Journal of Universal Computer Science, vol. 26, no. 5 (2020), 549-582
submitted: 11/3/19, accepted: 6/4/20, appeared: 28/5/20 CC BY-ND 4.0

An Ontological Approach to Support Dysfunctional Analysis for Railway Systems Design

Sana Debbech
(COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, Univ Lille
F-59650 Villeneuve d'Ascq, France
sana.debbech@gmail.com)

Simon Collart-Dutilleul
(COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, Univ Lille
F-59650 Villeneuve d'Ascq, France
Institut de Recherche Technologique Railenium, F-59300 Famars, France
simon.collart-dutilleul@ifsttar.fr)

Philippe Bon
(COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, Univ Lille
F-59650 Villeneuve d'Ascq, France
philippe.bon@ifsttar.fr)

Abstract: Dysfunctional analysis is an essential and demanding task in the early development stages of safety-critical systems (SCSs). Nevertheless, current practices present several drawbacks. Generally, a common dysfunctional analysis conceptualization is missing and it is dependent on safety analysis techniques. Moreover, some safety analysis methods require well-known system behaviors expressed by dynamic models such as sequence diagrams and finite automata. However, the dynamic character of these models increases their susceptibility to changes and then they are not obtainable in the early design stages. Since dysfunctional analysis highly relies on the experience of safety analysts and the feedback (REX) obtained from previous systems development, there is a need to formalize this knowledge domain in a structured way to ensure its future reuse. Furthermore, safety measures derived from this dysfunctional analysis approach must be strongly linked to a goal-oriented perspective and adapted to a specific context. For this purpose, this paper presents a real-world semantics interpretation and conceptualization of dysfunctional analysis related concepts based on the Unified Foundational Ontology (UFO) and well-known standards to avoid ambiguities. The proposed Dysfunctional Analysis Ontology (DAO) aims to provide a systematization of the goal-oriented dysfunctional analysis through a terminological clarification in order to prevent hazards in the first design phases. Then, a DAO formalization is proposed using the Web Ontology Language (OWL). Finally, the DAO pattern is applied to two different real critical scenarios from the railway domain in order to illustrate and evaluate this ontological approach.

Key Words: Dysfunctional Analysis Ontology, Safety reasoning, Goal, Context, UFO, OWL, Safety critical railway systems
Category: M.0, M.1, M.4, M.8

1 Introduction

During the design phases of Safety-critical systems (SCSs), safety analysis should be integrated as early as possible [Debbech et al. 2018a], as required by safety standards in several domains, e.g., EN50129 [EN-50129 2003] for railway systems, [ISO/DIS 26262-1 2009] for the automotive domain and [IEC61508 2010] for generic control systems. According to the standard EN50129 [EN-50129 2003], a hazard is a condition that can lead to an accident. Based on this definition, safety analysis can be performed through several methods following a set of steps:

1. identifying risks using preliminary hazard analysis (PHA),
2. defining how involved components contribute to hazardous situations,
3. deriving safety requirements to mitigate hazards [Heimdahl 2007].

For the purpose of safety assessment, several efforts have been devoted to exploit the hazard knowledge identified among safety analysis to elicit safety requirements. However, this task is not always easy especially when the system development is still in progress. Besides, the traditional safety analysis techniques are always based on a good knowledge of system behaviors, which is not easy to acquire in the first design stages. Consequently, there is lack of a conceptual clarification and a complete taxonomy aiming to allow a well-established formalization of a failure and its related concepts in the SCSs terminology.

Dysfunctional analysis elements such as a failure, its causes and effects are usually formulated in an informal way by what they present and how they are presented. At this level, we consider two dysfunctional analysis aspects: a system component is exposed to a failure and a system component causes a failure, which triggers accidents. The causality relationship through cascading failures has to be considered in the dysfunctional analysis process.

The hazard definition by the standard EN50129 [EN-50129 2003] represents some ambiguities in terms used such as “condition”, “accident” and the causality relationship between them (“lead to”). It suffers from a lack of a precise definition of these terms in real-world semantics (object, relation, property, event, etc). To the best of our knowledge, most existing works are only focusing on the hazard knowledge capture in order to directly derive related safety requirements that mitigate the hazard. But some of these works conceptualize specific safety analysis methods without real world semantics. Consequently, there is a need of a common conceptualization of all dysfunctional analysis aspects in order to allow an interoperable view of safety analysis methods such as the Preliminary Hazard Analysis (PHA), the Failure Modes and Effects Analysis (FMEA) and the Failure Tree Analysis (FTA). Furthermore, relations between safety measures and Goal-Oriented Requirements Engineering (GORE) concepts such as goal, task and the

related context are not considered. Therefore, the need to match the safety and the GORE perspectives arises in order to obtain a shared view between actors involved in the SCSs design.

In this paper, a Dysfunctional Analysis Ontology (DAO), a domain ontology grounded in UFO, is proposed in order to deal with semantic heterogeneity and disagreement problems. Then, the proposed interpretation is based on the extraction of relevant definitions from standards aiming to provide a common vocabulary between design engineers and safety analysts. Furthermore, the goal-oriented perspective is considered in order to establish a consistent safety reasoning which can be adapted to a context. From this perspective, we formulate the following research questions:

- **RQ1:** Can we provide a structured interpretation of the dysfunctional analysis with real-world semantics to fill gaps mentioned above?
- **RQ2:** How can we semantically interpret derived safety measures and link them to goal and context concepts in order to make better safety-related decisions in the SCSs development?

In order to answer these RQs, this paper is organized as follows: Section 2 discusses the knowledge engineering and the ontologies hierarchy, the reused concepts of UFO and related works in SCSs domains. Section 3 presents the dysfunctional analysis conceptualization based on Unified Foundational Ontology (UFO). Then, the interpretation of relations between safety-related concepts and some GORE concepts is defined. Besides, the Web Ontology Language (OWL) formalization of the proposed DAO is provided in order to allow a better expressiveness, re-usability and reasoning capabilities. Section 5 represents a case study from railway systems and a real accident scenario in order to illustrate and evaluate the proposed domain ontology. Finally, the conclusion and future works are outlined in Section 6.

