

INTER-IoT-EWS

D1.1 – Semantic translations

University of Twente, Netherlands

João Moreira ([j.luizrebelomoreira@utwente.nl](mailto:j.luizrebelomoreira@utwente.nl))

+31(0)53 4894337 / +31(0)6 17108979

Building Zilverling 4074, P.O. Box 217, 7500 AE Enschede

Small Collaboration

Abstract

In accordance with the INTER-IoT challenges [1], we intend to support the semantic and syntactic interoperability among IoT artefacts and early warning systems (EWS), i.e. enable data to be understandable for both sender and receiver. In particular, we focus on coordinating emergency services based on IoT devices, alerting the involved parties (e.g. emergency command control, first responders and employees) when an accident occurs.

Our objective is to test the applicability of INTER-IoT solutions in the context of semantic interoperability capabilities of the INTER-IoT framework through the IoT-based EWS, enabling data exchange among heterogeneous IoT artefacts by developing emergency application services that use IoT semantic translations provided by INTER-IoT Inter-Platform Semantic Mediator (IPSM) component. Furthermore, we propose a low-cost business model for the cross-domain scenario of accidents at the port of Valencia (id.9), where transportation companies, haulers and insurance companies can benefit from the IoT EWS by reducing disaster risks involving their employees and the goods being transported.

In this deliverable we describe the ontologies analyzed according to the use cases’ requirements. In particular, the roles of W3C Semantic Sensor Network (SSN), along with the Sensor Observation Sampling Actuator (SOSA), and the Smart Appliances REFerence (SAREF) ontologies are presented. Furthermore, this document presents the required mappings to enable interoperability of IoT platforms and the EWS. These mappings were discussed with the INTER-IoT consortium and the alignments developed according to the IPSM translation language, tested with sample data in INTER-IoT test environment (RedNode and IPSM). Besides these ontologies, other assets were studied to address information requirements from the use cases, representing the domains of e-Health (e.g. HL7 aECG, FHIR, UFO ECG), logistics (LogiCO, LogiTrans, LogiServ) and emergencies (OASIS EDXL, EM-DAT, W3C EIIF, incident management ontology). Recommendations for the extension and other alignments with these assets are included.

Contents

[Abstract 2](#_Toc494279923)

[Contents 3](#_Toc494279924)

[1 Accidents at the port area scenario 4](#_Toc494279925)

[1.1 Description 4](#_Toc494279926)

[1.2 Stakeholders 7](#_Toc494279927)

[1.3 Goals 8](#_Toc494279928)

[1.4 Requirements 11](#_Toc494279929)

[1.5 IoT artefacts involved 15](#_Toc494279930)

[1.5.1 Fleet Management system: Movildata 16](#_Toc494279931)

[1.5.2 Valenciaport Port Community System (PCS) 17](#_Toc494279932)

[1.5.3 Truck’s owner (haulier) IoT platform 17](#_Toc494279933)

[1.5.4 Port Control System 17](#_Toc494279934)

[1.5.5 Emergency Control Centre IoT platform 17](#_Toc494279935)

[1.5.6 Port authority IoT platform 17](#_Toc494279936)

[1.5.7 IoT solution for INTER-Health 18](#_Toc494279937)

[1.5.8 Accelerometer data provider (Android-based mobile application) 21](#_Toc494279938)

[1.5.9 Azure IoT platform 21](#_Toc494279939)

[1.6 Related scenarios 21](#_Toc494279940)

[1.7 Terms and definitions 22](#_Toc494279941)

[2 Use cases 23](#_Toc494279942)

[2.1 UC01: Vehicle collision detection 26](#_Toc494279943)

[2.2 UC02: Hazardous change in driver’s vital signs 30](#_Toc494279944)

[2.3 UC03: Vehicle collision related (temporarily) with changes in driver’s vital signs 34](#_Toc494279945)

[2.4 UC04: Vehicle in wrong way 35](#_Toc494279946)

[2.5 UC05: Accident involving dangerous goods 35](#_Toc494279947)

[2.6 Risks identified for the implementation of the use cases 36](#_Toc494279948)

[References 36](#_Toc494279949)

# Use cases’ requirements

## (UC01) Vehicle collision detection

Obs.1: consider Threshold\_VehicleCollision equivalent to 4g.

