and T is a variable.

Table 2
Full list of prefixes and namespaces used in WiseNET ontology and in this document

Prefix	Namespaces	Description
`DUL`	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#	DOLCE+DnS Ultralite ontology
`event`	http://purl.org/NET/c4dm/event.owl#	The event ontology
`ifcowl`	http://ifcowl.openbimstandards.org/IFC2X3_TC1#	The IFC2X3 ontology
`owl`	http://www.w3.org/2002/07/owl#	The OWL 2 Schema vocabulary
`person`	http://www.w3.org/ns/person#	ISA Person Core Vocabulary
`rdf`	http://www.w3.org/1999/02/22-rdf-syntax-ns#	The RDF Concepts Vocabulary
`rdfs`	http://www.w3.org/2000/01/rdf-schema#	The RDF Schema Vocabulary
`ssn`	http://purl.oclc.org/NET/ssnx/ssn#	Semantic Sensor Network Ontology
`time`	http://www.w3.org/2006/time#	OWL-Time ontology
`wisenet`	http://wisenet.checksem.fr#	The WiseNET ontology
`xml`	http://www.w3.org/XML/1998/namespace	XML Specification
`xsd`	http://www.w3.org/2001/XMLSchema#	XML Schema Definition

– The `DUL` ontology, which provides a set of concepts used for interoperability among different ontologies [12]. This ontology gives the necessary properties to combine spatial information with different types of data.
– The `event` ontology, which deals with the notion of events and their different properties such as location, time, agents, factors and products [32]. This ontology provides most of the vocabulary required for describing activities and events that may happen.
– The `ifcowl` ontology, which is a semantic representation of the IFC schema (standard for representing building and construction data) [28]. Most of the environment concepts required (such as the structure of the building, its topology and the different elements contained in a space) can be obtained from the IFC. The IFC data used during the experimentations was defined using the IFC2x3 specification, therefore we focused on that version of the `ifcowl`. However, the system should be able to cope with newer versions of the IFC.
– The `person` ontology, which provides a minimum set of classes and properties for describing a natural person [31].
– The `ssn` ontology, which describes sensors, observations, sensing processes, measurement capabilities and related concepts [6]. This ontology provides the required vocabulary to describe the sensors used in our system.
– The `time` ontology, which provides concepts for describing the temporal properties of resources [15]. This ontology provides all the required concepts about instants, intervals, their duration and their topological relations.

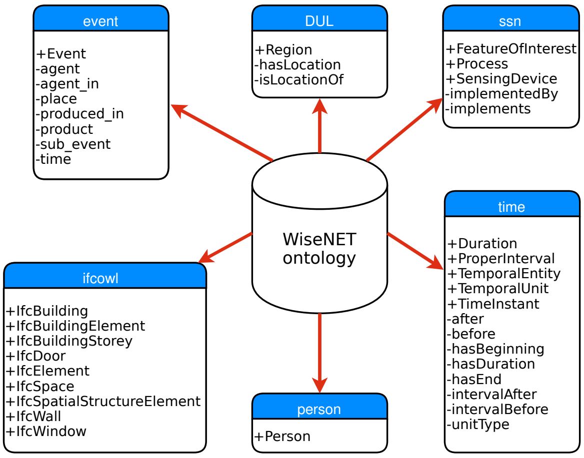

Fig. 1. Primary classes and properties reused by the WiseNET ontology. Classes are marked with (+) and properties with (-). The namespaces and brief description of the ontologies can be found in Table 2.

Figure 1 shows the primary classes and properties reused by the WiseNET ontology. The external ontologies were not imported, from most of them only some concepts are reused, except from the `ifcowl` from which some instances are also considered. Not importing the external ontologies gives two main benefits: easing the ontology maintenance and improving the performance. The performance improvement is a very important factor due to the goal of the WiseNET ontology is to perform real-time reasoning.

3.1.3. *Classes and properties of WiseNET*

Many of the competency questions involves more than one type of knowledge. Hence, the WiseNET ontology should be able to collect and combine the information from those different domains. In that context, it is necessary to define new concepts (classes and properties) that will allow us to: complete the information from the different domains, to describe attributes of instances according to our needs and, more importantly, to relate (*i.e.,* link) the different domains.

The Tables 3 and 4 present some selected classes and properties with emphasis on built environment information. We propose to enhance the IFC information by adding functional facts to the spaces, *i.e.,* information about the space usage (if it is a corridor, a reception, a coworking space, etc). Additionally, we propose to add information about the different types of alarm and security systems present in the built environment, specifically the security systems of the doors (*e.g.,* key-lock system, card reader system and biometric system). This can allow the deduction of knowledge regarding the security restrictions of a space.

After having the complete terminology of the ontology, some constraints and characteristics of the class expressions and the property axioms need to be defined. Axioms are a set of formulas taken to be true and that every assignment of values should satisfied. Those constraints and characteristics will determine the expressiveness and decidability of the ontology, and their definition will depend on the description logic used.

3.2. *Ontology decidability*

Description logics (DLs) are a family of formalism for representing knowledge [1]. The most notable applications for the DLs is to provide the logical formalism for ontologies languages such as OWL. OWL-2, the current W3C ontology language recommendation, is based on the expressive description logic $\mathcal{SROIQ}(\mathcal{D})$ [17]. $\mathcal{SROIQ}(\mathcal{D})$ extends the well known description logic $\mathcal{SHOIN}(\mathcal{D})$ by including: a role constructor by general inclusion ($\mathcal{R}$) and a qualified number restriction ($\mathcal{Q}$). $\mathcal{SROIQ}(\mathcal{D})$ provides high expressive power with high computational cost of reasoning. Hence, to meet a more suitable compromise between the expressive power and the computational cost, the WiseNET ontology was defined using the $\mathcal{SHOIQ}(\mathcal{D})$ language which is more expressive than $\mathcal{SHOIN}(\mathcal{D})$ yet less expressive than $\mathcal{SROIQ}(\mathcal{D})$ and less computational complex [22]. A definition of the $\mathcal{SHOIQ}(\mathcal{D})$ constructors and some examples referencing the WiseNET ontology can be found in the Table 5.

Horrocks and Sattler presented a tableau decision procedure for $\mathcal{SHOIQ}(\mathcal{D})$ that solves the ontology consistency problem and allows the use of reasoning services, thus demonstrating the decidability of $\mathcal{SHOIQ}(\mathcal{D})$ [19]. One of the few requirements to preserve the decidability in $\mathcal{SHOIQ}(\mathcal{D})$ is to restrict the application of the *qualified number restriction* to simple roles, *i.e.,* roles that are neither transitive nor have a transitive subrole [19]. This restriction is satisfy in the WiseNET ontology.