2 Knowledge engineering and Ontologies

An ontology is a structured representation of a domain knowledge. The original definition of ontology comes from [Gruber 1993] as *"an explicit specification of a conceptualization"*. Then, Borst defined the ontology as *"a formal specification of a shared conceptualization"* [Borst 1997]. The combination of these definitions shows that the conceptualization should express a shared view between different parts and the explicit specification should be expressed in a formal way. Consequently, we propose to define the ontology as a conceptual model of a structured representation of a domain knowledge consisting of a set of concepts, relations, axioms, and semantics in order to interpret them, as mentioned below:

Definition 2.1 (Ontology). Let O be the ontology considered as a 5-tuple: $O = \{D, C, R, A, S\}$ where:

-D is the domain of discourse;

-C is the set of concepts or classes within this domain;

-R is the set of binary relations between these concepts which can be taxonomic or associative relationships;

-A is the set of axioms to constrain values of classes or instances and relations;

-S represents semantics used to interpret concepts and relations between them.

An ontology has two important aspects to be considered: The completeness in terms of real-world semantics employed for interpretation of the domain concepts, and the re-usability to allow the extensibility without modifying well-founded/upper concepts. Several ontologies exist in the literature aiming to conceptualize the hazard, such as [Sigwarth et al. 2015] and [Cheatham et al. 2017]. However, they don't consider dysfunctional analysis concepts in terms of component failures, human errors or unsafe behaviors of any environment object, their causes and their effects. Besides, they don't consider a real-world interpretation of concepts and relations between them.

Real-world semantics aim to establish relations between dysfunctional analysis concepts and foundational concepts such as object, event, situation, disposition in the development of a domain ontology [Guizzardi 2005]. Consequently, these foundational concepts provide the externalization of real-world semantics of ontology concepts, the choice of a pattern to represent a domain knowledge and its sound and consensual top level justification. That is a good reason to choose a foundational ontology, which is a model of the common concepts and relations, to answer the **RQ1**. There are several foundational ontologies in the literature, such as GFO [Herre et al. 2006], BFO [Arp et al. 2015], DOLCE [Masolo et al. 2003] and UFO [Guizzardi 2005]. In this study, we are particularly interested in the Unified Foundational Ontology (UFO) which provides a complete set of foundational concepts, comparing it to others, in order to cover the dysfunctional analysis aspects such as **Moment**, **Substantial**, **Situation** and **Event**. The discussion around this choice is argued and illustrated in a previous work [Debbech et al. 2018b]. The UFO ontology concepts and their relations reused in this study are described in the next section.

2.1 The Unified Foundational Ontology-UFO

In recent years, several efforts have been devoted to use foundational ontologies (also known as upper level or top-level ontologies) to support a real-world semantics representation of SCSs in several domains and to provide a reference model of a given domain. Furthermore, top-level ontologies allow a better conceptualization of a domain in terms of clarity, expressiveness and truthfulness

regardless of the requirements. From this purpose, we reuse UFO concepts in order to instantiate other concepts which are able to represent dysfunctional analysis in a structured way.

As a foundation ontology, UFO provides a wide set of concepts and causal relations that are able to cover important aspects of dysfunctional analysis. A full description of UFO may be found in [Guizzardi 2005]. Relevant foundational concepts and relations for this study are illustrated in Figure 1 using OntoUML, a Unified Modeling Language (UML) extension for the ontology-driven conceptual modeling based on ontological distinctions put forth by UFO-A [Guizzardi 2005]. In this diagram, concepts are represented as rectangles, associative relations are labeled by "▶" for the reading direction, cardinality is mentioned on each end of associative relations and the subsumption relationship is represented by "△" connecting a sub-concept to its super-concept. In the remainder of this section, both concepts and their instances are used interchangeably to discuss some railway illustrative examples. The definitions to be known for the understandability of railway examples are detailed below. The track circuit is a technical device which detects the occupancy of the area. The Movement authority (MA) is a distance which ends by an End Of Authority (EOA). The EOA denotes a signal to stop the train. More details about the railway concepts may be found in [Schön et al. 2014]. In this paper, concepts and relations between them are written respectively in bold and in italic styles.

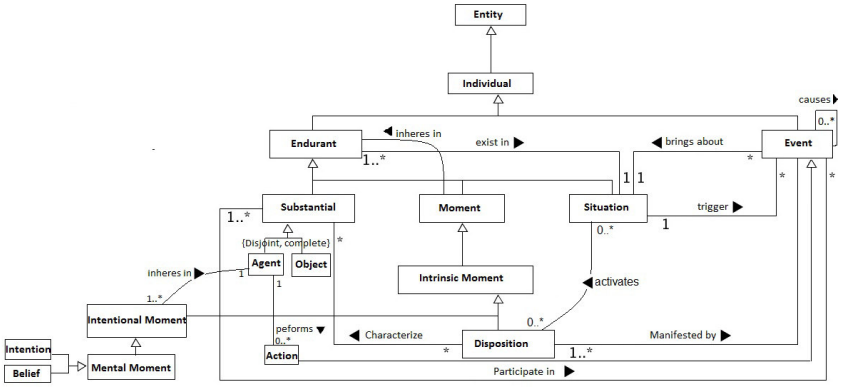

Figure 1: *Fragment of UFO showing Events and Endurants*

As presented in Figure 1, UFO distinguishes two types in the taxonomy of individuals between **Event** and **Endurant**. An event, i.e.: an instance of **Event**, is an entity which extends in time while obtaining its constituent temporal parts. In other words, event parts cannot exist simultaneously and an event depends

existentially on its parts. For instance, the constituent parts of the event **collision between two trains** are "collision in an occupied area" and "trains bound off", which exist in a chronological order. Contrarily, an endurant, i.e.: an instance of **Endurant**, is an entity with a unique identity and keeping it over time.