Obs.2: accelerometerShimmer and accelerometerMobile implements the interface Accelerometer (W3C).

A classification of severity

The situation types (rules) to be addressed in this use case are:

**ST\_UC01\_01.** Vehicle collision with one accelerometer (Shimmer): activated when collision is detected from the Shimmer’s accelerometer.

IF (ComputeCrossAxialEnergy(accelerometerShimmer) > Threshold\_VehicleCollision)

Obs.: Classification of severity and urgency according to accelerometer (A) data and threshold (B) is described in the table below. In summary, if the cross-axial energy computed is greater than the threshold and less than 20% above the threshold, then it might be a light collision (minor severity). If it is in-between 20% and 40%, then the collision is greater (moderate severity), if it is in-between 40% and 60%, then the collision is severe. Above 60% represents a strong impact, thus, an extreme severity, which probably needs immediate urgency for emergency response.

|  |  |  |
| --- | --- | --- |
| **Range** | **Severity** | **Urgency** |
| B < A <= B \* 1.2 | Minor | Expected |
| B \* 1.2 < A <= B \* 1.4 | Moderate | Immediate |
| B \* 1.4 < A <= B \* 1.6 | Severe | Immediate |
| B \* 1.6 < A | Extreme | Immediate |

**ST\_UC01\_02.** Vehicle collision with one accelerometer (mobile): activated when collision is detected from the mobile’s accelerometer.

IF (A=ComputeCrossAxialEnergy(accelerometerMobile) > B=Threshold\_VehicleCollision)

Obs.: same classification of severity and urgency of **ST\_UC01\_01**.

**ST\_UC01\_03.** Vehicle collision with two accelerometers (Shimmer and mobile): activated when collision is detected from the Shimmer’s accelerometer and the mobile phone within a configurable delta time.

IF (exists(over window:time(10s, ST\_UC01\_01 AND ST\_UC01\_02))

AND ST\_UC01\_01.Driver = ST\_UC01\_02.Driver)

Obs.1: we consider an initial value of 10s for delta time. Figure XX illustrates two examples of timeline where this situation type is activated (accident is detected).

Obs.2: notice that one situation type needs to be linked to the other (as an “INNER JOIN”) through some identity object, such as *Driver* or *Truck* or *Device*. Otherwise, this situation type would be activated based on two different trucks, which is incorrect.

Obs.3: same classification of severity and urgency of **ST\_UC01\_01**.

**ST\_UC01\_04.** Vehicle collision with one accelerometer (Shimmer or mobile): activated when collision is detected from the Shimmer’s accelerometer or the mobile phone, where the decision is based on the battery consumption of the device (and the effect on data accuracy).

IF (exists(over window:time(10s, ST\_UC01\_01 OR ST\_UC01\_02))

AND ST\_UC01\_01.Driver = ST\_UC01\_02.Driver)

Obs.1: we consider an initial value of 10s for delta time. Figure XX illustrates two examples of timeline where this situation type is activated (accident is detected).

Obs.2: same classification of severity and urgency of **ST\_UC01\_01**.

## (UC02) Hazardous health changes

In this

## (UC03) Temporal relations between UC01 and UC02

In this

## (UC04) Wrong-way driving

In this

## (UC05) Accidents with dangerous goods

In this

# Assets studied

The initial.

## W3C Semantic Sensor Network (SSN)

The W3C Semantic Sensor Network (SSN) is an ontology developed by W3C, (current published version 1.0, 2011). It provides a comprehensive framework to describe sensors, devices, observations, measurements and other terms, enabling reasoning of individual sensors and the connection of sensors, such as wireless networks. SSN 1.0 is grounded in a set of existing ontologies and standards, such as CSIRO, SWAMO, SEEK Extensible Observation, SemSOS and OGC SensorML. The main concept of SSN is the Sensing Device, which is a sensor that reports measurements and observations of real world phenomena. A sensor is different in nature from other types of devices, e.g. actuators, because of its event-based behavior, which requires temporal relationships. SSN enables reasoning, which can facilitate the development of advanced applications, for ex- ample, by reasoning about sensor measurements, considering constraints as power restriction and limited memory. It consists of 10 modules, representing 41 concepts and 39 object properties. It inherits 11 concepts and 14 object properties from the foundational ontology DOLCE-UltraLite (DUL). In this paper we cover the following modules:

