

# PROMES: An Ontology-Based Messaging Service for Semantically Interoperable Information Exchange During Disaster Response

Linda Elmhadhbi<sup>1</sup>, Mohamed-Hedi Karray<sup>1</sup>, Bernard Archimède<sup>1</sup>, J. Neil Otte<sup>2</sup>, and Barry Smith<sup>3</sup>

<sup>1</sup>University of Toulouse, INP-ENIT, Tarbes 65000, France

<sup>2</sup>Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland 20707, USA

<sup>3</sup>University at Buffalo, New York 14203, USA

[linda.elmhadhbi@enit.fr](mailto:linda.elmhadhbi@enit.fr)

## Abstract

Disaster response requires the cooperation of multiple emergency responder organizations (EROs). However, after-action reports relating to large-scale disasters identify communication difficulties among EROs as a major hindrance to collaboration. On the one hand, the use of two-radio communication, based on multiple orthogonal frequencies and uneven coverage, has been shown to degrade inter-organization communication. On the other hand, because they reflect different areas of expertise, EROs use differing terminologies, which are difficult to reconcile. These issues lead to ambiguities, misunderstandings, and inefficient exchange of data and information among those involved, which can impede the response process and slow decision making. We, therefore, hypothesize that promoting semantic interoperability across ERO information systems might improve information exchange among stakeholders and thereby allow a more coherent response to the disaster. We propose an ontology-based messaging service on the basis of the Emergency Data Exchange Language (EDXL) standards. The parties involved will continue to use the terminologies to which they are accustomed, but the system will resolve inconsistencies and thereby enhance mutual understanding among EROs by ensuring semantic translation of the exchanged information. The evaluation of the semantic translation demonstrated the effectiveness and accuracy of the proposed service.

**Keywords:** disaster response, semantic interoperability, information exchange, messaging service, ontology, emergency responder organization.

## I. Introduction

A multi-agency disaster response process requires the cooperation and coordination of multiple emergency responder organizations (EROs). EROs thus require timely, accurate and relevant information to support resource allocation, casualty prioritization, and other key subprocesses of disaster response. However, numerous after-action reports from major disasters have cited communication difficulties among EROs as a major challenge (Clarke, 2003). An example can be found in the concluding report on the terror attack in Norway on June 22, 2011, which states that the various EROs (including firefighters, police, healthcare services, and

others) were unable to communicate effectively and coordinate their efforts. These challenges were highlighted also by the 9/11 (Kean and Hamilton, 2004) and 11/13 Paris terrorist attacks. Even though almost fourteen years passed between these two terror attacks, the same response deficits appear in both. In the 11/13 Paris attacks, for example, there were two sites where victims did not receive medical care due to a lack of communication between firefighters and healthcare units (Philippe et al., 2016). In addition, police forces claimed that:

- “by the time the information gets out and finds its way up to the central organization, mobilizing the specialized units takes a relatively long time.”
- “our police are not organized along local lines. Everything has to filter up to the central organization at the prefecture.”
- “We have a police force that is disconnected from the field.” (Nossiter, 2015)

Indeed, the use of radio to connect individual actors – with each ERO using its own frequency – contributes both to the lack of cross-ERO interoperability and to the practical difficulty of inter-organizational communication (Manoj & Baker, 2007). To make things worse, operational actors find themselves all too often with poor radio coverage. In a recent survey of EROs (in Building Public Safety Communication Survey, 2018), more than 65% of emergency responders (ERs) said they had experienced some sort of communication failure within the past twenty-four months while responding to an emergency. Moreover, in the Paris attacks, firefighters and healthcare units pointed out that the use of radio communication means such as the ANTARES network was unsatisfactory during the different interventions (Boutinaud, 2017). An additional problem is that radio communication typically does not enable information to be analyzed or stored for review.

A further problem is that each ERO deploys an information system adapted to its own needs, resources, and processes. Associated information is therefore heterogeneously stored, using different formats and relating to disparate data sources. Moreover, because each ERO has its own specific area of expertise, vocabularies and terminologies, too, are heterogeneous. The resultant semantic heterogeneity of data and the absence of a common language are becoming ever more important issues as the amount of available data grows (Kristiansen et al., 2019, Bharosa et al., 2010).

The semantic heterogeneity of data leads to very serious issues when these data need to be combined. One word may be interpreted differently from one context to another. Take the word “tank” as example, in an armored vehicle context, the term refers to a certain kind of specialized armored vehicle, but in a firefighter context, it refers to a type of container used for holding water. Where information needs to be exchanged between persons working in these two contexts, it is not evident how the expression “we need a tank immediately” should be interpreted (Bittner et al., 2005).

To overcome these issues, we address in what follows how semantic interoperability problems can be resolved in such a way as to empower data exchange among first responders including firefighters, police forces, gendarmerie, healthcare units, and public authorities. Accordingly, we propose an ontology-based messaging service that resolves terminology inconsistencies and ensures mutual understanding among stakeholders. It is developed within the scope of POLARISC project (Elmhadhbi et al., 2018) (see Figure 1). Our goal is to find a way to achieve the sharing of semantically unambiguous information across different EROs through a service we call PROMES (for ‘POLARISC Ontology-Based Operational Messaging Service’) based on the Emergency Data Exchange Language (EDXL) suite of standards (Gusty and Dwarkanath, 2010). These latter are common standards, initiated by the US Department of

Homeland Security and developed by the Organization for the Advancement of Structured Information Standards (OASIS) and have been used in multiple disaster management applications including the POLARISCO (POLARISC Ontology) (Elmhadhbi et al., 2019). EDXL is a collection of messaging standards designed to facilitate emergency information sharing and data exchange across EROs.

In this work, we describe the functioning of PROMES for use in information exchange among first responders. We present the exploitation of POLARISCO that defines ER terminologies and details some of its classes and relations. We also provide the evaluation and validation of the proposed service.

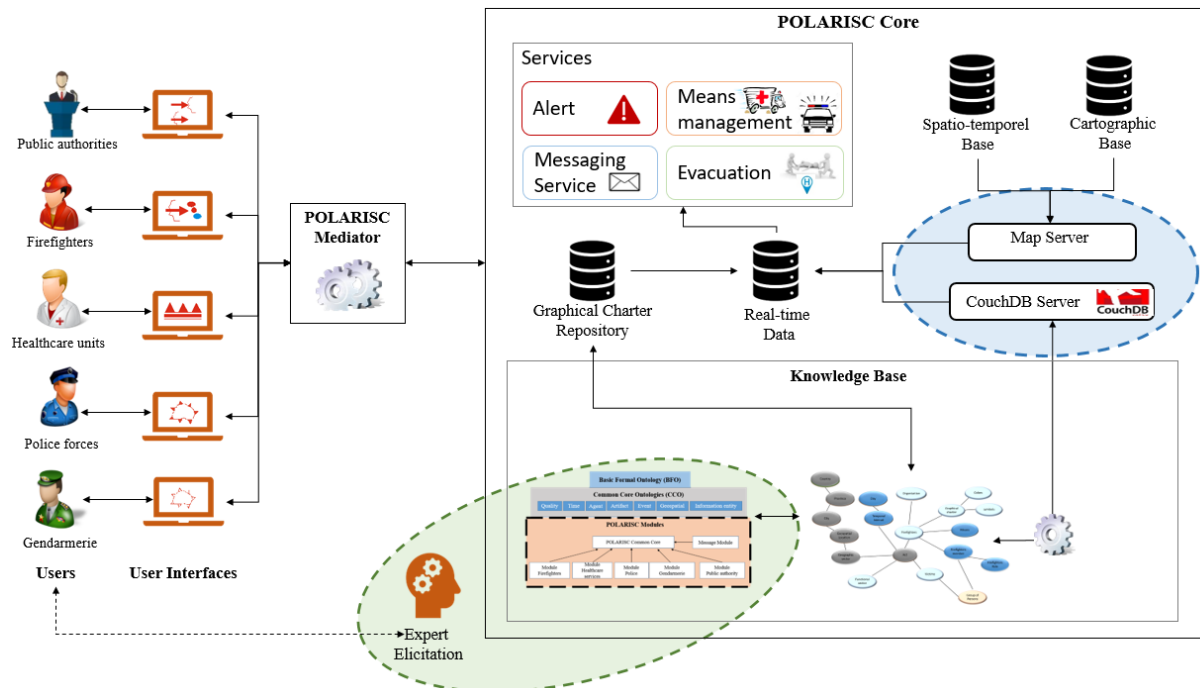


Figure 1. POLARISC architecture

## II. Related work

Information exchange across ERs have posed a long-standing challenge. (Bhattacharjee, 2016) proposes an Android mobile phone disaster messenger application that enables situational information sharing in the absence of a network infrastructure. However, this solution is oriented around use by volunteers for purposes of disseminating post-disaster information.