An **Endurant** has many sub-concepts such as **Substantial**, **Situation**, **Moment** and **Disposition**, that are useful for this work. A substantial, i.e.: an instance of **Substantial**, is an endurant existentially independent in time of others edurants. For instance, *a train* is a substantial whose existence is independent of others. A situation, i.e.: an instance of **Situation** is established by one or many endurants. In this context, a situation is considered as a state of affairs or a combination of circumstances at a given time. For example, "a train is moving" is a situation considered as a continuous behavior that *triggers* the event "the area is occupied". A moment, i.e.: an instance of **Moment** depends on the existence of several other endurants such as the *occupancy of the area* is dependent of the presence of the *train* and the *track circuit*. It justifies the relation *inheres in* between a **Moment** and an **Endurant**. In contrast, a disposition, i.e.: an instance of **Disposition** is a special type of **Moment** and it depends existentially on one single endurant. For instance, *the train speed* depends only on the train.

A **Disposition** is manifested in certain **Situations** by the occurrence of an **Event**. The relation between the **Disposition** and its dependent **Endurant** is named *characterize*. The relation between the **Situation** and its composed **Endurants** is named *exist in*. Foundational Causal relations defined in the UFO between **Situation** and **Event** are named *trigger* and *brings about*. Furthermore, an **Event** occurs by the *manifestation of* different **Dispositions** *existing in* a **Situation** (*trigger* relation). For example, "the train enters in an area crossing a closed signal" event is the manifestation of "the train movement" disposition from the train and "the permission to cross an End of Authority (EOA)" disposition from the traffic agent. Then, an **Event** can change a state of affairs from a **Situation** to another one by the *brings about* relation. For instance, "the train cross an EOA" event changes the reality from "the train is moving at a specific speed" situation to "the train is moving at the target speed at the EOA". There is a technical link between the target speed and the need of stopping in the EOA.

Comparing UFO with other foundational ontologies, one of the differences consists in defining two concepts to distinguish the type of **Substantial**. In the present study, only the **Agent** and **Object** concepts are considered. An **Object** is defined as a non-agentive substantial particular. An **Agent** as a **Substantial** is a concrete particular that bears intentional properties (**Mental Moments**) such as **Belief**, **Intention** and **Desire**. **Intentions** represent the internal commitment of the **Agent** to act towards the goal by a plan to accom-

plish it [Negri et al. 2017]. A **Belief** is based on **Stakeholder's Assumptions** and denotes a **Situation** that a **Stakeholder** believes to be true.

In order to answer to the **RQ2**, we reuse the fragment of mental moments proposed by UFO and illustrated by Figure 2 in order to cover the goal-oriented perspective.

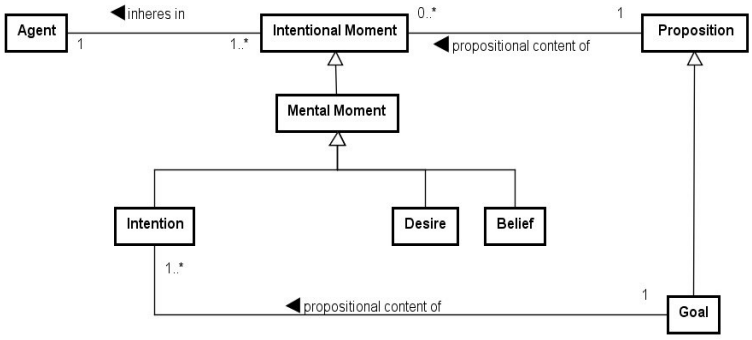

Figure 2: *Conceptual model of mental moments proposed by UFO [Guizzardi 2005]*

A **Goal** as a *propositional content of* an **Intention** is always associated with a plan. Moreover, an **Action** is an **Event** *performed* by an **Agent** in order to satisfy the **Intentions** or the **Belief** of that **Agent**. The pattern justification of these concepts interpretation and the illustration by railway examples may be found in [Debbech et al. 2018b]. The systematization of the goal-oriented safety reasoning provides a shared conceptual harmonization of the overall process between the design and safety stakeholders. In a later stage of the design phases, it ensures the goal-oriented safety management process and the induced requirements management process due to the dynamic changes on the system behavior.

2.2 Related work & Discussion

In the last decades, ontologies have been widely used in several domains for the safety analysis and the design engineering process. Most of existing works considered the conceptualization of a specific safety analysis method in order to provide a structured representation and management of the knowledge domain. In [Rehman and Kifor. 2016], an ontology is proposed to support the Failure Modes and Effects Analysis (FMEA) knowledge management for the automotive domain. Then, it provides mitigated actions aiming to deal with the expected risk and allows the information retrieval using its operational version. However,

it is still dependent on the system behavior analysis and focus on the component failures analysis only, rather than the human errors and the environment disturbances. Furthermore, the related taxonomy is not proposed in real-world semantics using an upper level ontology-driven interpretation and the ontology development is not performed using a systematic approach in order to clearly identify the purpose of the ontology and to evaluate it regarding some criteria. In [Zhou et al. 2017], a conceptualization of the hazard knowledge is proposed in order to identify the hazard from the design stages and elicit safety requirements that mitigate this hazard. Although the proposed hazard ontology is grounded in UFO, there is a lack of an OWL formalization in order to allow reasoning and to enhance its expressiveness and its reuse. Then, the terminological clarification of hazard causes such as failure types and sources is not considered. Moreover, the safety requirement elicitation is not performed using a requirement engineering approach and there is a lack of a semantic link between the requirement engineering and safety aspects. In this paper, we try to fill the gaps mentioned above and we propose a reference domain ontology of dysfunctional analysis with an interoperable view between existing safety analysis methods. In other words, the present study aims to complement the existing literature.