DUL module5: represents the foundational categorization of Designed Artifact, Method, Physical Object, Quality, Re- gion and Situation. For example, a Sensing Device (Meas- uring module) is a Designed Artifact and a Physical Object, which observes a Property (Skeleton module). A Property is an observable Quality of an Event or Object, i.e. ”an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor”. Skeleton module: represents the most basic concepts re- garding sensors, as Sensor, Sensing, Property and Obser- vation. A Sensor may be a physical device implementing Sensing, i.e. it has a sensing method observing some Prop- erty. ”Sensing is a process that results in the estimation, or calculation, of the value of a phenomenon”. Measuring module: covers the elements Sensing Device and Sensor DataSheet. The prior is the main element of SSN. The former represents the data sheet specifications of a sensor. Usually, the properties of a sensor are recorded dir- ectly with hasMeasurementCapability property of a Sensor. System module: represents the System concept as a Phys- ical Object (DUL) composed by sub-systems (hasSubSys-tem), which has deployment(s) (hasDeployment), operat- ing range(s) (hasOperatingRange) and location(s) relative to other entities (onPlatform). Measuring Capability module: represents core concepts of SSN, i.e. properties and capabilities of measurements, such as Accuracy, MeasurementProperty and Measurement- Capability. Relevant object properties are the hasMeas- urementCapability and hasMeasurementProperty. Measure- mentCapability represents a characteristic of a sensor’s ob- servations or ability to make observations (e.g. accuracy and range). MeasurementProperty represents the collection of measurement properties and environmental conditions in which those properties hold. Device module: covers Device, which is a physical piece of technology (a ”system in a box”) and can be composed of other (smaller) devices and software components.

Currently the W3C is updating the entire SSN ontology

towards a new version (2.0), which is available as a public draft document prepared by theW3C and the Open Geospa- tial Consortium (OGC)7. In this new version, a new onto- logy is introduced, namely the Sensor Observation Sampling Actuator (SOSA), absorbing a number of classes and prop- erties from SSN 1.0, such as Sensor, Observation and ob- serves. SOSA is aligned with DUL and SSN 2.0 is aligned with SOSA. SOSA is also aligned with the foundational on- tologies BFO and PROV-O8. Most important, the compat- ibility of our work with SSN 2.0 is not compromised because the specification provides alignments of SSN 2.0 with SSN 1.0, including complex ones.

3https://www.w3.org/2005/Incubator/ssn/ssnx/ssn 4http://ontologydesignpatterns.org/wiki/Ontology:

DOLCE+DnS Ultralite 5https://www.w3.org/2005/Incubator/ssn/wiki/DUL ssn

<https://www.w3.org/TR/vocab-ssn>

https://www.w3.org/2015/spatial/wiki/SOSA Ontology

## ETSI Smart Appliances REFerence (SAREF)

Recently, the European Telecommunications Standards

Institute (ETSI) along with the European Comission (EC), theNetherlands Organisation forApplied Scientific Research (TNO), the Universidad Polit´

ecnica de Madrid (UPM) and

other partners, developed the Smart Appliances REFerence (SAREF) ontology [4, 5]9. At first this ontology was built as a reference model targeting smart appliance solutions for the smart home domain10. However, SAREF has evolved to cover the IoT domain in general, being acknowledged by the EC as the ”first ontology standard in the IoT ecosys- tem, and sets a template and a base for the development of similar standards for the other verticals to unlock the full potential of IoT” [6]. The SAREF ontology provides building blocks that enable re-utilization of different parts of the ontology according to specific requirements. The new version SAREF 2.011 brings a number of changes towards this evolution, including new alignments with OneM2M for services’

provision of smart things. SAREF facilitates the matching of existing assets, since

it was developed based on standards, ontologies, data mod- els and protocols of the IoT domain, providing a high-level mapping of them (available in SAREF’s first interim study report). One of these assets is SSN, which inspired the definition of the main elements of SAREF, namely Device, Sensor, Unit of Measure and Time/Duration, according to the high-level mappings provided in the SAREF initial doc- umentation [4]. A Device (e.g. a Sensor) represents tan- gible objects designed to accomplish one or more functions in diverse types of locations (e.g. households and buildings). For example, a Sensor has Function of type Sensing func- tion. The SAREF ontology offers a list of basic functions that can be combined towards more complex functions in a single device. For example, a Switch can offer an Actuating function of type “switching on/off” and a Sensing function of type Light Sensor, so if there is illumination in the envir- onment then the switch turns off the light. Each Function has some associated Commands, which can also be picked up as building blocks from a list. For example, the “switch- ing on/off” is associated with the commands “switch on”, “switch off” and “toggle”. Depending on the Function(s) it accomplishes, a device can be found in some corresponding State(s) that are also listed as building blocks.