(Meissner et al., 2006) propose an integrated mobile information and communication system called MIKoBOS (Mobile Information and Communication System for Public Safety Organizations) which enables data communication among stakeholders during emergency response operations by integrating the operations of different types of mobile terminals, communication technologies, and advanced satellite communication.

Another approach for ensuring data exchange is presented in (Chipara et al., 2011). This concerns an information system designed to deliver data by using a client-server architecture and an ad hoc routing protocol. In reality, this system has low reliability when deployed.

The studied projects are intended to improve communication among ERs by proposing technological solutions. They do not consider the semantics of the exchanged data.

To overcome semantic heterogeneity and to guarantee a consistent shared understanding of information, the use of ontologies is crucial (Bittner et al., 2005, Arp et al., 2015, Smith et al., 2009). Ontologies serve both as knowledge representation and as mediation to enable heterogeneous systems interoperability.

Multiple approaches have been advanced to semantically translate information expressed in different terminologies in an ontology framework (Shvaiko and Euzenat, 2011).

An automatic model transformation methodology is proposed in (Wang et al, 2015). It combines semantic and syntactic checking measurements into the model transformation process. To do so, a semantic thesaurus has been created on the basis of WordNet. The mapping between source and target models is automatically done basing on an approximate value generated between two words. However, the chosen semantic meaning may not be exact to the word within a specific context of the source model. WordNet is not sufficient to be used in specific terminology such as ERs terminologies.

Bicer et al. (2005) proposed AMEF (Artemis Message Exchange Framework) that provides a semantic mediation among healthcare institutes. This mediation is done through a mapping tool that produces a mapping definition to transform a source ontology into a target ontology. For this purpose, an OWL ontology mapping tool was developed in order to allow semantic mappings among distinct ontologies. Then, the mapping definition is used by AMEF to automatically transform the source ontology message instances into target message instances. However, this engineering approach is only applicable in case of information exchange across healthcare institutes.

Real et al. (2017) formalized domain-specific terminologies from the UK Civil and Protection Terminology lexicon. They gathered some of the most common terms that UK agencies use in emergency response scenarios and developed an extension for WordNet. Then, they proposed a domain-aware semantic matching. The aim was to match words with similar meanings from various sources.

In the case of military and first responder command and control applications, OntoNet is a platform for connecting sensors, services, and agents on the network (Nguyen et al., 2010). It proposes a knowledge-based approach to message addressing and matching. In fact, the focus of OntoNet is on effectively matching messages and receivers and not on the semantic meaning of the message's content.

In the context of INTER-IoT (Interoperability of Heterogeneous IoT Platforms) project, SEMIOTICS (SEmantic Model-driven development for IoT Interoperability of emergenCy serviceS) aims to detect accident risks with trucks that deliver goods at the Valencia port area. To do so, Moreira et al. (2017) present an interoperable framework architecture for the integration of different IoT architectures. Authors propose the Inter-Platform Semantic Mediator (IPSM) software tool that enables real-time semantic translations following five steps: make semantics explicit, define a central modular ontology, define uni-directional alignments between the central ontology and ontologies of communicating artifacts, and establish communication architecture in order to facilitate translations between ontologies. The mapping between Semantic Sensor Network (SSN) and Smart Appliances REFERENCE (SAREF) ontologies follows a logical sequence of ontological analysis of their TBox to create a new SAREF-based ontology. Authors start by the specification of the possible mapping and rules in natural language to show how an instance of the source ontology can be represented with the target ontology, and then they implement these mappings. IPSM create a SPARQL query for each rule in order to find instances and generating new ontology instances.

If we project this work in our context, the automatic mapping between ontologies cannot be used to define equivalences between stakeholders' knowledge. The semantic translation of ERs exchanged information is a very delicate process as disaster response deals with human lives. It necessitates an accurate mapping between stakeholders' concepts to guarantee the exact meaning of the exchanged information. Accordingly, we propose a semi-automatic process that starts by defining manually the semantic mapping between stakeholders' ontological modules by referring to emergency experts, then the semantic translation of the message is done automatically based on the defined mapping.

Accordingly, PROMES exploits POLARISCO to promote semantic interoperability across the different stakeholders. The parties involved will continue to use the terminologies to which they are accustomed, but the system will resolve inconsistencies and thereby enhance mutual understanding by enabling a semantic translation among ERs. PROMES design will be guided by the following requirements:

- The information system should, under the hood, as it were – most users will not be aware of its operations – semantically translate information into some unambiguous expression in such a way as to ensure share meanings.
- The semantic translation process should not amplify the processing time.
- The semantic translation process should be as most accurate as possible.

### **III. PROMES: POLARISC Ontology-Based Operational Messaging Service**

#### **1. PROMES architecture**

The messaging service PROMES is responsible for ensuring semantically enhanced information exchange among ERs. The main purpose of PROMES is that each stakeholder will receive the message according to his own vocabulary and with his own semantics. The semantic translation of the message content is based on the semantic relationships that exist among stakeholders' ontological modules as defined in POLARISCO. PROMES is used on-demand by POLARISC mediator, which is connected to POLARISC platform, to perform the semantic interpretation of the information to be exchanged (see Figure 2). The messaging service process is divided into four steps; the message editing, textual transformation, message validation, and then semantic translation of the message. Hence, the transformation of the message is twofold; a textual transformation and a semantic transformation. Therefore, we developed two algorithms, (i) the message's textual transformation algorithm (TT) and (ii) the message's semantic transformation (ST) algorithm.

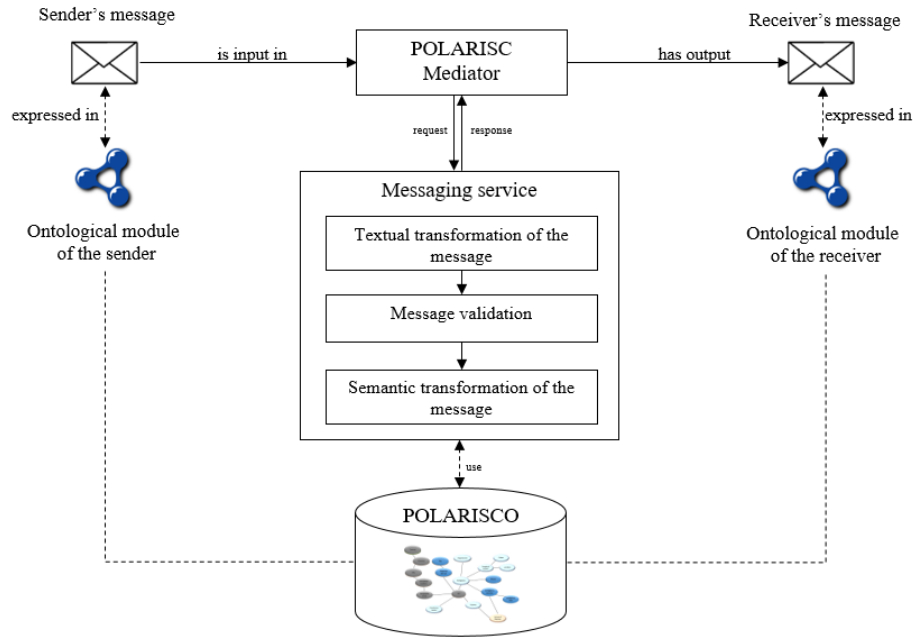


Figure 2. PROMES architecture

As seen in the previous section, there are various types of messages to be exchanged among ERs at the operational level. Concerning the French EROs, there is a predefined structure to compose the message in a succinct, but clear way by providing a means for standardizing these latter in order to guarantee the clarity of the information. After understanding the structure of each message, we developed a graphical user interface that respects the predefined structures (see section VI). Once the type of the message is chosen, the user is guided to structure the message class by class. These classes proposed by the graphical interface are loaded from POLARISCO. In such a way, the message editing process takes less time since it is selected class by class and not manually written. Moreover, this oriented way of message's edition enables the effectiveness of the semantic translation of the message since the inputted classes are loaded from the ontology.

At this level, the result from the edition step is a set of classes chosen by the sender. The next step is the textual transformation of the message from a set of classes to a textual message using an algorithm based on the pre-defined structure of the exchanged information of each ERO. Once the user validates the proposed textual transformation, the mediator checks if the sender and the receiver share the same ontological module. If this is not the case, the mediator uses the messaging service to semantically transform the message according to the vocabulary of the receiver. Hence, it transforms information expressed according to the ontological module of the sender into equivalent information defined using the ontological module of the receiver. Through a dedicated algorithm. Figure 3 shows the interactions sequence among these different components.