However, knowledge representation formalisms of the semantic web (such as DLs) have expressive limitations, for example composition of complex classes from classes and properties. Those limitations can be overcome by rule-based knowledge, specifically by using SWRL (Semantic Web Rule Language) rules [18]. SWRL rules are represented as implication of an antecedent (Body) and a consequent (Head):

$$b_1, b_2, ..., b_n \rightarrow h \,,$$

where $b_1, b_2, ..., b_n$: Body and h : Head.

Table 3
Selected WiseNET classes

Class	Description
BoundingBox	Box covering a region, detection or a detectable object. It is defined by the coordinates of the top left point, its width and its height
BuildingSecuritySystem	Security systems used in a built environment
Corridor	Space used for connecting different spaces. People doesn't stay for a long time on it
CoworkingRoom	Shared working space. Usually more than one person is present
DetectableObject	Abstraction of a real world objects/entities that may be observed by the camera
DoorSecuritySystem	Systems used to secure a door. It may be a simple key system, a card reader, a biometric system, etc
Environment	The environment where the system is located, it may be a building, hospital, park, train station, airport, etc
FieldOfView	Field of view (FOV) of a camera. Complete region of an image
ImageAlgorithm	Image processing algorithm to analyze and/or manipulate a digitized image
InstantEvent	Type of event that occurs in an specific point in the time/space
IntervalEvent	Type of event that occurs in a time interval
RegionOfInterest	Abstract regions of interest (ROI) on a field of view
Room	Space that is part of a building storey. It can have many functions, such as: toilet, reception, corridor, etc. A room may be restricted or non restricted
SecuredDoor	Door with a special security system, such as: card reader, code, biometric identification, etc
SmartCamera	Type of camera that has signal processing unit and communication interfaces
SpaceConnector	Element that connects spaces, for example a door
StandardDoor	Door with a key lock system
SystemReaction	Reaction of the system to different events, such as: trigger alarm, start recording, call for assistance, etc

Table 4
Selected WiseNET properties

Property	Description
aggregates	Similar to ifc:IfcRelAggregates. Property used for saying that a spatial structure element contains another one
appearsOnFieldOfview	Physical elements or objects that appears on a field of view
aroundSpaceConnector	Property that defines if a person has been detected around a space connector
cameraID	ID number of a smart camera
fieldOfViewOverlaps	Two fields of views overlap if they observe the same door
hasBoundingBox	Bounding box covering a region, an object or a detection
hasLocation	Position of an object or detection in a space
hasNearbyCamera	Relation between two cameras that are in the same space or that are located in connected spaces
hasRegionOfInterest	Abstract region of interest (ROI) of a field of view
hasSecuritySystem	Type of security system used. For example for a door it could be a key system, a card reader, etc
personPreviousLocation	Previous location of a person
personStillInSpace	Boolean property that says if a person is still on a space or not.
regionOfInterestRepresents	Region on the field of view of a camera that represents a building element
restrictedArea	Boolean property to say if a space is restricted or not
spaceConnectedTo	Two spaces are consider connected if they share a door
spaceContains	Similar to ifc:ContainedInSpatialStructure. Property for relating spaces to different building elements such as doors, walls, windows, etc
spaceHasAlarm	Alarm system of a space
visualDescritors	Visual features used to describe a detection
xywh	Vector that defines a bounding box:'x,y' are the coordinates of the top left point 'w,h' are the width and the height of the bounding box

Table 5
Definition of $\mathcal{SHOIQ}(\mathcal{D})$ constructors

$\mathcal{SHOIQ}(\mathcal{D})$ constructor	Definition	Examples in WiseNET ontology using Turtle syntax [45]
$\mathcal{S}$	Transitivity of roles	`time:before rdf:type owl:TransitiveProperty.` `:isPartOf rdf:type owl:TransitiveProperty.`
$\mathcal{H}$	Role hierarchies	`:aggregates rdfs:SubPropertyOf :environmentProperties.` `event:agent rdfs:SubPropertyOf :eventProperties.`
$\mathcal{O}$	Nominals	`_:x rdf:type owl:AllDifferent;` `owl:distictMembers(:PersonDetector,:FaceDetector).`
$\mathcal{I}$	Inverse role	`DUL:hasLocation owl:inverse DUL:isLocationOf.` `ssn:implements owl:inverse ssn:implementedBy.`

$\mathcal{Q}$	Qualified number restrictions	`_:x rdf:type owl:Restriction;`
		`owl:onProperty :spaceContains;`
		`owl:minQualifiedCardinality "1"^^xsd:integer;`
		`owl:onClass :SpaceConnector.`

$(\mathcal{D})$	Datatypes	`:isRestrictedArea rdf:type rdfs:Datatype.` `:xywh rdf:type rdfs:Datatype.`

`_:x` represents a blank node (anonymous individual).

Reasoning becomes undecidable for the combination of OWL + SWRL, therefore the expressivity of SWRL needs to be reduced in order to assure decidability. Although many procedures exists to guarantee decidability of SWRL, the DL-safe rules was adapted [25]. This procedure consists on restricting the number of possible variables assignments, *i.e.,* restricting the application of rules only to known OWL individuals (named individuals).

Examples of DL-safe rules implemented in the WiseNET ontology are presented in Listing 1 and Listing 2. The first one states that if there are two spaces 'x' and 'y', and both contain the door 'd', then those spaces are connected to each other. The second one states that if there are two spaces 'x' and 'y', and two smart cameras 's1' and 's2', and 'x' is connected to 'y', and 's1' is located in 'x' and 's2' is located in 'y', then those smart cameras are nearby each other.

Listing 1: SWRL rule for `spaceConnected`.

```
Space(?x), Space(?y),
Door(?d), spaceContains(?x,?d),
spaceContains(?y,?d) -> spaceConnectedTo(?x,?y)
```

Listing 2: SWRL rule for `hasNearbyCamera`.

```
Space(?x), Space(?y),
SmartCamera(?s1), SmartCamera(?s2),
spaceConnectedTo(?x,?y), hasLocation(?s1,?x),
hasLocation(?s2,?y) -> hasNearbyCamera(?s1,?s2)
```

To recapitulate, the WiseNET ontology decidability was insured by restricting the application of the *qualified number restriction* and by using DL-safe rules. As a result, a semantic reasoner can be employed for inferring logical consequences from a set of asserted facts or axioms. After finishing the formalization of the ontology the next step is to connect it to an operational architecture that enables the insertion of the required facts.