The composition of a Device can be represented through

the saref:consistsOf self-relationship, e.g. the WM30 wind sensor (a device) can be defined as a composition of wind direction and wind speed sensors. A Device measures a specific property, represented by the object property saref:- measuresProperty to a Property. For example, a Smoke- Sensor (Sensor) measures Smoke (Property), analogously a WindSensor measures Wind. Regarding a measurement observed by a sensor in time, SAREF represents it through the saref:makesMeasurement object property of a Device to Measurement(s), representing the relation between a device and the measurements it makes. A Measurement element links of the value of the Measurement, its Unit of Measure and the Property to which it relates. A Device offers a Service, which is a representation of a Function to a network that makes the function discoverable, registerable and remotely controllable by other devices in the network. A Service can represent one or more functions. A Service must specify the Device that offers the Service, the function(s) to be represented, and the (input and output) parameters necessary to operate the service, supported by the ontology alignments with OneM2M ontology.

## HL7 aECG

This use case

## HL7 FHIR

Wrong way = opposite direction .

## UFO ECG

Wrong way = opposite direction .

## LogiCO

Wrong way = opposite direction .

## LogiTrans

Wrong way = opposite direction .

## LogiServ

Wrong way = opposite direction .

## OASIS EDXL

## W3C EIIF

Wrong way = opposite direction .

## Others related to emergency management

Wrong way = opposite direction .

# Ontology alignments and extensions

## INTER-IoT approach (IPSM)

In the scenario of detecting emergencies at the port by monitoring drivers’ vital signs withmedical wearable devices, semantic integration is required between IoT arti- facts based on SSN and smart appliances based on SAREF, e.g. building sensors for vehicle collision detection, security and electrical systems. INTER-IoT is currently developing the Inter-Platform Semantic Mediator (IPSM) tool. IPSM is a software tool that follows the semantic interoperability design patterns identified in INTER-IoT [7], and is intended to be used as part of the translation process defined in the methodology (INTER-Meth). The process of achieving se- mantic interoperability involves the following steps: (i) lift semantics to a common format and make it explicit; (ii) develop, or choose, a central modular ontology; (iii) pre- pare uni-directional alignments between ontologies of com- municating artifacts and the central ontology; (iv) establish a multi-channel (1-1, 1-many, many-1) communication ar- chitecture to facilitate translations in all needed contexts, with the central ontology as intermediary.

INTER-IoT provides its own alignment format, based on

an alignment API with a declarative ontology alignment lan- guage (XML-based), inspired on EDOAL2, to represent in- terconnections between semantic data ofmultiple ontologies. IPSM utilizes alignment files and provides a multi-channel environment for any artifact. Pairs of uni-directional align- ments between the central ontology and artifact ontology are used to translate messages to and from the central on- tology. This enables connection of new artifacts without jeopardizing the existing channels, and requires each par- ticipant to provide only a pair of alignments. While com- plete ontologies are used to build semantic understanding, only conversation-specific alignments are stored and used for actual translations. Ontology alignments and transla- tion channels can be managed through the REST Manager. Because of space limitations in this paper we omit other re- lated work, which can be found in our prior publications that cover the state-of-the-art in this area [7, 8, 9, 10].

## SSN x SAREF

According to themethodology used, the mappings between

SSN and SAREF were specified through an ontological ana- lysis of their TBox, i.e. concepts and roles definitions (pre- dicates) with logical operations. A study was made on how SSN and SAREF describe the characteristics of sensors, in- cluding their capabilities of observation. The mappings fol- low a logic sequence according to the main elements and similar structures of SSN and SAREF. Here we detail only the mappings from SSN to SAREF as a first step towards the creation of the bi-directional translations. For eachmap- ping a code snippet is presented as a pseudo-code to illus- trate the algorithm for the creation of the SAREF-based ontology. This pseudo-code include an IN representing the input SSN element(s). When creating a new SAREF class or property, var is used and createTriple function creates a triple (class, object property, class).