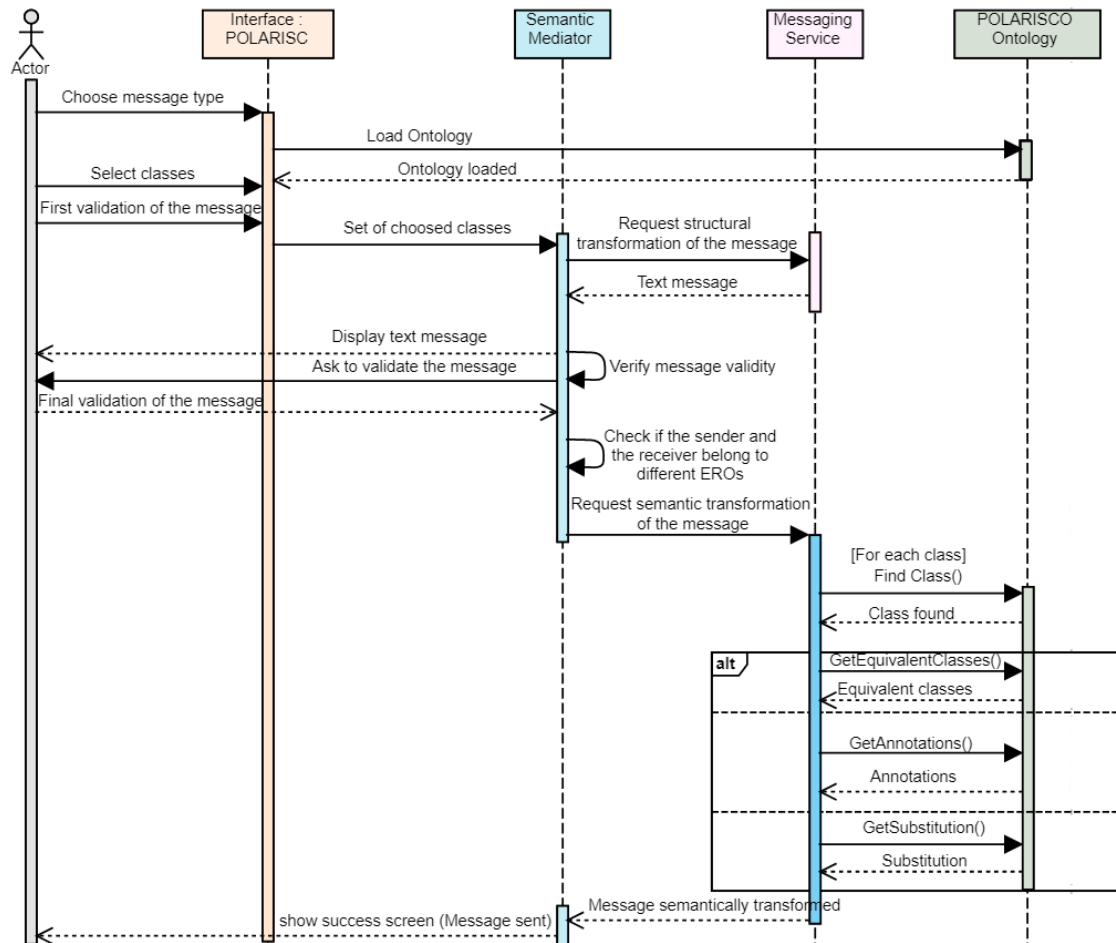


Figure 3. Sequence diagram of the message transformation process

## 2. POLARISCO ontology

We use the French disaster response doctrine as our starting point, with the intention of creating a general framework, which can be specified to match the counterpart doctrines in other countries at a later exchange. The terminologies used by the different French ERs were analyzed to design a common shared vocabulary<sup>1</sup>. This was then used to design a suite of modular ontologies, named POLARISCO (for POLARISC Ontology)<sup>2</sup>, in order to semantically capture the knowledge of the different stakeholders of the commandment level and as well as the operational level of disaster response. More specifically:

- POLARISCO defines the different kinds of disasters (disaster's type, needed resources, the corresponding acts, etc.).
- POLARISCO defines the vocabulary of each actor (firefighters, police, gendarmerie, healthcare units, and public authorities) including processes of intervention, means, roles, and so on. Furthermore, disaster response is about taking care of victims. Accordingly, we define victims and their different states.
- A disaster occurs in a specific spatial-temporal region. These latter are fundamental to effectively respond to a disaster. For this purpose, POLARISCO provides a temporal and geospatial description of disasters and stakeholders' actions.

<sup>1</sup> There is a French and an English version of the vocabulary

<sup>2</sup> <https://github.com/LindaElmhadhbi/POLARISCO>

To take account of the diversity of ER vocabularies we adopted the principles of modularization to build our ontology. Thus, we proposed five modules to represent the knowledge of the different involved ERs namely firefighters module, healthcare units module, police module, gendarmerie module, and public authorities module. After that, we built a Glossary of Terms (GT) for each module by referring to the knowledge elucidated during the acquisition step. Terms include classes, properties, instances, and relations. We found that there are several terms in common among the stakeholders' modules, which led us to define a core module named PCC (POLARISCO Common Core). It includes the general classes that all stakeholders share (e.g. disasters, transmission means, victims) in order to ensure more semantic interoperability among the modules and to facilitate their integration. Afterward, we defined a message module that formalizes acts of communication between stakeholders, and a healthcare resources module that defines victims' diseases and the associated staffing and equipment. To summarize, to formalize the ERs' knowledge, POLARISCO is an extension of BFO and CCO<sup>3</sup> that integrates eight modules. Figure 4 illustrates the proposed modules and their import structure.

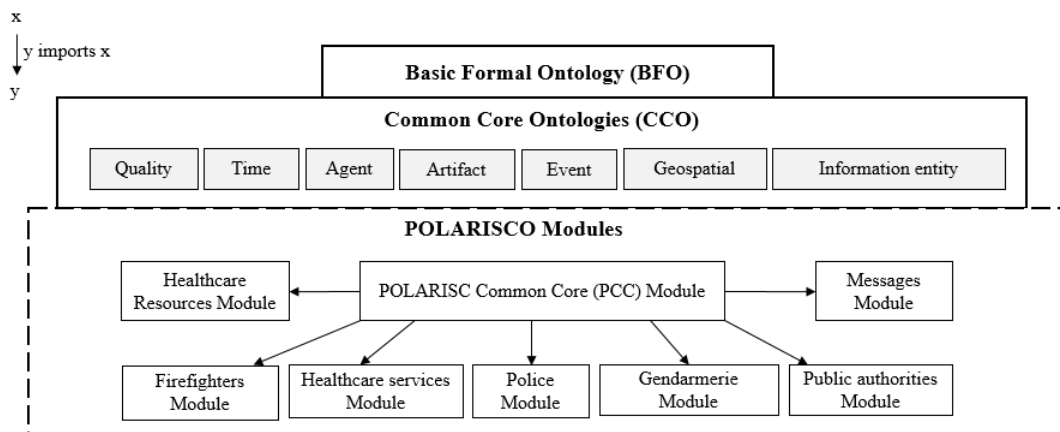


Figure 4. POLARISCO modules

To define the message module, we used the EDXL standard as the basis for ontology development. EDXL has been applied already in the EDXL-RESCUER ontology and PS/EM ontology. EDXL-RESCUER ontology is the conceptual model of the RESCUER project that represents information exchange among legacy systems for emergency and crisis management (Barros, 2015). It is based on EDXL-DE and EDXL-CAP for the creation of alerts addressed to persons affected by a disaster. It defines mainly message type, status, scope, sender, certainty, severity, and urgency. The EDXL-RESCUER ontology is not yet used in the real world.

PS/EM ontology is used to provide a foundation for semantic interoperability between different PS/EM communication systems (Chan et al., 2017). It is developed basing on the four EDXL standards; EDXL-DE, EDXL-RM, EDXL-HAVE, and EDXL-CAP. Specifically, PS/EM defines the different types of messages; alert message, alert acknowledgment message and alert rejection message. Furthermore, it defines incident response activities such as evacuation and finding shelters. In fact, EDXL-RESCUER and PS/EM ontologies focus only on alert messages and do not cover other types of communication among ERs.

In the message module, we reused classes from the PS/EM ontology rather than EDXL RESCUER because of the fact that PS/EM ontology was grounded in the BFO upper-level ontology. Accordingly, POLARISCO and PS/EM ontology follow the same vision for

<sup>3</sup> except units of measure ontology and currency unit ontology



formalizing what exists. The reused classes are marked in the following figures with the prefix EDXL.