4. Operational modeling

The WiseNET ontology is the kernel of the WiseNET system. WiseNET is a semantic-based system that fuses heterogeneous sources of data such as data coming from sensors and the different contextual information. Due to the application of the paper (Panoptes building), we will focus in a specific type of sensor: smart cameras; however, the system is defined to in-

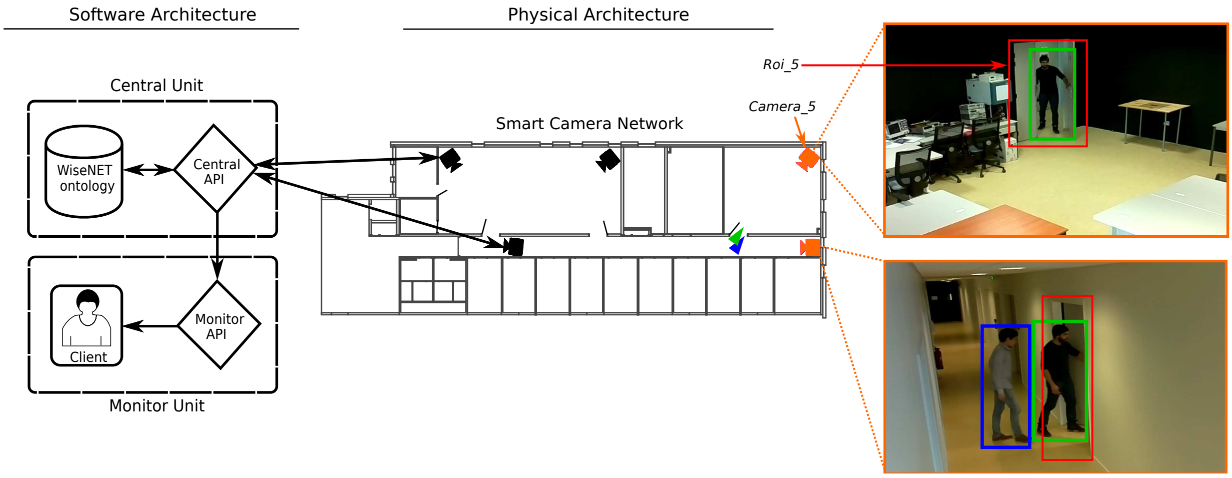

Fig. 2. WiseNET system overview: the smart camera network is in charge of extracting pertinent information from a scene, to convert the data into knowledge and to send it to the central API, which afterwards will populate the WiseNET ontology with that knowledge. The central API is also responsible for reconfiguring the image processing algorithms in the smart cameras and for transferring data to the monitor unit from which a client/user may visualize the history of activities. Notice that all the smart cameras are connected to the central API, however to keep the image cleaner, only the connection of two smart cameras are shown. The definition of the colors is as follows: in the 2D map (central part of the image), the orange cameras represent cameras that are detecting people (*e.g.,* `Camera_5`), and the blue an green triangles represent the position of the only two people presented in the storey; in the camera views (right part of the image), the green and blue bounding boxes represent detected people and the red one represent regions of interest (*i.e.,* `Roi_5`), in this case the region around the door.

clude other type of sensors such as temperature, humidity, depth sensor, etc. The main goal of WiseNET is to improve classical computer vision and deep learning systems by considering the contextual information of the environment and by performing real-time reasoning. As a result, WiseNET may overcome some limitations of computer vision (*e.g.,* false detections and missed detections), some drawbacks of deep learning (*e.g.,* the need of large amount of training and testing data) and limitations of multi-camera based system (presented in Section 1) while allowing real-time event/anomalies detection and system reconfiguration.

Figure 2 illustrates the architecture of the WiseNET system. The system is articulated in three sections: the smart camera network, the central unit and the monitor unit. The SCN is a set of smart cameras distributed in an environment. The main functions of the SCN, in the WiseNET system, is to extract low-level features from a scene (such as person detection as shown in blue and green in the right side of the Figure 2), to convert the extracted data into knowledge and then to send it to the central unit. More information regarding the type of smart cameras used can be found in [23]. The central unit is composed of two elements the central API and the WiseNET ontology. The central API is in charge of the management of the ontology, for example: capturing the knowledge coming from the SCN and insert it to the ontology, retrieving inferred knowledge from the ontology, transferring data to the monitor unit, sending new configurations to the smart cameras, and other services. The WiseNET ontology is responsible of enabling interoperability between the incoming knowledge streams and the contextual data (*e.g.,* environment information, previous knowledge and sensor information) in order to deduce new knowledge and detect events/anomalies based on the history of activities. The central unit is also in charge of re-identifying people in the system by using their visual features extracted by the SCN. Eventually, the central unit could request extra information to the SCN. The monitor unit has as main function the visualization of the static and dynamic information; this unit will automatically retrieve information and will present it in a graphical manner, for example an occupancy map (as shown at the center of Figure 2). The monitor unit implements some queries to answer questions such as: how many people is in a room? what is the location of a person? and many others (see Table 1).

The proposed system is context-aware and combines the information extracted by the SCN with logic rules and knowledge of what the camera observes, building information and events that may occurred. Our system differs from other computer vision systems mainly by three factors. First, no images are sent, the smart cameras extract the knowledge from the images and then this knowledge is sent to the central unit (for more details see Section 6.1). Second, the WiseNET system combines context information with the camera information to improve the detections, in this way, it can overcome missed detections or non-detectable information (*e.g.,* people out of sight of a camera). Third, the system uses an ontology to fusion the different kinds of information presented in a Panoptes building, such as: information of the environment, time, smart cameras, events, detectable objects, etc.

Notice that the architecture shown in Figure 2 presents the WiseNET system as a centralize system where there is no communication between the smart cameras, however the system could also be deployed in a distributed manner as presented in [2].

Once the ontology is formally defined and implemented in a triplestore (a database for the storage of triples), the last step of the ontology development (as stated in Section 3) is to populate it. The next two sections will present the population from the IFC file and from the sensors respectively. Note that the second one consists of two parts. Firstly, the initialization of the sensors is populated, consisting on the description of the sensors and their relation to the built environment. Secondly, an ongoing population is performed each time the smart cameras detect a person in the building.

5. Ontology population from IFC

After inserting the WiseNET ontology in the system, the *a priori* information about the built environment needs to be populated (inserted). The required information can be extracted from the IFC file of the environment. This population will be performed only once at the initialization of the system, therefore is considered as a `static population`.

The I3M (Institut Marey et Maison de la Métallurgie) building located in Dijon (France), will be used as an example for this section. The I3M building is composed of three building storeys from which we will focus on the third storey where a SCN has been deployed.