M01. ssn:SensingDevice -> saref:Sensor

While the main element of SSN is the Sensing Device, a sub- class of Device and Sensor, in SAREF the main element is Device, which can be specialized as a Sensor related to a Sensing Function. The characteristics of ssn:SensingDevice, inherited from ssn:Sensor and ssn:System, are mapped to saref:Sensor, inheriting saref:Device properties, including the relationship with the saref:SensingFunction (saref:has- Function). Figure 2 illustrates the elements involved in this mapping. Notice that, indirectly, this mapping also trans- forms from ssn:Sensor to saref:Sensor if the ssn:Sensor is a ssn:SensingDevice. All RDFS properties are copied from ssn:SensingDevice to saref:Sensor.

Listing 1: Pseudocode snippet for M01

M02. ssn:hasSubSystem -> saref:consistsOf

Listing 2: Pseudocode snippet for M02

After executing M01, M02 checks the composition relation- ship of a device, i.e. the components that are part of a device. In SSN, the object property ssn:hasSubSystem relat- ing two ssn:System represents this relationship. In SAREF, the object property saref:consistsOf plays this role, relating two saref:Device in a similar way, both illustrated in Figure 2 as a self-relationship. Therefore, when ssn:hasSubSystem is used between the ssn sensingDevice (from M01) and a ssn:- System, which must be a ssn:Device or a ssn:SensingDevice, then it is mapped to saref:consistsOf object property of the saref sensor (created in M01). If the device component is a ssn:SensingDevice, then a recursive algorithm is used by applying M01 to it. If the device component is a ssn:Device, then it is created a saref:Device.

M03. ssn:observes -> saref:measuresProperty After executing M01 and M02, M03 maps the measurement property which the sensor is able to observe. For example, a wind sensor is able to observe both wind direction and wind speed. Therefore, the ssn:observes of a ssn:SensingDevice is mapped to saref:measuresProperty of a saref:Device. These object properties relate to ssn:Property and saref:Property, respectively. Therefore, this mapping also includes the cre- ation, if it does not exist, of the ssn:Property. At last, this mapping needs to create the relationship back from the saref:Property to the saref:Device through the object prop- erty saref:isMeasuredByDevice. The code snippet below il- lustrates this mapping.

Listing 3: Pseudocode snippet for M03

### Issues identified

The main issue identi- fied is the lack of a naive element in SAREF to describe the measurement capabilities of a sensor, which SSN enables through the ssn:hasMeasurementCapability object property of ssn:Sensor. To cope with this issue we suggest that a new mapping is created to align the structure from SSN, i.e. create the object property ssn:hasMeasurementCapability on saref:Sensor with the restriction of only ssn:MeasurementCa- pability. In addition, the mapping must consider to align both ssn:MeasurementCapability and ssn:MeasurementPro- perty as (is-a) saref:Property. This would enable the link of a saref:Sensor to the ssn:MeasurementCapability, which incorporates the links to the ssn:MeasurementProperty. A conceptual issue in SSN was identified regarding the

element ssn:Sensor. The description of this element states that it “allows sensors, methods, instruments, systems, al- gorithms and process chains as the process used of an ob- servation (. . . ) they are all grouped under the term sensor”. Thus, the description includes that a ssn:Sensor can be a ”system”, but ssn:Sensor was not implemented as a special- ization of ssn:System. In this way, ssn:Sensor could also in- herit the composition relationship (ssn:hasSubSystem) and, thus, can represent a set of sensors. Issues identifie

Issues identified in WM30 ontology include: (i) theWM30:Vaisala WM30 com- position of wind direction (WM30:WM30 WindDirection) and speed (WM30:WM30 WindSpeed) sensors. Both are ssn:Sensor, but the composition relationship (ssn:hasSubSy- stem) is applied from a ssn:System to a ssn:System. There- fore, the M02 mapping must not consider the types of the subject or the object of the ssn:hasSubsystem property. (ii) the universal quantifier on ssn:forProperty (Figure 4) is wrong. A practical issue when mapping to SAREF is to con-

sider extending the taxonomy of sensor ”types” by creating a new element when the type does not exist in SAREF. For example, smoke and temperature sensors are classified as saref:SmokeSensor and saref:TemperatureSensor, respectively, having function (saref:hasFunction) and measure property (saref:measuresProperty), linking the type of the sensor with functions it has and the type of property it measures (saref:- Smoke and saref:Temperature. Thus, in our example, the correct implementation for the wind sensor is to create the subclass saref:WindSensor, with function sensing and meas- uring the properties of wind direction and wind speed, which is guaranteed by M01. A