We define an “edxl: message” as a “cco: information bearing artifact” (see Figure 5). Moreover, a message has a sender and a receiver. It can be sent simultaneously to multiple receivers. Each of the transmitted messages is then a distinct “cco: information bearing artifact”. However, the content of the message is the same. For this purpose, we defined the content of the message as a “cco: information content entity”, where each information bearing artifact is related to the relevant information content entities using the *bearer of* relationship. When the message is semantically transformed and sent to different receivers, this results in different information content entities. The fact that these are all transformations of one single message is captured by using a single ID, defined as a “code identifier”.

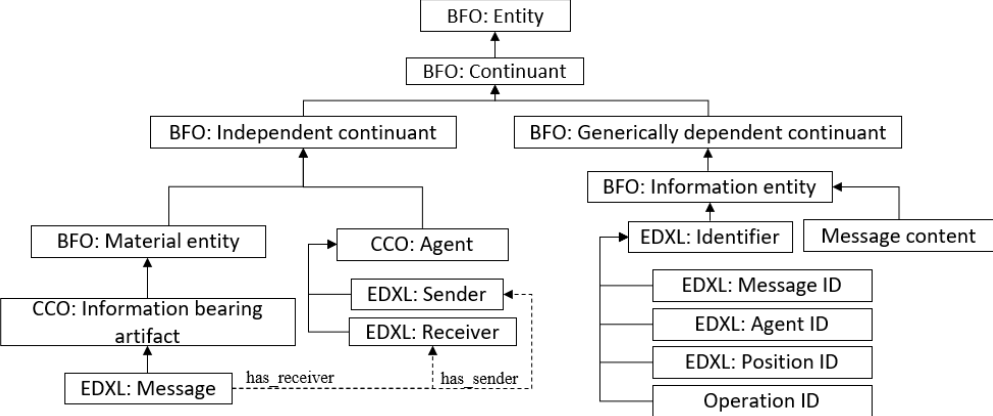


Figure 5. Definition of a message in POLARISCO

In disaster response, there is a classification of messages in order to standardize communication among stakeholders. The messages are classified into three types basing on their objectives including “*informative message*”, “*request message*” and “*response message*”. “*Informative message*” includes “*information message*”, “*alert message*”, and “*report message*”, and so forth. The aim of “*information message*” is to inform EROs of an event in a formal manner. It can be about the notification of an emergency plan launching, departure or arrival time of agents on the disaster scene and so on. Then, “*request message*” is about asking for additional resources. The commander of the on-going operation on the disaster scene requests supplementary resources or backup if the available resources are not enough to effectively manage the situation. The “*request message*” can be also about asking for information and updating the situation. A “*response message*” concerns a “*request message*”. For this purpose, we used the relationship “is about” to relate these latter as shown in Figure 6.

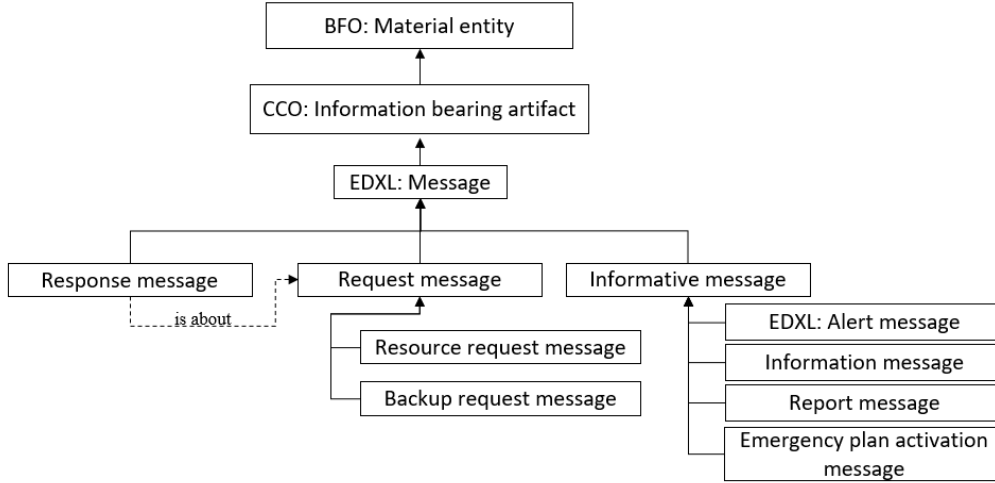


Figure 6. Types of exchanged messages

In addition, a message is characterized by some features that can be perceived as “*quality*” in BFO. In fact, “*quality*” is defined in BFO as a specifically dependent continuant that depends or inheres in an entity at all and is fully exhibited or manifested or realized in that entity. Accordingly, we defined the state (“*treated*”, “*untreated*” or “*ongoing*”), the confidentiality (“*public*”, “*private*” or “*limited*”) and the degree of criticality of the information to be exchanged (“*extreme*”, “*moderated*” or “*secondary*”) as subclasses of “*bfo: quality*”. Then, we associate a message to the defined qualities using the relationship “*has quality*”. Figure 7 shows the mentioned classes.

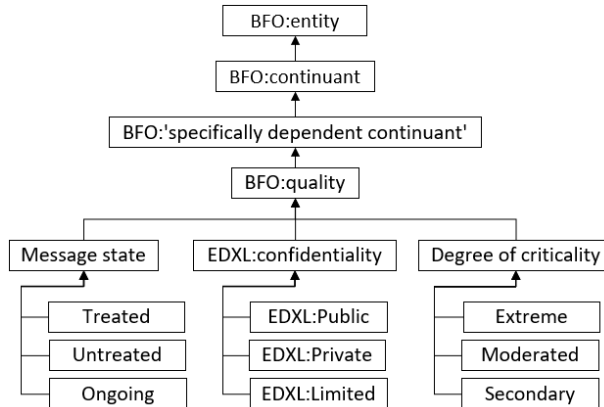


Figure 7. Qualities of a message

### 3. Message-Driving Formalism and definitions

In this subsection, we present a formal definition of an ontology and a message that will be used when elaborating the proposed algorithms.

**Definition 1:** An ontology  $O$  is formally defined as a 4-tuple:

$$O = \langle C, R, A_x, I \rangle$$

where

- (i)  $C$  represents the set whose members are the classes in the ontology.
- (ii)  $R$  is a set of relations that exist between these classes, where  $R \subset C \times C$ .
- (iii)  $A_x$  is a set of axioms.
- (iv)  $I$  is a set of instances.

**Definition 2:** The set of ontological modules is denoted by  $M$ , where  $M = \{m_1, m_2 \dots m_n\}$ , and defined as a 4-tuple:

$$m = \langle C_m, R_m, A_{xm}, I_m \rangle$$

Where

- (i)  $C_m \subseteq C$
- (ii)  $R_m \subseteq R$
- (iii)  $A_{xm} \subseteq Ax$
- (iv)  $I_m \subseteq I$

**Definition 3:** A modular ontology  $O$  is the integration<sup>4</sup> of different ontological modules  $m$  and the relations that exist among those modules where  $O = \bigcup_{i=1}^n m_i$ .

- (i) Let  $R_{intra\ m}$  be the set of intra-modules relations.
- (ii) Let  $R_{inter\ m}$  be the set of inter-module relations.
- (iii)  $R = R_{inter\ m} \cup R_{intra\ m}$

**Definition 4:** Each ontological module represents the vocabulary of an actor. We write  $A$  for the set of all actors

Where

$\forall a_i \in A$ , there is  $m_i \in M$  such that:

$$f: A \rightarrow M$$

$$a_i \rightarrow m_i$$

**Definition 5:** An act of communication between two actors is represented by a message. A message  $msg$  is defined as 5-tuple

$$\langle msgType, sender, receiver, msgIC_{source}, msgIC_{target} \rangle$$

Where

- (i)  $msgType$  represents the type of the message such as  $msgType \in MSGType$
- (ii)  $sender$  identifies the actor source of this message such as  $sender \in A$ .
- (iii)  $receiver$  identifies the actor target of this message such as  $receiver \in A$ . We consider that a  $receiver$  can be one or more.
- (iv)  $msgIC$  represents the information content of the message which is composed of a set of classes and instances from the ontology  $O$  such that:
  - $msgIC_{source}$ : information content of the sender's message where  $msgIC_{source} \subset C_{source} \cup I_{source}$
  - $msgIC_{target}$ : information content of the receiver's message where  $msgIC_{target} \subset C_{target}$

**Definition 6:** The function  $TR_s$  yields the semantic transformation of the message's information content:

$$TR_s (msg(msgIC_{source})) \rightarrow (msg(msgIC_{target}))$$

In what follows, we apply these generic definitions to POLARISCO and PROMES:

- The modular ontology  $O = POLARISCO$
- $M = \{PCC, firefighters, police, healthcare\ units, gendarmerie, public\ authorities, and\ message\}$
- $POLARISCO = m_{PCC} \cup m_{firefighters} \cup m_{police} \cup m_{healthcareUnits} \cup m_{gendarmerie} \cup m_{publicAuthorities} \cup m_{Message}$

---

<sup>4</sup> Ontology integration, or interrelations, means to put together (interrelate) multiple ontology modules to compose a new ontology.