In the railway domain, ontologies are generally used for the big data risk analysis in order to represent the railway data integration and the traceability of safety information [Tutcher 2014]. They provide decisions management support for GB railway systems [Lewis 2012] and [Van Gulijk et al. 2015]. Then, ontologies development over railways have been focused on the documents formalization of System Requirement Specification of the ERTMS/ETCS system [Hoinaru et al. 2013] and the Case-Based Reasoning (CBR) for railway accidents support [Maalel et al. 2012]. However, there is a lack of a reference ontology domain of dysfunctional analysis which considers all its aspects, the standards definitions of its terminology and the railway systems needs. In the proposed approach, the common vocabulary and the systematic approach to develop our proposed ontology makes it original and reusable for other safety critical domains. Furthermore, the knowledge matching between safety and requirement engineering provides a shared view between actors and avoids ambiguities between them. The OWL formalization with the set of defined axioms allows the tractable reasoning and the data retrieval in order to support safety and design decisions management. The development of the proposed DAO is detailed in Section 3 and it aims to bridge the identified gaps in the literature and to particularly deal with the railway domain needs by the systematization of the goal-oriented safety analysis process.

3 The proposed Dysfunctional Analysis Ontology (DAO) for the SCSs design

In order to build the proposed DAO for the SCSs terminology, we apply a Systematic Approach for Building Ontologies (SABiO) [Falbo 2014]. This approach consists of five main phases:

1. Purpose identification and requirements elicitation (Section 3.1);
2. Ontology capture and formalization (Section 3.2);
3. Operational ontology design (Section 3.3);
4. Operational ontology implementation (Section 3.3);
5. Testing (Section 5).

This approach has been widely used for the development of ontologies in several domains such as Software Ontology [de Souza et al. 2017] and Software Process Ontology [de Almeida Falbo and Bertollo 2009]. Moreover, SABiO focuses on using foundational ontologies in the ontology development process in order to ensure the clarity and formality. SABiO phases are based on the main activities of the Requirement Engineering (RE) process life-cycle such as knowledge acquisition, reuse, formalization, etc.

3.1 The DAO Purpose Identification

In the first phase of the SABiO approach, a set of Competency Questions (CQs), which are questions that the ontology must be able to answer [Falbo 2014], are defined to refine the scope of the ontology and to be used in the verification process. Then, the verification and the validation of the proposed conceptual model is performed using validation and verification techniques defined by SABiO.

The proposed DAO for the SCSs design aims to provide an ontological clarification of the dysfunctional analysis terms such as failure, causes, effects and safety measures throughout the SCSs design. The proposed systematization of the dysfunctional analysis is independent of safety analysis methods that are required for the SCSs design. It provides indeed an interoperable view of these methods, since the DAO conceptualization and formalization consider several aspects such as failures of the system components, failures due to the environmental factors (objects) and due to human errors from both the system and its environment perspective. Moreover, the alignment between the dysfunctional analysis and the goal-oriented safety measures elicitation is performed by the interpretation of the relations between derived safety measures and GORE concepts such as goal, context and task. It is proposed through a conceptual model

grounded in UFO. The conceptual harmonization of both failure and its surrounding concepts is based on an extensive extraction of international standards definitions and on the reuse of reference models of both system engineering and railway domain knowledge since it is our application domain. In this context, CQs are elicited regarding the UFO-driven conceptual modeling as follows:

- **CQ1:** What is a failure?
- **CQ2:** How can a failure occur?
- **CQ3:** What are the situations that result from a failure?
- **CQ4:** What is a safety measure?
- **CQ5:** How to link a safety measure to the goal, context and task concepts?

3.2 The Dysfunctional Analysis Conceptualization

In the remainder of this paper, concepts and relations between them are respectively written in **bold** and in *italic* styles in order to improve readability. In the context of collaborative decision-making, the conceptual modeling is a preliminary activity to provide an understandable representation of a knowledge domain based on real world assumptions. The main idea of this paper is to provide a new common conceptualization with grounded elements of dysfunctional analysis and to systematize its early integration in the SCSs design process. The real world interpretation aims to establish relations between the dysfunctional analysis domain and the foundational distinctions of UFO.

The proposed conceptual model of DAO grounded in UFO improves the conceptual clarification of the safety reasoning and the early safety decisions management process. This ontological description of the dysfunctional analysis knowledge answers the **RQ1**. The knowledge capture is based on the railway domain knowledge and standards definitions, which makes the DAO taxonomy flexible and reusable for several SCSs. Table 1 summarizes the proposed taxonomy of dysfunctional analysis for SCSs in order to make it clear and reusable for other domains. In the conceptual modeling stage, highly-expressive languages should be used to create a reference ontology in order to approximate as well as possible the ideal representation of a domain. Figure 3 shows the conceptual model of the proposed DAO using the ontologically well-founded language of conceptual modeling OntoUML. Figure 4 shows the DAO fragment of relations between **Safety Measures** and RE concepts in order to ensure the multi-view modeling and to assist the safety management. The alignment between the proposed goal-oriented fragment and the dysfunctional analysis aspect is established in order to fulfill the **RQ2**. The goal-oriented perspective conceptualization is based on the reuse of a fragment of a reference model of GORE [Negri et al. 2017]. This reference domain ontology is grounded in UFO and allows the interoperability between GORE approaches.