An IFC2x3 file, describing all the elements composing the I3M building, was used in this project. It was obtained from the company in charge of the construction of this building and it was generated using the Revit CAD software[3]. Only a small portion of the IFC file is needed in the WiseNET system. Therefore, to improve the ontology performance, only the required information will be extracted from the IFC file and populated in the ontology.

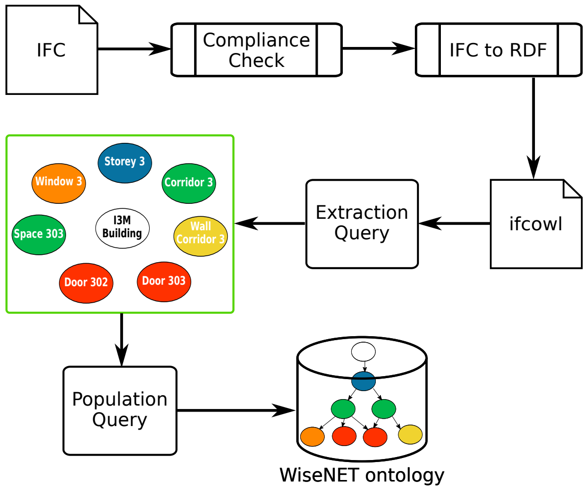

Fig. 4. Extraction and population framework.

Figure 3 shows, in the form of a graph, the main classes and properties required to be extracted from an IFC file and populated on the ontology. The extracted/populated data consists on information about the building, building storeys, spaces and elements contained on those spaces (such as doors, windows, walls). Furthermore, the topology of the building is also required, *i.e.,* the relation between the building and the building storeys, between the building storeys and the spaces, between the spaces and the different elements (doors, windows and walls) and the relations between two spaces. We assume that two spaces are connected if and only if they share a door.

A framework was developed for extracting and populating the required IFC data into the WiseNET ontology (see Figure 4). The extraction/population framework employs semantic web technologies and it consists mainly of four processes: a compliance check of the IFC, a conversion of the IFC into `ifcowl`, the extraction of the pertinent instances from the `ifcowl` and finally, the population of the extracted instances and their relating properties into the WiseNET ontology.

The framework starts with an IFC file of a built environment, in this case the I3M building. The requirements of the IFC is to contain the following entities: `IfcBuildingStorey`, `IfcRelSpaceBoundary`, `IfcRelDecomposes`, `IfcBuilding`, `IfcDoor`, `IfcWindow`, `IfcWall`, and `IfcSpace`. The compliance check will verify the fulfillment of those re-

[3]http://www.autodesk.com/products/revit-family/overview

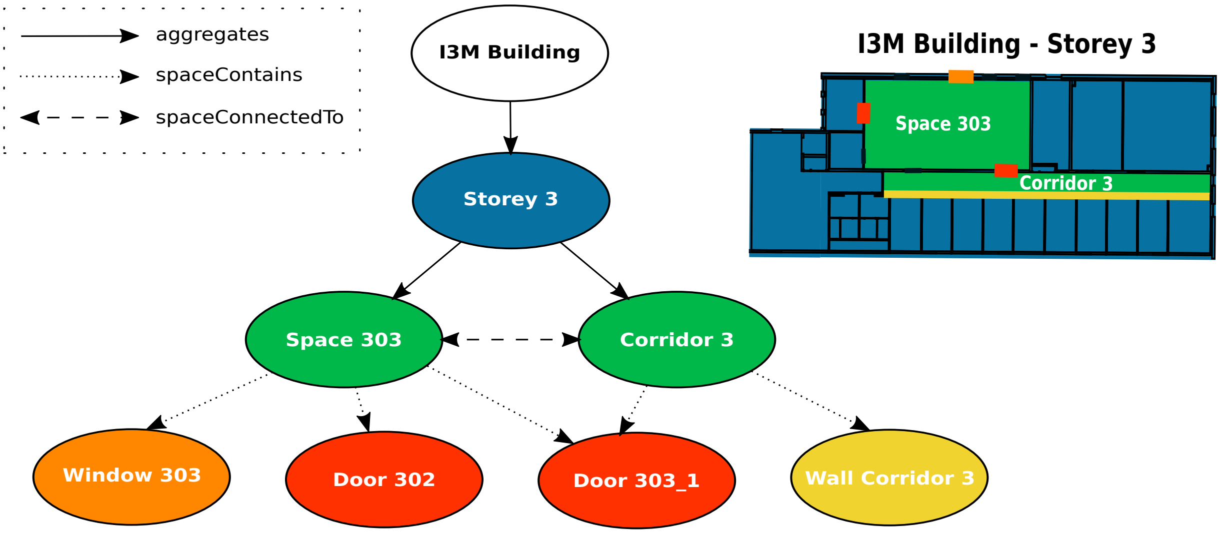

Fig. 3. A fragment of the graph extracted from the IFC file of the I3M building. In the right side it can be observed a 2D view of the third storey of the I3M building. The graph represents a small selection of spaces and elements presented in the third floor. The colors of the graph nodes have two functions, first to make a correlation between the graph and the 2D view, second, to denote different IFC classes: the Storey 3 belongs to the class `IfcBuildingStorey`; the Space 303 and the Corridor 3 belongs to the class `IfcSpace`; the Door 302, the Window 303 and the Wall Corridor 3, belongs to the classes `IfcDoor`, `IfcWindow` and `IfcWall` respectively. The graph shows the main classes and properties required to be extracted from the IFC file.

quirements. Afterwards, the IFC is converted to RDF by using the *IFC-to-RDF converter* by Pauwels and Oraskari [29]. The result of the conversion is the `ifcowl` ontology with instances of the I3M building.

5.1. *Extraction query*

The `ifcowl` will be queried using SPARQL in order to extract the pertinent instances. The Listing 3 shows the SPARQL code used for the extraction. Line 4 obtains the building instance by using its class. Line 7 acquires the array of building storeys that decompose the building; and line 8 obtains the storeys inside that array. The same is done for the spaces that decompose the storeys on lines 11-12. Lines 15-16 obtain the elements that are contained in a space. Lines 19-22 filter out the undesired elements just leaving the doors, windows and walls. Finally, line 25 saves all the classes of the elements. The result of the extraction query is a table where the columns corresponds to the variables used with the SELECT operator (line 1). This table is called the *extracted table*.