- $C = \{“operation”, “agent”, “mean”, “mean type”, “action center”, “degree of criticality” \dots\}$
- $R_{inter\ m} = \{“realized in”, “caused by”, “has role”, “is part of”, “use” \dots\}$
- $R_{intra\ m} = \{“is equivalent to”, “belongs to”, “supervises” \dots\}$
- $A = \{Firefighters, Police, healthcare units, Gendarmerie, Public authorities\}$
- $msgType = \{“response message”, “resource request message”, “information message”, “alert message”, “report message”, “Arrival message”, “departure message” \dots\}$

#### 4. Textual transformation of the message

The input of PROMES is a list of classes and instances that belongs to the sender’s ontological modules. As a first step, this list is transformed into text using the TT algorithm (see Figure 8). In fact, this latter takes into account the predefined messages’ structures of EROs. In fact, it uses the set of selected terms to compose the message by adding text to relate them according to the predefined structure of EROs. The output of this step is a message written in formal language rather than a list of non-related technical terms and acronyms. It is a first step towards extending the message and make it more understandable by the rest of the stakeholders.

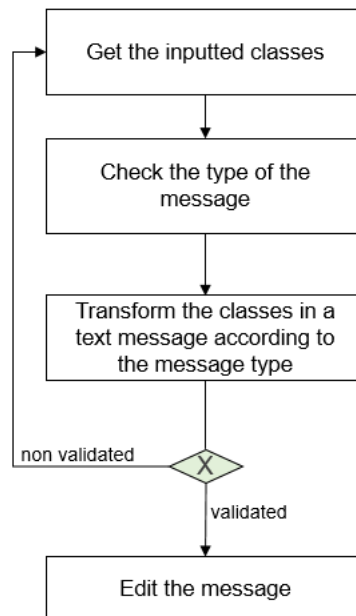


Figure 8. The different steps of the textual transformation process

#### 5. Semantic translation of the message

To perform the semantic translation, we need to carry out the semantic mapping in order to address the heterogeneity gap between stakeholders’ knowledge by identifying the related concepts. The proposed approach consists of two steps as shown in Figure 9; the mapping between the ERs ontological modules and the semantic translation of the message.

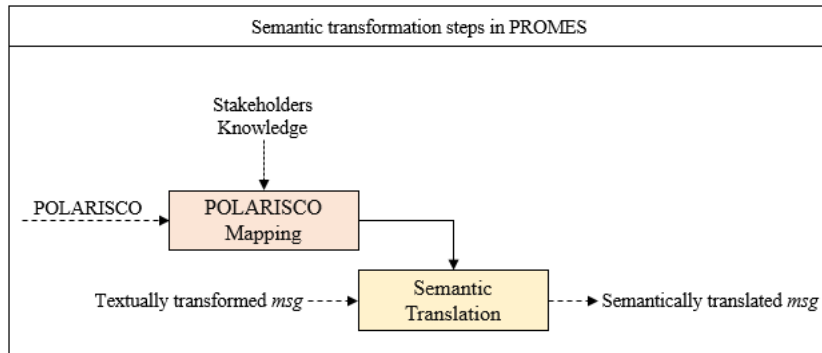


Figure 9. The ontology-driven semantic translation approach

### 5.1 The mapping between stakeholders ontological modules

The use of top-level ontologies facilitates the integration of different domain ontologies defined in their terms and thereby promotes interoperability of the associated data. Once the stakeholders modules are merged in one ontology, POLARISCO, the next step is to perform the mapping between these latter. We consider one possible kind of mapping between classes, which is “equivalentTo”. It is about representing the equivalences between the classes of the ontology module of stakeholder 1 and stakeholder 2. The semantic mapping is done in collaboration with emergency experts to ensure the effectiveness of the defined equivalence relationships. These defined relationships are the keystone of the semantic translation; they will facilitate the interpretation of the information by stakeholders. Figure 10 shows an example of equivalence relationships between firefighters module and healthcare units module. For instance, an absolute emergency “AE” for firefighters is equivalent to “P0” for healthcare units. These relationships will guide how the input message formalized with the source ontology of the sender can be transformed in message represented with the target ontology of the receiver.

When defining the mapping between the different classes, we formulated a SPARQL query in order to check the efficiency of the defined equivalent classes in each direction. This step aims to find possible conceptual errors that enables the early correction of the mappings before the implementation step.

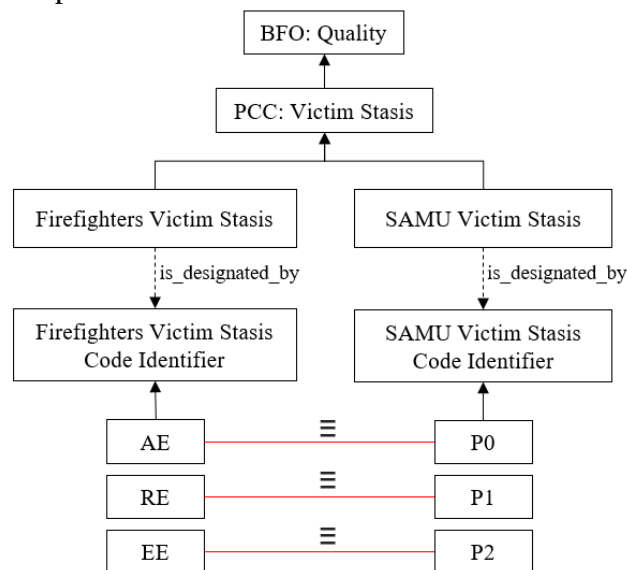


Figure 10. Example of mapping between firefighters and healthcare units modules

## 5.2 The semantic translation algorithm

If the sender and the receiver don't belong to the same ERO, PROMES proceeds to the semantic translation. To do so, we propose a semantic translation algorithm (ST) that transforms the list of terms, selected by the sender, according to the ontological module of the receiver. For each term of the information content source ( $msgIC_{source}$ ), ST checks if it is a class or an instance in POLARISCO. If it is an instance, the algorithm gets its class in order to perform the rest of the transformation. The, ST verify if the class belongs to PCC module. if it is the case, the class remains the same because the different EROs share the same term. If it is not, ST checks its equivalent class. If there is an equivalent class that belongs to the module of the receiver, the class will be substitute by its equivalent class. In case there is no equivalent class found defined in POLARISCO, ST doesn't stop at this level. Contrariwise, it adds more semantic in the message by enriching it using the class annotations (definition or/and acronym's meaning). The annotations can be used to reveal the meaning of the term even though the EROs don't use it, they can understand what is it about based on its definition. In case the class has no annotation, we search for the annotation of its superclass.

The pseudo-code of the ST algorithm is provided in Listing 2. To summarize, the ST algorithm is mainly about 5 steps as shown in Figure 11;

- (1) Find the term in the ontology.
- (2) Get its class, if it is an instance.
- (3) Check if the class doesn't belong to PCC module.
- (4) Get its equivalent class that belongs to the ontological module of the receiver and substitute them.
- (5) If there is no equivalent class, get its annotation and add it in the message.
- (6) If it has not any annotation, get the annotation of its superclass and add it in the message.

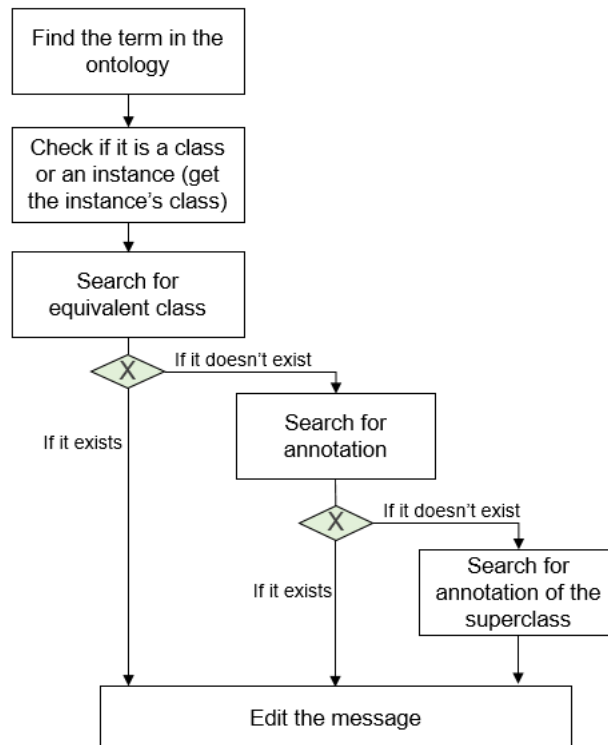


Figure 11. The different steps of the semantic translation process

---

**Algorithm 2:** Algorithm Message's Semantic Transformation

---

**Input :**  
 $msgIC_{source} = \{ic_{source1}, ic_{source2} \dots ic_{ourcen}\}$  : set of terms that compose the sender's message  
 $POLARISCO = \{m_{pcc} \dots\}$  : The global ontology  
 $C$  : set of the ontology's classes