### SSN 2.0 and SOSA

An issue that must be ad- dressed is revisiting these mappings when SSN 2.0 is pub- lished, and decide whether the mappings will be updated or simply use the ontology alignments provided by SSN 2.0 specification. It includes the description of complex align- ments which are presented with the property-chain axioms, as the alignment of oldssn:observes (property used in our third mapping) to the equivalent sosa:observes with chain axioms oldssn:hasMeasurementCapability oldssn:forProperty and oldssn:madeObservation oldssn:observedProperty. In ei- ther ways this work will still be applicable and will not be- come obsolete. Although this work only covers three vocab- ulary terms from the SSN 1.0, here we address the main elements used to characterize sensors, thus, providing a sig- nificant contribution for the state of the art.

### Alignment implementation and sample data

SAREF to SSN/SOSA:

<?xml version="1.0" encoding="utf-8" standalone="no"?>

<!DOCTYPE Alignment

[

<!ENTITY sripas "http://www.inter-iot.eu/sripas#">

<!ENTITY sosa "http://www.w3.org/ns/sosa/">

<!ENTITY sosaExt "http://www.w3.org/ns/sosa/extension/">

<!ENTITY geo-sf "http://www.opengis.net/def/sf/">

<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">

<!ENTITY saref "https://w3id.org/saref#">

<!ENTITY sarefInst "https://w3id.org/saref/instances/">

<!ENTITY time "http://www.w3.org/2006/time#">

]

>

<Alignment

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:sripas="http://www.inter-iot.eu/sripas#"

xmlns="http://www.inter-iot.eu/sripas#"

xmlns:sosa="http://www.w3.org/ns/sosa/"

xmlns:saref="https://w3id.org/saref#"

xmlns:time="http://www.w3.org/2006/time#"