5.2. *Population query*

The *extracted table* contains a set of instances without any relations between them. That is why, the population query will create those relations using the WiseNET properties and then will insert them to the WiseNET ontology. To accomplish this, the population query will process row by row the *extracted table*. For exemplification, lets assume that the first row of the *extracted table* has the following values:

```
?building = inst:Building_I3M,
?storey = inst:Storey_3,
?space = inst:Space_303,
?element = inst:Door303_1,
?elementType = ifcowl:IfcDoor,
```

where `inst` is a special prefix, defined in `ifcowl`, used to define instances. Now, those instances will be the input of the population code shown in the Listing 4. Line 3-5 relate the extracted instances of the `ifcowl` with the WiseNET properties. Line 1 inserts the previously created relations into the WiseNET ontology. This process needs to be repeated for all the rows of the *extracted table*, which can be achieved by using an external loop.

As aforesaid, we assumed that two spaces are connected if and only if they share a door. This property `spaceConnectedTo` could be obtained from queries but its quite complex, therefore it was decided to formulate a rule to obtain this property (see Listing 1).

To summarize, in order to populate the WiseNET ontology with the *a priori* environment knowledge, the

Listing 3: SPARQL query for extracting instances from the `ifcowl` ontology. Lines that start with # are comments.

```
SELECT ?building ?storey ?space ?element ?elementType
WHERE {
   # Get building
   ?building rdf:type ifcowl:IfcBuilding.

   # Get building storeys
   ?storey_array ifcowl:relatingObject_IfcRelDecomposes ?building.
   ?storey_array ifcowl:relatedObjects_IfcRelDecomposes ?storey.

   # Get spaces: room, corridors, hall, etc
   ?space_array ifcowl:relatingObject_IfcRelDecomposes ?storey.
   ?space_array ifcowl:relatedObjects_IfcRelDecomposes ?space.

   # Get elements: doors, windows, walls, floor, furnitures, etc
   ?element_array ifcowl:relatingSpace_IfcRelSpaceBoundary ?space.
   ?element_array ifcowl:relatedBuildingElement_IfcRelSpaceBoundary ?element.

   # Filter elements to just keep doors, walls and windows
   {?element rdf:type ifcowl:IfcDoor}
   UNION {?element rdf:type ifcowl:IfcWall}
   UNION {?element rdf:type ifcowl:IfcWallStandardCase}
   UNION {?element rdf:type ifcowl:IfcWindow}.

   # Get the classes of elements
   ?element rdf:type ?elementType
}
```

Listing 4: SPARQL query for inserting relations on the WiseNET ontology. Lines that start with # are comments.

```
INSERT DATA {
    # Creating relations between extracted instances
    inst:Building_I3M wisenet:aggregates inst:Storey_3.
    inst:Storey_3 wisenet:aggregates inst:Space_303.
    inst:Space_303 wisenet:spaceContains inst:Door_303_1.

    # Inserting the type of element in wisenet
    inst:Door_303_1 rdf:type ifcowl:IfcDoor
}
```

relevant information from the IFC file needs to be extracted, then share it and connect it to the WiseNET ontology using queries, rules and linked data techniques such as Uniform Resource Identifiers (URIs) and RDF. Linked data technology connects data from one data-source to other data-sources, in our case, linked data allows the WiseNET ontology to obtain extra information from the `ifcowl` if required (*e.g.,* the dimensions of a door, its material and the dimensions of a wall).

6. Ontology population from smart cameras

The built information has already been added to the WiseNET ontology, the next step is to populate the information about the sensors. A complete smart camera network has been installed in the third storey of the I3M building. The smart cameras are based on the Raspberry Pi 3 system [23] [4].

[4] A demonstration of the deployed network can be found at `http://wisenet.checksem.fr/`

There are two types of information that needs to be populated concerning the smart cameras. Firstly, there is the smart camera setup information, which consist on describing the smart cameras and their relation to the built environment. Secondly, there is the detection information, which occurs each time the smart cameras performs a detection. The first one will be populated once, at the system initialization, therefore is considered as a static population. The second one will be populated multiple times (each time there is a detection), therefore it will be considered as a `dynamic population`.

6.1. *Static population*

Figure 5 presents the user interface of the software developed for adding and setting up smart cameras in the system. The software helps to perform the following tasks:

- Assigning a smart camera to a space. First, the user has to input the camera name and then the system will automatically propose a set of spaces that are obtained by querying the `ifcowl`.
- Giving semantic meaning to image regions. There is a semantic gap between the visual information of an image and its physical representation. Ideally 3D information obtained from the IFC file could be used for bridging this gap by making an automatic labeling of the built elements (such as doors, walls, etc) in the camera image. However, some problems have been encountered while performing automatic matching between the 3D image (bottom-left in Figure 5) and the camera image (top-left in Figure 5). The main challenge lies in obtaining the 3D coordinates and angles of the camera in the real world. Consequently, it was chosen to manually label the camera image by drawing regions of interest (ROIs) and assigning their representation in the built environment (as shown in the Figure 5). The system automatically proposes a set of elements for representation (*e.g.,* doors and windows) according to the selected space, information obtained by querying the `ifcowl`.

There is optional information that may be added using the setup software, such as:

- Assigning specific image processing algorithms to the smart camera, by default it is person detection.
- Assigning specific function to the space, by default they are simple spaces.
- Saying if the space has an alarm or not and the type of alarm, by default there is no alarm.
- Assigning some security system to the doors, by default the doors have the key-lock system.

The information concerning the restriction of a space is deduced by adding a rule stating: if a space has a door with a security system different than key-lock then that space is a restricted area. A space with doors that have key-lock system could also be a restricted space but it needs to be stated directly.

The *WiseNET smart camera setup* software is connected to a SPARQL endpoint that inserts the information set by the user by running the query shown in the Listing 5. This step can be seen as a soft camera calibration that requires only the location of the cameras in the building. This differs from many multi-camera based systems, that required overlapping between the fields of views of the cameras and their orientation, leading to a time-consuming and skill-dependent calibration process [36]. An important contribution of using an ontology during the smart camera setup is the automatic suggestion of pertinent elements according to the space. This is achieved by using the `static population` of the building information.

6.2. *Dynamic population*

As aforementioned, the main functions of the smart cameras is to firstly detect pertinent information using different image processing algorithms and secondly to extract the knowledge from the images and to send it to the central unit.

For the Panoptes building application the main detectable object of the smart cameras is the person. Therefore, three different image processing algorithms have been implemented, person detection, face detection and fall detection[5]. After detecting some pertinent information, the smart camera describes what it "observes" by using the vocabulary defined in the WiseNET ontology, *i.e.,* the smart camera extracts the knowledge of the scene. This addition of semantic meaning to what the camera "observes" is a problem known as `semantic gap` [39].