**Output :**  
 $msgIC_{target}$ : set of terms that compose the receiver's message

**Variables :**  
 $ic'_{source}$ : class  
 $ec$ : equivalent class  
 $sp$  : superclass  
 $A$  : set of annotation  
 $indiv$  : instance of a class  $c$

**begin**  
  Initialize a list  $msgIC_{target} \leftarrow \{\}$  ;  
   $indiv = null$  ;  
   $ic'_{source} = null$  ;  
  **while**  $msgIC_{source} \neq \emptyset$  **do**  
    **foreach**  $ic_{source} \in msgIC_{source}$  **do**  
       $find(ic_{source})$  in  $POLARISCO$ ;  
      **if**  $ic_{source} \in C$  **then**  
        |  $ic'_{source} \leftarrow ic_{source}$ ;  
      **else**  
        |  $ic'_{source} \leftarrow getClass(ic_{source})$ ;  
        |  $indiv \leftarrow ic_{source}$ ;  
      **end**  
      **if**  $ic'_{source} \notin m_{pcc}$  **then**  
        |  $ec \leftarrow getEquivalentClass(ic'_{source}, m_{receiver})$ ;  
        | **if**  $ec \neq null$  **then**  
          |  $msgIC_{target} \leftarrow indiv + ec$ ;  
        | **else**  
          |  $A \leftarrow getAnnotation(ic'_{source})$ ;  
          | **while**  $A = \emptyset$  **do**  
            |  $sp \leftarrow getSuperClass(ic'_{source})$ ;  
            |  $A \leftarrow getAnnotation(sp)$ ;  
          | **end**  
          |  $msgIC_{target} \leftarrow ic'_{source} + indiv + A$ ;  
        | **end**  
      | **else**  
        |  $msgIC_{target} \leftarrow ic_{source}$ ;  
      | **end**  
    | **end**  
  | **end**  
  | **end**  
  |  $return (msgIC_{target})$ ;  
**end**

---

#### IV. Implementation and use-case evaluation

To build the proposed ontology-driven messaging service prototype, PROMES, Java and Eclipse IDE are used. Maven is used for managing the project. OWL API<sup>5</sup> has been applied to manipulate the ontology. It is a Java API used for the creation and manipulation of OWL ontologies when developing ontology-based applications. In addition, Java Swing is used to implement the GUI. A use-case is presented in the following to show a scenario test in order to evaluate the functioning of PROMES.

---

<sup>5</sup> <http://owlapi.sourceforge.net/>

Let's consider an example of an act of communication between firefighters and healthcare units. Once the firefighters' unit is on the field, they figure out that they don't have enough vehicles of succor and assistance to victims (VSAV). Accordingly, they need backup from healthcare units to handle the rescue of a large number of victims in a critical situation. The firefighter commander (FC) uses PROMES which is integrated into the POLARISC platform as a communication tool.

FC uses the PROMES guided user interface (GUI) and starts by choosing the receiver and the type of the message to send, which is a resource-request message in this case. Then, FC chooses the appropriate terms step by step as required by the GUI as shown in figure 12. Once the terms are selected, the textual transformation of these latter is done. It is up to the firefighter commander to validate the result of this first step. After the validation, the semantic translation is started.

For the current use-case, PROMES starts the checking process class by class (see Figure 13 and 14). First, the term "TA75" is an instance so it looks for its class ("Terrorist Attack"). "Terrorist Attack" class belongs to PCC module. Hence, PROMES keeps the same term. Second, "VSAV" is a subclass of "firefighters' vehicle" and there is no equivalent class in the healthcare units module, so PROMES adds the annotation of VSAV to explain the meaning of the acronym and its definition. Third, "victims'rescue" is a subclass of "act" and belongs to PCC module so it remains unchangeable. Then, "AE" (Absolute Emergency) is a subclass of "firefighters' victim stasis code identifier" which is equivalent to the subclass of "SAMU victim stasis code identifier" "P0". Therefore, "AE" is substituted with "P0". The action center of the ongoing operation is called "AC\_PS\_75" by firefighter while it is called "P\_75" by healthcare units. Thus, "AC\_PS\_75" is replaced by "P\_75". We can notice that the semantically transformed message is extended and improved so it can be less ambiguous when received by the healthcare units (see Figure 15).

The screenshot shows the POLARISC GUI with the following configuration:

- Receiver:**  Firefighters,  Healthcare units,  Police forces,  Gendarmerie,  Public authorities. The dropdown menu is open, showing: Medical rescue director, Medical rescue commander, Regulator doctor, Ambulance driver.
- Message type:** Information, Alert, Resource request (selected), Response.
- Operation:** TA75 (selected), Fire65, ACD34, RSC98.
- Resource:** Ground vehicle (selected), Aircraft, CCF, FPT, VSAV (selected), VPC.
- Nombres:** 1
- Act:** Victims' rescue (selected), Victims' evacuation, Victims' transfer.
- Degree of criticality:**  AE,  EE,  RE,  PE.
- Action center:** AC PS 75 (selected), AC F 65, AC A 34, AC R 98.

Buttons: Send, Reset.

Figure 12. PROMES' guided user interface



Information content source	{TA75, VSAV, victims' rescue, AE, AC PS 75.}
Output of TT	<b>TA75: we needs VSAV for victims' rescue in AE in AC PS 75.</b>
Output of ST	{TA75, VSAV, Vehicle of Succor and Assistance to Victims, VSAV is a firefighters vehicle reserved for victims' first aid , P0 , P 75.}
Information content target	<b>TA75: we needs VSAV*(<u>Vehicle of Succor and Assistance to Victims</u> ) for victims' rescue in <u>P0</u> in <u>P 75</u>. <u>*VSAV is a firefighters vehicle reserved for victims' first aid.</u></b>