Concepts	Definitions
Exposure	An **Exposure** is a *subtype of* **Disposition** (a special type of **Moment**). It denotes the **Exposure Moment** which *inheres in* **Objects** and is *activated by* the **Hazardous State** (a *subtype of* **Situation**).
Hazardous usage	A **Hazardous usage** is a *a subtype of* **Exposure**. It depicts the **Moment** in which the **Stakeholder** performs a hazardous manipulation and it is *manifested by* the **Stakeholder-caused Failure**.
Defect	A **Defect** is a *subtype of* **Exposure**. A **Defect** denotes a **Fault** when it is *manifested by* a **Fault emergence Failure**. A **Fault** *subsumes* an **Environment Object Fault** and a **System Equipment Fault**.
Fault emergence Failure	A **Fault emergence Failure** is a *subtype of* a **Failure**. It represents any **Failure** caused by an **Object Fault**.
Erroneous Stakeholder Action	An **Erroneous Stakeholder Action** is a *subtype of* **Stakeholder Action**. It represents any erroneous action performed by the **System** and the **Environment Stakeholders**. It *causes* a **Stakeholder-Caused Failure**.
Stakeholder-caused Failure	A **Stakeholder-caused Failure** is a *subtype of* a **Failure**. It *subsumes* a **Non-intentional Stakeholder-caused Failure** and an **ill-intentional Stakeholder-caused Failure**.
Non-intentional Stakeholder-caused Failure	It denotes any **Stakeholder-caused Failure** that *is led by* a **Stakeholder False Belief**.
ill-intentional Stakeholder-caused Failure	It denotes any **Stakeholder-caused Failure** that *is led by* a **Stakeholder ill-intention**.
Stakeholder False Belief	It denotes a **Situation** in the Stakeholder's cognitive model that he believes to be true. However, it is based on wrong **Assumptions**.
Stakeholder ill-intention	It represents the *internal commitment of* the **Agent** to act towards the goal by a plan to accomplish it. However, it is not a malicious intent but the associated plan is wrong and don't satisfy the intended goal.

Table 1: *The DAO concepts definition*

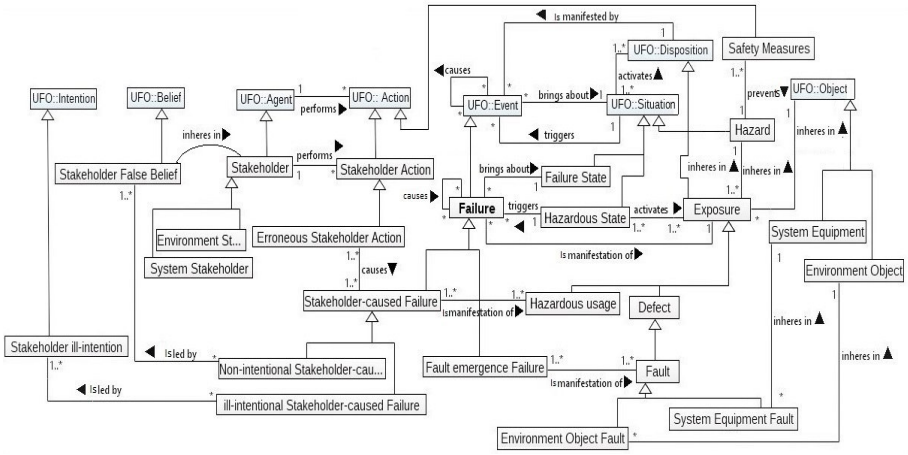

Figure 3: *Conceptual Model of the Dysfunctional Analysis Ontology for SCSs*

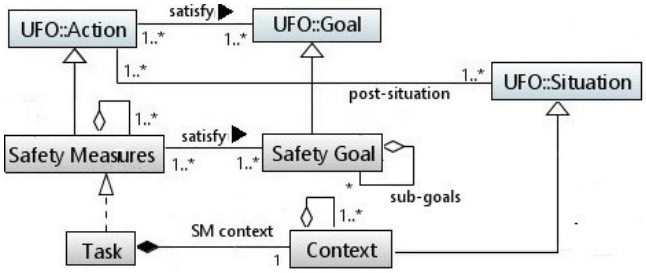

Figure 4: *DAO fragment showing relations between safety measures, goal and context concepts*

As the proposed DAO is grounded in UFO, it provides a common and an unambiguous vocabulary with a high level of abstraction of dysfunctional analysis related concepts. Furthermore, it shows a good completeness, consistency and flexibility of concepts, which aim to deal with the complexity of socio-technical and environmental terms. Once the conceptual model is well-established and grounded in UFO, it is possible to make an OWL formalization of DAO, as presented in Section 3.3, which allows a better reuse and understanding of the domain representation. The expressive power of OWL supports a wide range of inferences and makes DAO efficient for semantic annotation and reasoning.

3.3 The OWL formalization

In order to have an operational version of the reference ontology and increase its reuse, the conceptual specification should be transformed in a machine-readable language. In this section, we discuss the classes and properties of DAO based on the previously described conceptual view. Then, the DAO design pattern is formally encoded using the Web Ontology Language (OWL). In order to enforce the system behaviour and constrain the proposed taxonomy, we make use of the Description Logics (DL) notation [Hitzler et al. 2009] for the axioms specification, since this improves their readability and understandability. In this paper, the DAO pattern is encoded using the logic fragment DLPE presented in [Carral et al. 2013], which allows a tractable reasoning. This tractable reasoning provides indeed an efficient implementation of DAO.