For instance, consider the scene "observed" by the smart camera `Camera_5` (top-right image in the Fig-

[5]The details of the image processing algorithm are outside of the scope of this paper and they will be presented in a future work

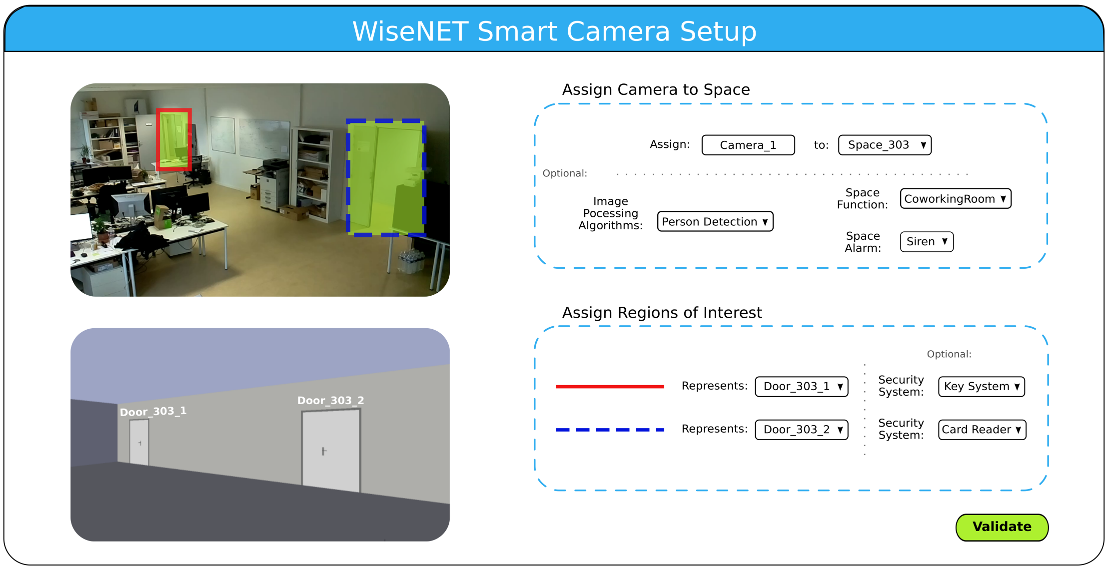

Fig. 5. User interface of the WiseNET smart camera setup software. In this example the *Camera_1* is being configured to perform *Person Detection* algorithm and is being assigned to the *Space_303*. Additionally, the *Space_303* is being defined as a *CoworkingRoom* with an alarm of type *Siren*. Furthermore, the blue region of interest (located in the top-left image) is assigned to represent the *Door_303_2* which has as security system *Card Reader*, similar for the red region of interest. The 3D view (bottom-left) was obtained from the IFC file. The boxes with the black arrows are combo boxes where the system makes automatic suggestions according to the WiseNET ontology and the `ifcowl`.

Listing 5: WiseNET camera setup query. Lines that start with # are comments.

```
INSERT DATA {
    # Creating new instances
    wisenet:Camera_1 rdf:type owl:NamedIndividual;
                     rdf:type wisenet:SmartCamera.
    wisenet:Fov_1 rdf:type owl:NamedIndividual;
                  rdf:type wisenet:FieldOfView.
    wisenet:Roi_1 rdf:type owl:NamedIndividual;
                  rdf:type wisenet:RegionOfInterest.
    wisenet:Roi_2 rdf:type owl:NamedIndividual;
                  rdf:type wisenet:RegionOfInterest.

    # Creating relations
    wisenet:Camera_1 DUL:hasLocation inst:Space_303;
                     wisenet:hasFieldOfView wisenet:Fov_1.
    wisenet:Fov_1 wisenet:hasRegionOfInterest wisenet:Roi_1;
                  wisenet:hasRegionOfInterest wisenet:Roi_2.
    wisenet:Roi_1 wisenet:regionOfInterestRepresents inst:Door_303_1.
    wisenet:Roi_2 wisenet:regionOfInterestRepresents inst:Door_303_2.

    # Optional relations
    inst:Door_303_1 wisenet:hasSecuritySystem wisenet:KeySystem.
    inst:Door_303_2 wisenet:hasSecuritySystem wisenet:CardReaderSystem.
    inst:Space_303 rdf:type wisenet:CoworkingRoom;
                   wisenet:spaceHasAlarm wisenet:Siren.
    wisenet:Camera_1 ssn:implements  wisenet:PersonDetection.
}
```

ure 2) where a person is being detected in the region of interest `Roi_5`. The smart camera will describe that scene in the following form:

```
cameraID: "Camera_5",
ImageAlgorithm: "Person Detection",
RegionOfInterest: "Roi_5",
xywh: "107,20,30,50",
visualDescriptors: "29,31,45",
```

where all the variables corresponds to terms defined in the WiseNET ontology (see Tables 3 and 4). The `xywh` are the coordinates of the detection box (green bounding box in Figure 2). The visual descriptors of the detection box were obtained by using the RGB histogram method. This is a classic image processing method that consist on taking the most preeminent color tone for each channel (R (red), G (green), B (blue)). It is possible to use different visual descriptors such as different color space or even physical characteristics (*e.g.,* height, head size and shoulder width) [4]. After describing the scene, the smart camera sends that knowledge to the central unit by using web services. Moreover, when a person is detected an `instant event` is created. An `instant event` is a type of event that occurs in an specific point in time/space. If it is the first time a person is detected in a specific space, then the event 'person in space' is also created. This is an `interval event` and, as its name indicates, it occurs in a time interval (*i.e.,* it has a starting and an end time). The 'person in space' event is an array containing the detections of a person in an specific space. Finally, the central API will dynamically insert this knowledge into the ontology. The central API has many functions such as performing the statics and dynamic population and managing the system reconfiguration. Currently, the reconfiguration task is in development process.

7. Discussion

This paper focused on creating an ontology model to fusion and re-purpose the different types of information required by a Panoptes building. Once the model is assembled it needs to be evaluated to verify if it satisfy its intent.

Currently, the smart camera network (SCN) has already been deployed and the image processing algorithms have been embedded on it. However, the central API is in development process, therefore it is not possible to perform an evaluation of the complete system. Moreover, the WiseNET ontology may be evaluated. According to Hitzler *et al.* the accuracy criteria is a central requirement for ontology evaluation [14]. This criteria consists on verifying if the ontology accurately captures the aspects of the modeled domain for which it has designed for. The WiseNET ontology development was based on some competency questions (see Table 1), therefore the evaluation consists on showing that those questions can be answered by the ontology. Listing 6 presents the queries to answer some competency questions. Those questions were selected because they involve aspects which are important in a Panoptes building such as: knowing how many spaces there are in the storeys, which doors a smart camera monitors, how many people are in a space, at what time a person enter/leave a space and how much time a person stayed on a space. Question 1 (of Listing 6) was answered by getting all the elements aggregated by the building storeys. For answering question 2, the regions of interest (ROIs) in a camera's field of view and their physical representation are obtained. Question 3 was answered by counting the number of 'person in space' events in the specific space. Question 4 and 5 were answered by using the time interval entity of the event 'person in space'. The time interval entity has a beginning, end and duration (the duration is giving in a temporal unit).