Figure 13. Message transformation example

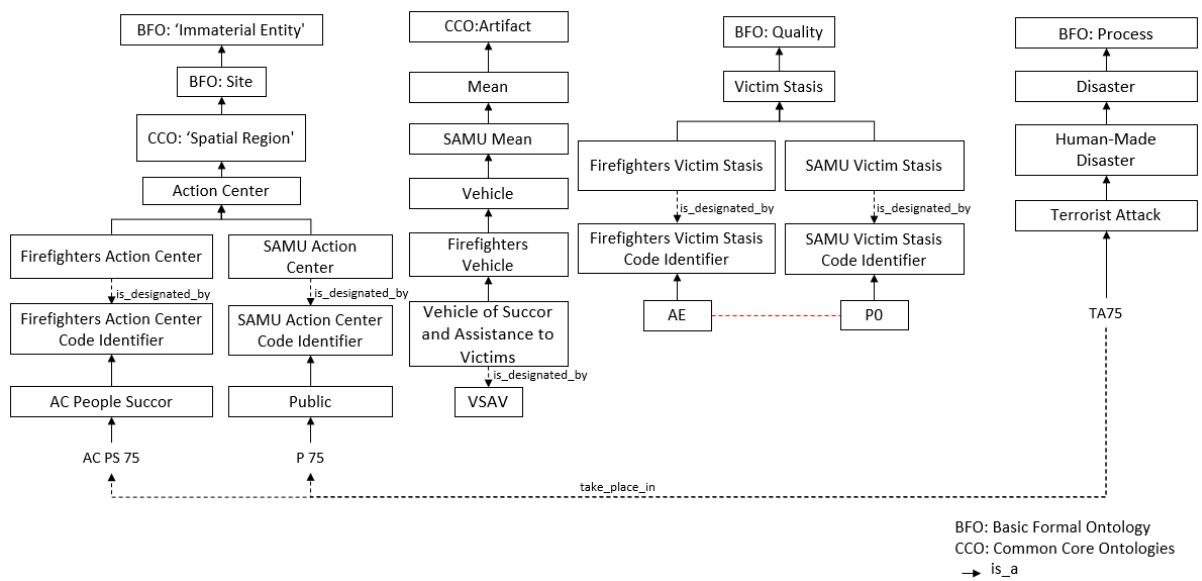


Figure 14. A sample part of an instance of a terrorist attack

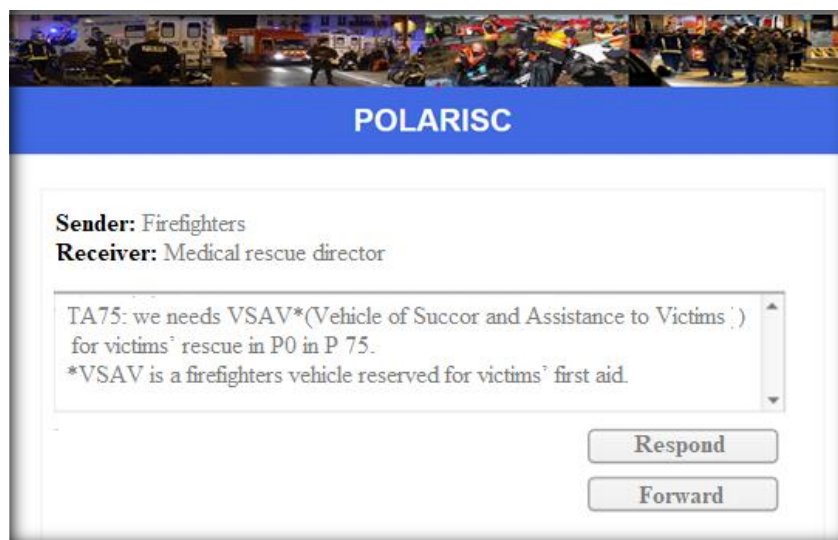


Figure 15. The received message

## V. PROMES validation

In order to validate the proposed approach, the performance of PROMES is analyzed in terms of efficiency and accuracy following the validation approach presented in (Kim et al., 2012) and (Moreira, 2019). It consists of describing the accuracy and the efficiency of a set of messages produced by the proposed approach. According to (Kim et al., 2012), translation accuracy is to “correctly translate and describe semantic information of an input message in the form of a target message”, and the translation efficiency is to “minimize communication delays by efficient translations of many semantic messages”. Hence, the translation efficiency is evaluated in terms of processing complexity. Accordingly, we will check if the proposed system responds to the fixed requirements.

To do so, as a first step, we defined a set of 100 test messages in the form of a message source and its planned translation output. Then, we tested each message and we compared the output message versus the expected result. Each information content of the input message `msgICsource`, was transformed using the TRs function (`TRs(msg(msgICsource)) → (msg(msgICTarget))`). In fact, we tested the proposed approach with the use of the GUI (1) and without it (2). In order to measure the semantic translation accuracy of the test messages, we calculated the number of output messages that match with the expected ones regarding the total number of test messages using the following formula:

$$\frac{\text{No. of matched to expected translation output correct semantic translation output}}{\text{No. of test messages}}$$

As shown in Figure 16, the results showed that using the GUI (1), the semantic translation is 100% accurate. In fact, there are no cases of untranslatable and syntax errors because the GUI input relies on well-defined classes loaded from POLARISCO. However, without the GUI (2), only forty-seven from one hundred tested messages were accurate. These results are due to the manual input of the message. Stakeholders may not make intention the correct spelling of each word and especially the acronyms ones; a lower case letter is written instead of an upper letter or the opposite handicap the semantic translation process. Thus, the use of the GUI guarantees the accuracy of the output message because it loads classes from the ontology in order to avoid syntax errors that may impede the semantic translation process. Using the GUI, all messages were syntactically correct. In addition, the bi-directional verification of the defined equivalent classes in the ontology mapping step guaranteed the semantic translation accuracy.

As expected, the time of the input using the GUI is lower than the manual input as shown in Figure 17. However, as the messages get larger, the input time is longer. In fact, like any new software application, the final users, ERs in this context, need to be formed to use the application. We did a test with stakeholders to get their feedback about the use of the proposed GUI. For this, we asked one stakeholder to input the same message ten times using the GUI and then to manually write the message. We can see that the input time is decreasing each time as depicted in Figure 18. Consequently, ERS preferred the use of the GUI because it will save time during operational response and guarantee an accurate semantic translation of the exchanged information.

Concerning the semantic translation efficiency, it was measured according to the total processing time of PROMES. To do so, (see Figure 19). We found that the processing time depends mainly on the number of classes that compose the message that doesn't belong to the PCC module and should be translated. One can notice that the processing time can be a bit longer than usual due to the fact that the message is semantically richer.