The central concept of this ontology is **Failure** since it is the "core" of dysfunctional analysis and its occurrence leads to many problems in the SCSs design. As defined in the standards [IEEE 610.12 1990], [IEEE 1012 2016] and in the literature [Johnson 2003], a **Failure** is an **Event**. Fortunately, the basic concepts provided by UFO allow a better understanding of how a failure occurs as an event during the operational phase of SCSs. In a system context, a **Failure** is considered as an event, in which a system or a component is unable to perform its required function as it is intended to. In other words, each event that conflicts with the agents goals and violates the whole safety state is considered as a failure. Moreover, as an **Event** characteristic, a **Failure** can *cause* other failures in a chain of events as a cascading failure. For instance, the defect of the railway signalling system can cause the defect of the track-circuit and all related subsystems. According to UFO, the causality relation **R** is declared a *strict partial order* relation [Guizzardi et al. 2013]. Hence, **R** is irreflexive, asymmetric and transitive and these properties are described as following using DL:

- **R** is irreflexive: $\top \sqsubseteq \neg\exists \mathbf{R}.Self$;
- **R** is asymmetric: $\exists(\mathbf{R} \sqcap \mathbf{R}^{-}).\top \sqsubseteq \bot$;
- **R** is transitive: $\mathbf{R} \circ \mathbf{R} \sqsubseteq \mathbf{R}$;

Axioms related to DAO are specified in order to constrain the proposed taxonomy using DL statements as follows:

$$Failure \sqsubseteq Event \sqcap \exists bringsAbout.FailureSate \sqcap \forall causes.Failure \tag{1}$$

$$FailureState \sqsubseteq Situation \sqcap \forall bringsAbout^{-}.Failure \tag{2}$$

$$HazardousState \sqsubseteq Situation \sqcap \forall triggers.Failure \sqcap \forall activates.Exposure \tag{3}$$

$$\top \sqsubseteq \leq 1 bringsAbout.\top \tag{4}$$

$$\top \sqsubseteq \leq 1 triggers^{-}.\top \tag{5}$$

$$causes \circ bringsAbout \sqsubseteq bringsAbout \tag{6}$$

$$causes \circ triggers^{-} \sqsubseteq triggers^{-} \tag{7}$$

$$triggers^{-} \sqsubseteq isTriggeredBy \tag{8}$$

As an **Event**, a **Failure** is related with two different **Situations** as enforced by Axioms (1), (2) and (3):

1. The situation that exists before the occurrence of the Failure is represented as a **HazardousState** that *triggers* the **Failure**. It indicates the situation (the state of being exposed to a risk) that *activates* the **Disposition** (the **Exposure**) that will be manifested in that failure.

2. The situation that is caused by the occurrence of the **Failure** when it *bringsAbout* a **FailureState**.

As an **Event**, a **Failure** transforms the state of affairs of a reality to another. The pre-situation consists of the existence of the **Disposition** to manifest the **Failure**, but the **Failure** does not occur if the **Disposition** is not activated. In the post-situation, the **Failure** is *triggered* and there is a transformation of the reality to the situation in which the system cannot perform its intended functions. DL Axioms (4) and (5) enforce the functionality of properties and automatize: **1)** a **Failure** *bringsAbout* at most one **FailureState**, **2)** it *isTriggeredBy* at most one **HazardousState**. The *causes* property is declared to be transitive and asymmetric, then *brings about*(f_1,*fs)* is entailed if *causes*(f_1,f_2*)* and *brings about*(f_2,*fs)* are the case for any individual f_2. This role chain is automatically generated due to Axiom (6). Similarly, *triggers*$^{-}$(f_1, *hs*) is entailed if *causes*(f_1,f_2*)* and *triggers*$^{-}$*(*f_2,*hs)* are the case for any individual f_2 as enforced by Axiom (7). The functionality of properties *bringsAbout* and *triggers*$^{-}$ prevent the creation of incorrect instances of the *causes* property. The restrictions stated by Axioms (4) to (7) are defined in order to retrieve and query about all existing **Failures** caused by a given **Failure** which *isTriggeredBy* (respectively *bringsAbout*) a given **HazardousState** (respectively **FailureState**). The *isTriggeredBy* property is defined as the inverse of *triggers* (8).

Axioms related to the subtypes of **Failure** and **Exposure** are specified below:

$$StakeholderCausedFailure \sqsubseteq Failure \sqcap \forall isManifestationOf.HazardousUsage \sqcap \forall causes^{-}.ErroneousStakeholderAction \tag{9}$$

$$FaultEmergenceFailure \sqsubseteq Failure \sqcap \forall isManifestationOf.Fault \sqcap \neg \forall causes^{-}.ErroneousStakeholderAction \tag{10}$$

$$Exposure \sqsubseteq Disposition \sqcap \exists inheresIn.Object \sqcap \exists inheresIn.Hazard \tag{11}$$

$$SystemEquipment \sqcap EnvironmentObject \sqsubseteq Object \tag{12}$$

$$HazardousUsage \sqcap Defect \sqsubseteq Exposure \tag{13}$$

As socio-technical systems, railway systems and their safety management involve human operators, components failures, dysfunctional interactions among system components, or even environment/external disturbances. In this paper, these aspects are considered in the conceptualization of the safety reasoning. A **Failure** *subsumes* two different subtypes: **StakeholderCausedFailure** and **FaultEmergenceFailure**. The former is a **Failure** that is directly *causedBy* (the inverse of *causes*) **StakeholderActions** as enforced by Axiom (9). The latter represents a **Failure** that *isManifestationOf* a **Fault** and it is not *causedBy* **Stakeholder Actions** as given by Axiom (10).

An **Exposure** represents the **Dispositions** that are existentially dependent to **SystemEquipments** and **EnvironmentObjects** as enforced by Axiom (11). Here, **Environment Objects** represent objects that are not related to the system and that exist in the system environment. Since railway systems are socio-technical systems, this aspect has to be considered in order to satisfy some specific real situations. The **Exposure** concept *subsumes* two sub-concepts represented as **Defect** and **HazardousUsage** by Axioms (12) and (13). They represent the type of **Disposition** that can *be activated* and *manifested by* **Failures**. From this context and based on Axioms (11) and (13), a **Defect** is a type of **Exposure** that *inheresIn* **Objects**. When it *isManifestedBy* (the inverse of *isManifestationOf*) a **FaultEmergenceFailure**, a **Defect** denotes a **Fault**. By enforcing the transitivity of the subsumption relationship, a **Fault** is considered as a *subtype of* a **Disposition** which *isManifestedBy* a **Failure**.