Regarding the built environment, in this paper the definition of a Panoptes building was given, as well as its formalization by using an ontology. Also, two types of populations in the Panoptes environment were defined, static and dynamic. Furthermore, we believe that a classification of smart building should be done due to the generality of the actual definition of smart building. This classification may be created by considering: the functionalities of the smart building, the devices utilize, or a combination of both. The third type of classification was applied by defining a Panoptes building as a smart building using only cameras and focusing in monitoring the activities of people. Moreover, we considered that applications concerning smart buildings should exploit the built environment information, specially the data obtained from the IFC. We also proposed to enhance the IFC information by adding functional facts to the spaces and information about the security systems in the built environment.

Concerning the WiseNET system, we believe that a semantic-based system may present great advantages over the classical computer vision and deep learning systems. First of all, the WiseNET system does not needs training and testing data like the other systems.

Listing 6: Queries to answer selected competency questions. Lines that start with # are comments.

```
###########
# QUESTION 1: How many spaces does the storeys have?
SELECT ?storey (COUNT (?x) AS ?numberOfSpaces)
WHERE{
        ?storey rdf:type ifcowl:IfcBuildingStorey;
                wisenet:aggregates ?x.
}
GROUP BY ?storey

###########
# QUESTION 2: Which building elements does the camera 'Camera_1' observes?
SELECT DISTINCT ?elementsType
WHERE {
        wisenet:Camera_1 wisenet:hasFieldOfView ?x.
                         wisenet:hasRegionOfInterest ?y.
        ?y wisenet:RegionOfInterestRepresents ?elements.
        ?elements rdf:type ?elementsType.
        FILTER (?elementsType != owl:NamedIndividual)
 }

###########
# QUESTION 3: How many people are in the space 'Space_303'?
SELECT COUNT (?x) AS ?numberOfPeople
WHERE {
        ?x rdf:type wisenet:PersonInSpace;
           event:place inst:Space_303.
}

###########
# QUESTION 4: A what time does the person 'Person_1' enter the space 'Space_303'?
SELECT ?enteringTime
WHERE {
        ?x rdf:type wisenet:PersonInSpace;
           event:place inst:Space_303;
           event:product wisenet:Person_1;
           event:time ?timeInterval.
        ?timeInterval time:hasBeginning ?timeInstant.
        ?timeInstant time:inXSDDateTime ?enteringTime.
}

###########
# QUESTION: How long does the person 'Person_1' stayed in the 'Space_303'?
SELECT ?timeDuration ?temporalUnit
WHERE {
        ?x rdf:type wisenet:PersonInSpace;
           event:place inst:Space_303;
           event:product wisenet:Person_1;
           event:time ?timeInterval.
        ?timeInterval time:hasduration ?intervalDuration.
        ?intervalDuration time:numericDuration ?timeDuration.
        ?intervalDuration time:unitType ?temporalUnit.
}
```

Secondly, by sending only the extracted knowledge of the image the amount of data transmitted through the network is lower than the other systems. Furthermore, a semantic-based system allow us to efficiently fusion different type of information (as shown in this paper), specially environmental information, which gives advantages over other type of systems. Those and other advantages will be studied in future work.

Regarding the privacy protection, it is important to remark that the SCN used in the WiseNET system do not send or save any image, in that way the privacy of the individuals is protected. However, one exception could be done if the ontology infers that an illegal act is occurring, in that case, the central API can use that inferred knowledge to send a message to the smart camera telling it to start recording and to save the images locally (to have them as a proof). Even in that special case no images are send through the network, in this way the system is more secured.

The queries shown in Listing 6 could be extended to obtain information that enables the reconfigure the SCN. The bi-directionality between the SCN and the central unit is a novelty in the semantic web domain. Specifically, the mechanism of using the ontology to interact with external devices. In general, the ontologies are not designed to be used in that way, therefore a central API is needed to be designed to perform two main tasks. Firstly, to receive all the messages from the SCN, synchronize them (by using timestamps) and then populate the ontology. Secondly, to check the ontology inferred knowledge and to use it for reconfigure the SCN by triggering a specific action (*e.g.,* recording and triggering an alarm), changing the image processing algorithm or by saying to the smart camera to focus on a specific ROI. The dependency on the central API leaves an open question about the automatic externalization of the ontology knowledge, *i.e.,* making the ontology to automatically output the inferred knowledge.

8. Conclusion and prospectives

According to the legend of Argus we tried to develop a "all-seeing" smart building, which we have called Panoptes building. With that motivation, the WiseNET ontology was developed. Its main goal is to fusions the different built environment contextual information with information coming from the smart camera network and other domain information, to allow real-time event detections and system reconfiguration. The purpose of the developed ontology is to create a kernel of a Panoptes building system (*i.e.,* the WiseNET system), rather than working towards publishing another generic ontology. The ontology development procedure was performed using different semantic technologies, and it consisted of: defining a set of questions that the ontology should answer (competency questions); reusing different domain ontologies (`DUL`, `event`, `ifcowl`, `person` and `ssn`); creating a set of classes and properties to connect the different domain ontologies and to complete the application knowledge; defining a set of constrains and extending the expressiveness by using logic rules; and finally, populating the ontology with static information (built environment and smart camera setup) and dynamic information (smart camera detections).

The WiseNET system is a semantic-based real-time reasoning system, that fuses different sources of data and is expected to overcomes limitations of multi-camera based system. The WiseNET system selects relevant information from the video streams and adds contextual information to overcome problems of missing information due to false/missed detections. Additionally, it relates events that occurred at different times, without human interaction. It also protects the user's privacy by not sending nor saving any image, just extracting the knowledge from them. It may as well, reconfigure the smart camera network according to the inferred knowledge. In few words, the WiseNET system enables interoperability of information from different domains such as the built environment, event information and information coming from the smart camera network. A future goal of the WiseNET system is to offer services to building users according to information coming from a network of sensors deployed on the built environment and contextual information. This is a highly complex task due to the large scope of the building system, that goes from the static physical structure of the built environment to the internal environment in terms of the dynamic building users and the way how they interact with the building facilities.

The future works will focus on completing the externalization of the ontology knowledge using the central API and on properly evaluate and compare the semantic-based system against a classical computer vision system and a deep learning system.