name="SAREF\_CO" version="1.0" creator="sripas">

<onto1>

<Ontology about="https://w3id.org/saref#">

<formalism>

<Formalism name="OWL2.0" uri="http://www.w3.org/2002/07/owl#"/>

</formalism>

</Ontology>

</onto1>

<onto2>

<Ontology about="http://www.w3.org/ns/sosa/">

<formalism>

<Formalism name="OWL2.0" uri="http://www.w3.org/2002/07/owl#"/>

</formalism>

</Ontology>

</onto2>

<steps>

<step order="1" cell="cell1"/>

<step order="2" cell="cell2"/>

<step order="3" cell="cell3"/>

</steps>

<map>

<Cell id="cell1">

<entity1>

<sripas:node\_CTA>

<rdf:type rdf:resource="&saref;Device"/>

<saref:makesMeasurement>

<sripas:node\_CTB>

<saref:relatesToProperty>

<sripas:node\_CTC/>

</saref:relatesToProperty>

<saref:hasValue>

<sripas:node\_CTV/>

</saref:hasValue>

<saref:hasTimestamp>

<sripas:node\_CTT/>

</saref:hasTimestamp>

</sripas:node\_CTB>

</saref:makesMeasurement>

</sripas:node\_CTA>

</entity1>

<entity2>

<sripas:node\_CTB>

<rdf:type rdf:resource="&sosa;Observation"/>

<sosa:madeBySensor>

<sripas:node\_CTA>

<rdf:type rdf:resource="&sosa;Sensor"/>

</sripas:node\_CTA>

</sosa:madeBySensor>

<sosa:hasSimpleResult>

<sripas:node\_CTV/>

</sosa:hasSimpleResult>

<sosa:observedProperty>

<sripas:node\_CTC/>

</sosa:observedProperty>

<sosa:phenomenonTime>

<rdf:Description>

<rdf:type rdf:resource="&time;Instant"/>

<time:inTimePosition>

<rdf:Description>

<rdf:type rdf:resource="&time;TimePosition" />

<time:numericPosition>

<sripas:node\_CTT/>

</time:numericPosition>

</rdf:Description>

</time:inTimePosition>

</rdf:Description>

</sosa:phenomenonTime>

</sripas:node\_CTB>

</entity2>

<relation>=</relation>

</Cell>

<Cell id="cell2">

<entity1>

<sripas:node\_CTA>

<rdf:type rdf:resource="&saref;SpeedMeasurement"/>

</sripas:node\_CTA>

</entity1>

<entity2>

<sripas:node\_CTA>

<rdf:type rdf:resource="&sosaExt;SpeedMeasurement"/>

</sripas:node\_CTA>

</entity2>

<relation>=</relation>

</Cell>

<Cell id="cell3">

<entity1>

<sripas:node\_CTA>

<rdf:type rdf:resource="&saref;AccelarationMeasurement"/>

</sripas:node\_CTA>

</entity1>

<entity2>

<sripas:node\_CTA>

<rdf:type rdf:resource="&sosaExt;AccelerationMeasurement"/>

</sripas:node\_CTA>

</entity2>

<relation>=</relation>

</Cell>

</map>

</Alignment>

SAERF sample data (INTER-IoT syntax):

{

"@graph":[

{

"@graph":[

{

"@id":"InterIoTMsg:meta66b05c61-d687-45a3-b5fb-6864bbec3b69",

"@type":[

"InterIoTMsg:Thing\_update",

"InterIoTMsg:meta"

],

"InterIoTMsg:conversationID":"conv99528eba-eb2d-47e8-9ee6-9dd40d19f89a",

"InterIoTMsg:dateTimeStamp":"2017-05-22T22:19:30.281+02:00",

"InterIoTMsg:messageID":"msg7e484a2c-f959-486e-8da0-31143f457234"

}

],

"@id":"InterIoTMsg:metadata"

},

{

"@graph": [

{

"@id": "sarefInst:exampleSmartPhoneSendingInfoTruck",

"@type": "saref:Device",

"@label": {

"@language": "en",

"@value": "Motorola Moto G5 Plus"

},

"geo:location": {

"@id": "sarefInst:test.1.1.LocationSmartPhone\_39.431478658043424\_-0.35860926434736484",

"@type": [

"owl:NamedIndividual",

"geo:SpatialThing"

],

"@label": {

"@language": "en",

"@value": "Location of the smartphone, should be the same location of the truck (?)"

},

"geo:latitude": 39.431478658043424,

"geo:longitude": -0.35860926434736484

},

"saref:makesMeasurement": [

{

"@id": "sarefInst:SpeedMeasurement\_\_Test.1.1\_1511466006.9682777",

"@type": "saref:SpeedMeasurement",

"@label": "Example of a speed measurement observed by a mobile device",

"saref:hasTimestamp": 1511466006.9682777,

"saref:hasValue": 14,

"saref:isMeasuredIn": "saref:SpeedUnit\_MeterPerSecond",

"saref:relatesToProperty": "saref:VelocityOrSpeed\_Vehicle"

},

{

"@id": "sarefInst:AccelerationMeasurement\_Test.1.1\_1511466006.9682777",

"@type": "saref:AccelarationMeasurement",

"@label": "Example of accelaration measurement observed by a mobile device",

"saref:hasTimestamp": 1511466006.9682777,

"saref:hasValue": 1.4,

"saref:isMeasuredIn": "saref:AccelerationUnit\_MeterPerSecondSquared",

"saref:relatesToProperty": "saref:Accelaration\_Vehicle"

}

]

}

],

"@id": "InterIoTMsg:payload"

}]

,

"@context":{

"InterIoTMsg":"http://inter-iot.eu/message/",

"InterIoT":"http://inter-iot.eu/",

"sarefInst" : "https://w3id.org/saref/instances/",

"schema":"http://schema.org/",

"qu":"http://purl.oclc.org/NET/ssnx/qu/qu#",

"owl":"http://www.w3.org/2002/07/owl#",

"saref":"https://w3id.org/saref#",

"xsd":"http://www.w3.org/2001/XMLSchema#",

"skos":"http://www.w3.org/2004/02/skos/core#",

"dim":"http://purl.oclc.org/NET/ssnx/qu/dim#",

"rdfs":"http://www.w3.org/2000/01/rdf-schema#",

"dct":"http://purl.org/dc/terms/",

"rdf":"http://www.w3.org/1999/02/22-rdf-syntax-ns#",

"xml":"http://www.w3.org/XML/1998/namespace",

"dcterms":"http://purl.org/dc/terms/",

"time":"http://www.w3.org/2006/time#",

"foaf":"http://xmlns.com/foaf/0.1/",

"om":"http://www.wurvoc.org/vocabularies/om-1.8/",

"geo":"http://www.w3.org/2003/01/geo/wgs84\_pos#"

}

}

## SAREF extension for health data (ECG)

Wrong way = opposite direction .