Axioms (14) to (16) state the different types of **Fault** as follows:

$$Fault \sqsubseteq Defect \sqcap \forall isManifestedBy.FaultEmergenceFailure \tag{14}$$

$$EnvironmentObjectFault \sqsubseteq Fault \sqcap \forall inheresIn.EnvironmentObject \tag{15}$$

$$SystemEquipmentFault \sqsubseteq Fault \sqcap \forall inheresIn.SystemEquipment \tag{16}$$

A **Fault** *subsumes* two distinct types: **SystemEquipmentFault** and **EnvironmentObjectFault**. As **Dispositions**, these types of **Fault** *inheresIn* **SystemEquipment** and **EnvironmentObject**. Otherwise, a **Fault** is a property of **Objects** which *isActivatedBy* (the inverse of *activates*) a specific situation. Moreover, the **HazardousUsage** is a subtype of **Exposure** that can exist in **Objects** with Axioms (11) and (13). It denotes the case in which it *isManifestedBy* a **StakeholderCausedFailure** as enforced by Axiom (9).

Axioms related to the **Stakeholder** types and their **ErroneouStakeholderAction** are stated below:

$$SystemStakeholder \sqcap EnvironmentStakeholder \sqsubseteq Stakeholder \tag{17}$$

$$Stakeholder \sqsubseteq Agent \sqcap \forall performs.StakeholderAction \tag{18}$$

$$ErroneousStakeholderAction \sqsubseteq StakeholderAction \sqcap \forall causes.StakeholderCausedFailure \tag{19}$$

Based on closed-world assumptions of the application domain, a **Stakeholder** *subsumes* two subtypes: **SystemStakeholder** and **EnvironmentStakeholder** (17). A **Stakeholder** *performs* an **ErroneousStakeholderAction** that *causes* **StakeholderCausedFailure** as enforced by Axioms (18) and (19).

Axioms related to the intentional properties of **Agent** are specified by the following DL statements:

$$NonIntentionalStakeholderCausedFailure \sqsubseteq StakeholderCausedFailure \sqcap \forall isLedBy.StakeholderFalseBelief \tag{20}$$

$$IllIntentionalStakeholderCausedFailure \sqsubseteq StakeholderCausedFailure \sqcap \forall isLedBy.StakeholderIllIntention \tag{21}$$

According to UFO, **Actions** *are led by* **Agent**'s **Intention** or **Belief**. An **Intention** is always associated to a plan to satisfy a **Goal**. However, a **Belief** is based on **Assumptions** as situations in the environment that the **Agent** believes to be true. In other words, they represent a **Belief** that a **Situation** exists in the environment. If those assumptions are wrong, they lead to situations that do not satisfy the **Goal** [Negri et al. 2017]. In the SCSs context, we consider that **StakeholderCausedFailure** subsumes two subtypes: **IllIntentionalStakeholderCausedFailure** and **NonIntentionalStakeholderCausedFailure**. In the railway domain, involved Stakeholders have the responsibility to ensure both the system and passengers safety. But there are some spontaneous errors, made by human operators, defined as a set of human actions that exceed some limit of acceptability [Swain and Guttmann 1983] and may be the significant causes of accidents. From this point of view, we assume that a **StakeholderCausedFailure** can be a **NonIntentionalStakeholderCausedFailure** which *isLedBy* a **StakeholderFalseBelief** as enforced by Axiom (20). For instance, a false interpretation of a situation is a **NonIntentionalStakeholderCausedFailure**. Moreover, there is a case in which a **StakeholderCausedFailure** can be an **IllIntentionalStakeholderCausedFailure** with Axiom (21). For instance, an erroneous behaviour caused by a lack of experience in a specific situation is due to a **StakeholderIllIntention**. This is not a malicious intention but it is due to some factors such as physical conditions or training.

Axioms related to **SafetyMeasures** and their associated relations with other concepts are specified as follows:

$$\begin{aligned} SafetyMeasures \sqsubseteq\ & Action \sqcap \forall hasPart.SubSafetyMeasures \\ & \sqcap \exists prevents.Hazard \sqcap \forall satisfy.SafetyGoal \end{aligned} \tag{22}$$

$$hasPart \circ hasPart \sqsubseteq hasPart \tag{23}$$

$$\top \sqsubseteq \exists hasPart.Self \tag{24}$$

$$\top \sqsubseteq \exists (hasPart \sqcap hasPart^{-}).\bot \tag{25}$$

$$SafetyGoal \sqsubseteq Goal \sqcap \forall hasPart.SubSafetyGoals \tag{26}$$

$$Task \sqsubseteq \exists realizes.SafetyMeasures \sqcap \exists hasContext.Context \tag{27}$$

$$Context \sqsubseteq Situation \sqcap \forall hasPart.SubContexts \tag{28}$$

$$\top \sqsubseteq \leq 1 hasContext.\top \tag{29}$$

$$hasContext \circ hasPart^{-} \sqsubseteq hasContext \tag{30}$$

Then, the **Exposure** disposition as a special type of **Intrinsic Moment**, *inheresIn* a **Hazard**, which is a **Situation** by Axiom (11). Here, it is important to mention that the hazard knowledge conceptualization is not considered in this study. We represent the **Hazard** as a **Situation** resulted from the failures occurrence based on the accidentology knowledge. In this level, **Safety Measures** have to be considered in order to *prevent* the **Hazard** occurrence. **SafetyMeasures** are a set of **Actions** to *be realized* (the inverse of the *realizes* property) within a **Task** in order to *satisfy* the required safety level and then the **SafetyGoal** as enforced by Axiom (22). The composition of **Safety Measures** into sub-safety measures is defined by the *hasPart* property (part-whole). The *hasPart* relation is transitive (23), reflexive (24) and anti-symmetric (25). Furthermore, the *hasPart* property denotes the composition of the **Safety Goal** as enforced by Axioms (26) in order to provide the hierarchy tree between complex safety goals, their composition into simple **SubSafetyGoals**. Then, these sub-safety goals are refined in safety requirements able to be assigned to stakeholders. This hierarchy aspect in the requirement management process will be considered in future works.