9. Acknowledgement

The authors thank the Conseil Régional de Bourgogne Franche Comté and the french government for their fundings.

We would also like to thank Ali Douiyek and Arnaud Rolet for the technical assistance and management of the technical team. As well as E.Grillet, A.Goncalves, D.Barquilla, E.Menassol, L.Lamarque, G.Kasperek and M.Le Goff for their help in the development of the 3D visualization tools of the building.

References

[1] F. Baader. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[2] L. Ben Khelifa, L. Maggiani, J.-C. Quinton, and F. Berry. Antcams network: a cooperative network model for silly cameras. In *Proceedings of the 10th International Conference on Distributed Smart Camera*, pages 104–109. ACM, 2016.

[3] D. Brickley, R. V. Guha, and B. McBride. RDF Schema 1.1. *W3C recommendation*, 25:2004–2014, 2014.

[4] A. Burbano, S. Bouaziz, and M. Vasiliu. 3d-sensing distributed embedded system for people tracking and counting. In *Computational Science and Computational Intelligence (CSCI), 2015 International Conference on*, pages 470–475. IEEE, 2015.

[5] W. Chaochaisit, M. Bessho, N. Koshizuka, and K. Sakamura. Human localization sensor ontology: Enabling owl 2 dl-based search for user's location-aware sensors in the iot. In *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on*, pages 107–111. IEEE, 2016.

[6] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al. The SSN ontology of the W3C semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32, 2012.

[7] C. F. Crispim-Junior, V. Buso, K. Avgerinakis, G. Meditskos, A. Briassouli, J. Benois-Pineau, I. Y. Kompatsiaris, and F. Bremond. Semantic event fusion of different visual modality concepts for activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1598–1611, 2016.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[9] T. M. de Farias, A. Roxin, and C. Nicolle. SWRL rule-selection methodology for ontology interoperability. *Data & Knowledge Engineering*, 105:53–72, 2016.

[10] M. Dibley, H. Li, Y. Rezgui, and J. Miles. An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, 28:1–14, 2012.

[11] C. M. Eastman, C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011.

[12] A. Gangemi. Ontology:DOLCE+DnS ultralite homepage. `http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite`. Lastest update on 4 March 2010.

[13] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *IEEE Signal Processing Magazine*, 22(2):38–51, 2005.

[14] P. Hitzler, M. Krotzsch, and S. Rudolph. *Foundations of semantic web technologies*. CRC Press, 2009.

[15] J. R. Hobbs and F. Pan. Time ontology in OWL. *W3C Working Draft*, 2016.

[16] J.-y. Hong, E.-h. Suh, and S.-J. Kim. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4):8509–8522, 2009.

[17] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. *Kr*, 6:57–67, 2006.

[18] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21:79, 2004.

[19] I. Horrocks and U. Sattler. A tableau decision procedure for SHOIQ. *Journal of automated reasoning*, 39(3):249–276, 2007.

[20] International Organization for Standardization. ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. `https://www.iso.org/standard/51622.html`.

[21] International Organization for Standardization. ISO 19439:2006 Enterprise Integration-Framework for Enterprise Modelling. `https://www.iso.org/standard/33833.html`.

[22] Y. Kazakov. RIQ and SROIQ are harder than SHOIQ. *Description logics*, 2008.

[23] R. Marroquin, J. Dubois, and C. Nicolle. WiseNET-smart camera network interacting with a semantic model: PhD forum. In *Proceedings of the 10th International Conference on Distributed Smart Camera*, pages 224–225. ACM, 2016.

[24] R. Mosqueron, J. Dubois, M. Mattavelli, and D. Mauvilet. Smart camera based on embedded HW/SW coprocessor. *EURASIP Journal on Embedded Systems*, 2008:3, 2008.

[25] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(1):41–60, 2005.

[26] T. Nawaz, B. Rinner, and J. Ferryman. User-centric, embedded vision-based human monitoring: A concept and a healthcare use case. In *Proceedings of the 10th International Conference on Distributed Smart Camera*, pages 25–30. ACM, 2016.

[27] N. F. Noy, D. L. McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[28] P. Pauwels, T. M. de Farias, C. Zhang, A. Roxin, J. Beetz, and J. De Roo. Querying and reasoning over large scale building data sets: an outline of a performance benchmark. In *Proceedings of the International Workshop on Semantic Big Data*, page 11. ACM, 2016.

[29] P. Pauwels and J. Oraskari. IFC-to-RDF-converter. `https://github.com/mmlab/IFC-to-RDF-converter`. Lastest update on 30 April 2016.

[30] P. Pauwels and W. Terkaj. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63:100–133, 2016.

[31] I. Programme. ISA programme person core vocabulary. `https://www.w3.org/ns/person/`. Lastest update on

18 November 2013.

[32] Y. Raimond and S. Abdallah. The event ontology. Technical report, Citeseer, 2007.

[33] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa. Object detection, tracking and recognition for multiple smart cameras. *Proceedings of the IEEE*, 96(10):1606–1624, 2008.

[34] J. C. SanMiguel, J. M. Martinez, and Á. Garcia. An ontology for event detection and its application in surveillance video. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 220–225. IEEE, 2009.

[35] B. Senouci, I. Charfi, B. Heyrman, J. Dubois, and J. Miteran. Fast prototyping of a SoC-based smart-camera: a real-time fall detection case study. *Journal of Real-Time Image Processing*, 12(4):649–662, 2016.

[36] S. Soro and W. Heinzelman. A survey of visual sensor networks. *Advances in multimedia*, 2009, 2009.

[37] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[38] A. Tomé, M. Kuipers, T. Pinheiro, M. Nunes, and T. Heitor. Space–use analysis through computer vision. *Automation in Construction*, 57:80–97, 2015.

[39] C. Town. Ontological inference for image and video analysis. *Machine Vision and Applications*, 17(2):94–115, 2006.

[40] R. Vanlande, C. Nicolle, and C. Cruz. IFC and building lifecycle management. *Automation in construction*, 18(1):70–78, 2008.

[41] T. Winkler and B. Rinner. Security and privacy protection in visual sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 47(1):2, 2014.

[42] W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. *computer*, 35(9):48–53, 2002.

[43] World Wide Web Consortium and others. OWL 2 web ontology language document overview. 2012.

[44] World Wide Web Consortium and others. SPARQL 1.1 overview. 2013.

[45] World Wide Web Consortium and others. RDF 1.1 concepts and abstract syntax. 2014.

[46] S.-I. Yu, Y. Yang, and A. Hauptmann. Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3714–3720, 2013.

[47] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.