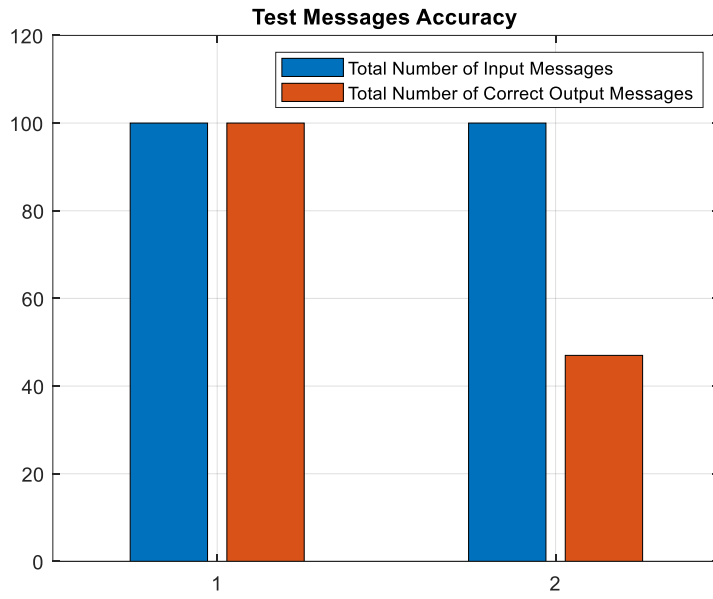


Figure 16. Results of the semantic translation accuracy

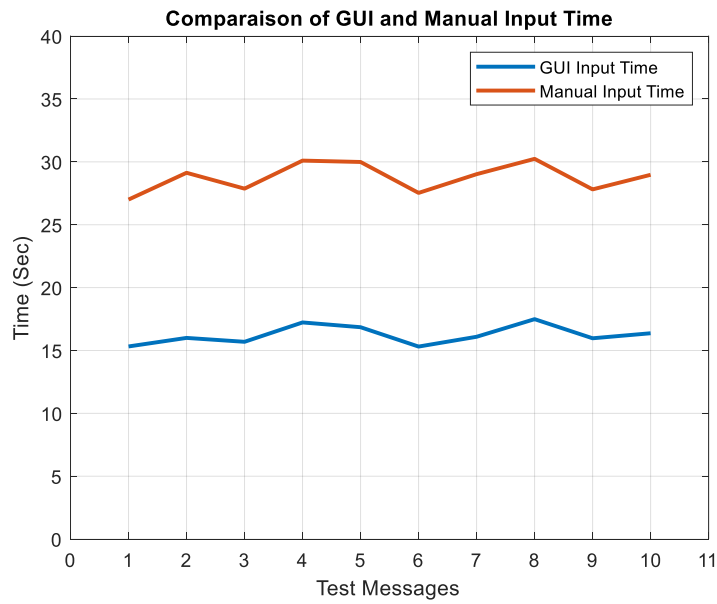


Figure 17. Results of GUI and manual input time

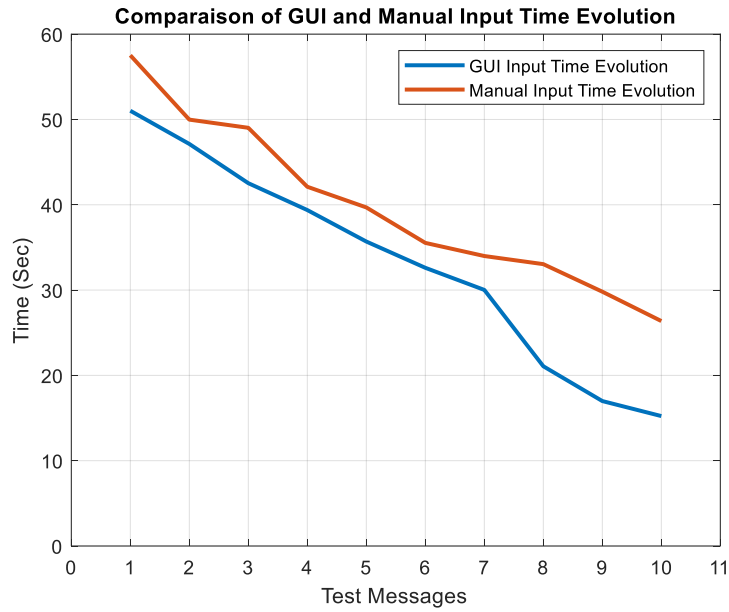


Figure 18. Results of the input time of the same message by the same stakeholder

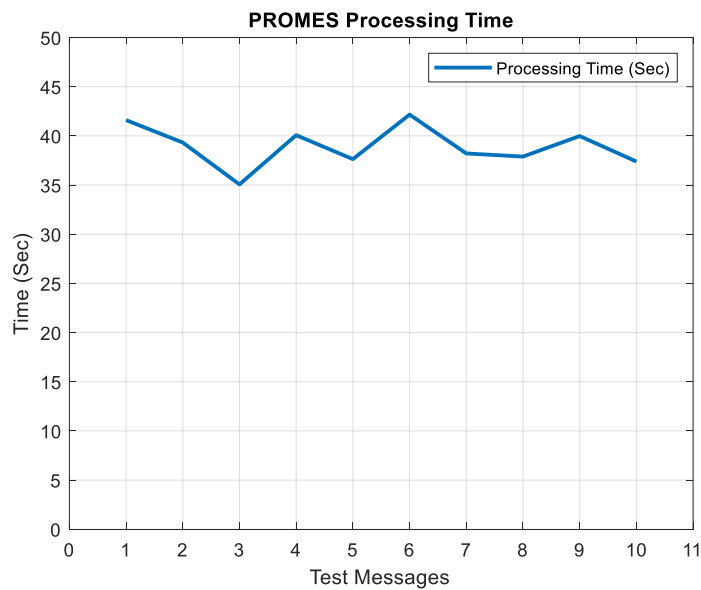


Figure 19. Results of the semantic translation efficiency

The numerical results demonstrate the efficiency and robustness of PROMES so that it can be used by ERs. It ensures effective and accurate semantic translation of the knowledge expressed using the source ontology of a stakeholder  $a$  into knowledge expressed using the target ontology of a stakeholder  $b$ . Accordingly, it ensures semantically interoperable information exchange by providing a mapping among stakeholders' vocabularies. The more the semantic translation of the message is based on the equivalent classes between vocabularies  $a$  and  $b$ ; the more the semantic interoperability is higher. Hence, we can conclude that PROMES responds to the fixed requirements and can greatly improve the efficiency of communication among stakeholders during disaster response.

## VI. Conclusion

ERs often under-perform due to a lack of proper communication and information sharing among them. In order to tackle this problem, we propose PROMES, within the scope of POLARISC project, that can greatly improve communication among stakeholders during disaster response. It enables information tractability and consolidation and ensures an efficient and accurate semantic translation of ERs exchanged information and subsequently empowers semantic interoperability among the involved stakeholders in the process of operational disaster response. As future work, further research to map the entire ontologies will be done in collaboration with emergency experts. Currently, we consider one possible kind of mapping between classes, which is “equivalentTo”. As future works, we plan to take into account properties equivalence. Accordingly, we aim to propose a semi-automatic mapping between properties. It is interesting if the different properties will be automatically analyzed in order to find correspondences between classes. Then, the mapping will be confirmed manually to ensure the correctness of the mapping results. For instance, when a new class is added in one stakeholder module, its properties will be analyzed to find out the possible equivalences with the rest of the ontology’s classes.

## Reference

- Arp, R., Smith, B., & Spear, A. D. (2015). *Building ontologies with basic formal ontology*. MIT Press.
- Boutinaud, B. G. F (2017). *Cooperation & Interaction: The Paris Terrorist Attacks*. In Norwegian conference for civil protection and crisis management.
- Bharosa, N., Lee, J., & Janssen, M. (2010). Challenges and obstacles in sharing and coordinating information during multi-agency disaster response: Propositions from field exercises. *Information Systems Frontiers*, 12(1), 49-65.
- Bicer, V., Laleci, G. B., Dogac, A., & Kabak, Y. (2005). Artemis message exchange framework: semantic interoperability of exchanged messages in the healthcare domain. *ACM Sigmod Record*, 34(3), 71-76.
- Bhattacharjee, S., Kanta, S., Modi, S., Paul, M., & DasBit, S. (2016). Disaster messenger: An android based infrastructure less application for post disaster information exchange. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* (pp. 1-5). IEEE.
- Bittner, T., Donnelly, M., & Winter, S. (2005). Ontology and semantic interoperability. *Large-scale 3D data integration: Challenges and Opportunities*, 139-160.
- Barros, R., Kislansky, P., Salvador, L., Almeida, R., Breyer, M., & Pedraza, L. G. (2015). EDXL-RESCUER ontology: Conceptual Model for Semantic Integration. In *ISCRAM*.
- Clarke, P. (2003). *Crisis communications: lessons from September 11*. Rowman & Littlefield Publishers.
- Chipara, O., Plymoth, A. N., Liu, F., Huang, R., Evans, B., Johansson, P., ... & Griswold, W. G. (2011). Achieving reliable communication in dynamic emergency responses. In *AMIA Annual Symposium Proceedings (Vol. 2011, p. 238)*. American Medical Informatics Association.
- Chan, S., Haugh, B. A., Loaiza-Lemos, F. L., Wartik, S. P. (2017). *Public Safety and Emergency Management Communications Ontology*. IDA Document D-8583
- Elmhadhbi, L., Karray, M. H., & Archimède, B. (2018). *Towards an Interoperable Operational Emergency Response System for Large-Scale Situations: POLARISC*. ISCRAM 2018

- Elmhadhbi, L., Karray, M. H., & Archimède, B. (2018). Toward the use of top-level ontologies for semantically interoperable systems: An emergency management use case. In *Enterprise Interoperability VIII*.
- Elmhadhbi, L., Karray, M. H., & Archimède, B. (2019). A Modular Ontology for Semantically Enhanced Interoperability in Operational Disaster Response. *ISCRAM 2019*
- Gusty, D., & Dwarkanath, S. (2010). Improving the effectiveness of emergency response through improved standards. In *Proceedings of the 11th Annual International Digital Government Research Conference on Public Administration Online: Challenges and Opportunities* (pp. 231-232). Digital Government Society of North America.
- Heiler, S. (1995). Semantic interoperability. *ACM Computing Surveys (CSUR)*, 27(2), 271-273.
- Kean, T. H., & Hamilton, L. H. (2004). *The 9/11 Report*. New York: St. Martin's Press.
- Kim, J., Kang, S., Lee, J., & Choi, B. W. (2012). A semantic translation method for data communication protocols. *Journal of Systems and Software*, 85(12), 2876-2898.
- Kristiansen, E., Johansen, F. H., & Carlström, E. (2019). When it matters most: Collaboration between first responders in incidents and exercises. *Journal of Contingencies and Crisis Management*, 27(1), 72-78.
- Manoj, B. S., & Baker, A. H. (2007). Communication challenges in emergency response. *Communications of the ACM*, 50(3), 51-53.
- Meissner, A., Wang, Z., Putz, W., & Grimmer, J. (2006). Mikobos-a mobile information and communication system for emergency response. In *Proceedings of the 3rd International ISCRAM Conference* (pp. 92-101).
- Moreira, J. L. (2019). SEMIoTICS: Semantic Model-driven Development for IoT Interoperability of Emergency Services: Improving the Semantic Interoperability of IoT Early Warning Systems.
- Moreira, J., Daniele, L., Pires, L. F., van Sinderen, M., Wasielewska, K., Szmeja, P., ... & Paprzycki, M. (2017, November). Towards IoT Platforms' Integration Semantic Translations between W3C SSN and ETSI SAREF. In *SEMANTICS Workshops*.
- Nossiter, A. (2015). Response to Paris Attacks Points to Weaknesses in French Police Structure. In *the New York Times* December. 31, 2015.
- Nguyen, D. N., Kopena, J. B., Loo, B. T., & Regli, W. C. (2010). Ontologies for distributed command and control messaging.
- Philippe, J. M., Brahic, O., Carli, P., Tourtier, J. P., Riou, B., & Vallet, B. (2016). French Ministry of Health's response to Paris attacks of 13 November 2015. *Critical Care*, 20(1), 85.
- Real, F. J. Q., McNeill, F., Bella, G., & Bundy, A. (2017). Improving Dynamic Information Exchange in Emergency Response Scenarios. In *ISCRAM*.
- Rudnicki, R. (2016) An Overview of the Common Core Ontologies, *CUBRC*.
- Shvaiko, P., & Euzenat, J. (2011). Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1), 158-176.
- Smith, B., Vizenor, L., & Schoening, J. (2009). Universal Core Semantic Layer. In *Ontology for the Intelligence Community: Proceedings of the Third OIC Conference*. CEUR, vol. 555. pp. 1-5.
- Wang, T., Truptil, S., & Benaben, F. (2015). Applying a semantic & syntactic comparisons based automatic model transformation methodology to serve information sharing. In *Proceedings of the International Conference on Information and Knowledge*

Engineering (IKE) (p. 3). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).