## SAREF extension for logistics data (transportation)

Wrong way = opposite direction .

## SAREF x EDXL

Wrong way = opposite direction .

## Other recommendations

Wrong way = opposite direction .

# References

1. Ganzha, M., et al., *Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective.* Journal of Network and Computer Applications, 2016.

2. INTER-IoT, *INTER-IoT deliverable: D2.4. Use cases manual*. 2016.

3. Moreira, J.L.R., et al. *Ontology-driven Conceptual Modeling for Early Warning Systems: Redesigning the Situation Modeling Language*. in *International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. 2017.

4. Moreira, J.L.R., et al., *Towards ontology-driven situation-aware disaster management.* Journal of applied ontology, 2015. **10**(3-4): p. 339-353.

5. INTER-IoT, *INTER-IoT deliverable: D4.1. Initial Reference IoT Platform Meta-Architecture and Meta Data Model (15/01/2017)*. 2017.

6. INTER-IoT, *INTER-IoT deliverable: D2.3. Requirements and business analysis*. 2016.

7. Fortino, G., et al., *BodyCloud: A SaaS approach for community Body Sensor Networks.* Future Generation Computer Systems, 2014. **35**: p. 62-79.

8. Wieringa, R., *Design Science Methodology for Information Systems and Software Engineering.* Springer Berlin Heidelberg, 2014: p. 493.

9. Mucheru, G., *Operation of Gyro sensor and 3-axis Accelerometer*. 2014, Helsinki Metropolia University of Applied Sciences.

10. Thompson, C., et al., *Using Smartphones to Detect Car Accidents and Provide Situational Awareness to Emergency Responders*, in *Mobile Wireless Middleware, Operating Systems, and Applications: Third International Conference, Mobilware 2010, Chicago, IL, USA, June 30 - July 2, 2010. Revised Selected Papers*, Y. Cai, et al., Editors. 2010, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 29-42.

11. Costa, P.D., et al., *Rule-Based Support for Situation Management*, in *Fusion Methodologies in Crisis Management: Higher Level Fusion and Decision Making*, G. Rogova and P. Scott, Editors. 2016, Springer International Publishing: Cham. p. 341-364.

12. Mane, A. and J. Rana. *Vehicle Collision detection and Remote Alarm Device using Arduino*. 2014.

13. San Vicente Foronda, L., *Automotive Events Detection using MEMS Accelerometers.* 2009: p. 129-129.

14. Sulochana, B. and B.A.S.M. Babu, *Monitoring and Detecting Vehicle Based On Accelerometer and MEMS Using GSM and GPS Technologies.* International Journal of Computer Science Trends and Technology (IJCST), 2014. **2**(4): p. 55-59.

15. Ibrahim, H.A., A.K. Aly, and B.H. Far. *A system for vehicle collision and rollover detection*. in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2016.

16. MIMAR, T., *Vehicle security with accident notification and embedded driver analytics*. 2016, Google Patents.

17. Nandaniya, K., V. Choksi, and M.B. Potdar, *Microcontroller Based Collision Detection and Warning System.* International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2014. **3**(5): p. 9565-9570.

18. Fortino, G. and R. Gravina, *Fall-MobileGuard: a smart real-time fall detection system*, in *Proceedings of the 10th EAI International Conference on Body Area Networks*. 2015, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Sydney, New South Wales, Australia. p. 44-50.

19. Shimmer, *Shimmer User Manual Revision 3p*. 2017, Realtime Technologies Ltd.

20. Gravina, R. and G. Fortino, *Automatic Methods for the Detection of Accelerative Cardiac Defense Response.* IEEE Transactions on Affective Computing, 2016. **7**(3): p. 286-298.

21. Barfod, C., et al., *Abnormal vital signs are strong predictors for intensive care unit admission and in-hospital mortality in adults triaged in the emergency department - a prospective cohort study.* Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine, 2012. **20**(1): p. 28.