The **Task** denotes the realization of **Safety Measures** as illustrated in the conceptual model (Figure 4) by the UML link "*realizes*". Then, the **Task** is carried out according to at least one specific **Context** by the property *hasContext*, as enforced by Axiom (27). We declare the **Context** as a *subtype of* a **Situation** defining circumstances and the validity of **Safety Measures** in the considered **Task** with Axiom (28). The *hasContext* property is defined as the function which associates a **Context** to a given **Task**. Due to Axiom (29), this property is declared to be functional since every **Task** is associated with a single

Context. The **Context** is composed into sub-contexts by the *hasPart* property, if it is considered as a complex set of heterogeneous elements related to the **Context**, such as climatic conditions, the rolling stock and the infrastructure capacities, the previous task in the same conditions, etc. These constraints are enforced by Axioms (28) and (29). Then, we can use the properties *hasContext* and *hasPart* in order to retrieve and query about all existing sub-contexts of the same **Context** which is associated to a given **Task**, as automatically generated due to Axiom (30). Consequently, the functionality of the *hasContext* property (29) and the role chain (30) automatically collapse into one single individual all sub-contexts within the same **Context** that is associated to a given **Task**. These constraints avoid ambiguities in the sub-contexts representation at a given **Task** over the same **Context**.

4 The DAO implementation

The DAO pattern is implemented using Protégé 5.2.0, which is one of the most popular open-source tools for ontology development thanks to its powerful capabilities to support creation, modification and querying of ontologies. Figure 5 shows the implementation of DAO classes on Protégé.

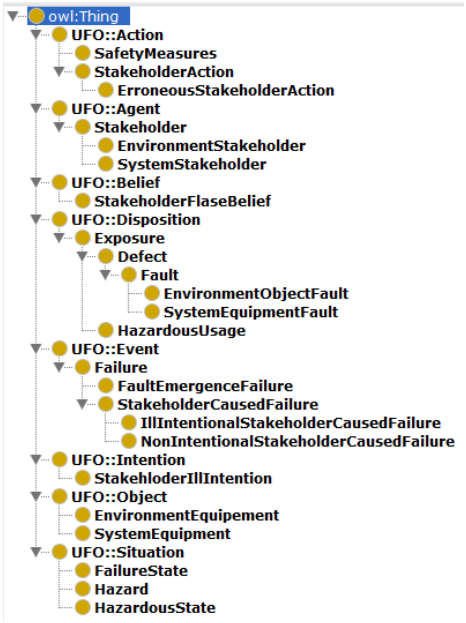

Figure 5: *The implementation of DAO classes on Protégé*

All classes are defined to be disjoint, but they are not shown in the paper in order to improve readability. This is not only a good practice in the OWL formalization, but also it is a necessary condition for the DAO pattern to be expressed in DLPE. Properties as a type of *Object properties* are implemented in order to establish relations between classes as shown in Figure 6.

Figure 6: *The implementation of DAO properties on Protégé*

Then, all cardinalities and the domain range restrictions as shown in the conceptual model are enforced in the OWL declaration. The domain range restrictions have to be considered in order to fill gaps in some scenarios. Axioms (31) and (32) are included as examples in order to show how to enforce these restrictions, where *HazardousUsage* is the range and *StakeholderCausedFailure* is the domain. Axioms enforcing domain and range for other translated classes and properties presented in Figure 3 are extended in the same way. An example of the integrated axioms into the DAO implementation is presented in Figure 7.

$$\exists isManifestationOf.HazardousUsage \sqsubseteq StakeholderCausedFailure \tag{31}$$

$$\exists isManifestationOf^{-}.StakeholderCausedFailure \sqsubseteq HazardousUsage \tag{32}$$

An example of instances (individuals represented by purple diamonds) for the **FaultEmergenceFailure** class is illustrated in Figure 8. The SwitchSystemFailure instance is a type of **FaultEmergenceFailure** and is linked to other instances by properties represented by blue rectangles.

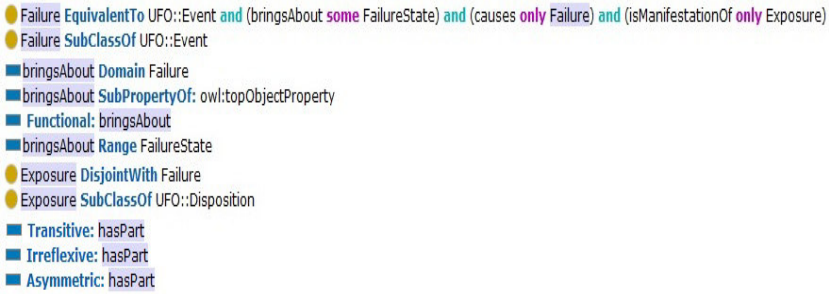

Figure 7: *The implementation of a part of DAO axioms on Protégé*

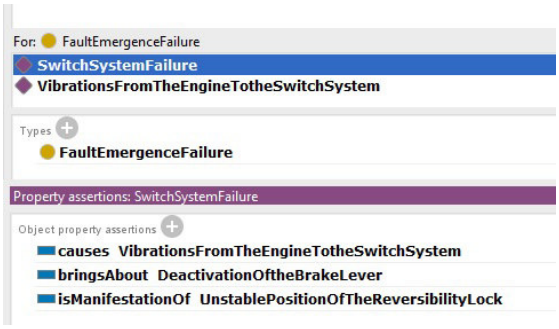

Figure 8: *The instantiation of DAO: an example from individuals of the* ***Fault-EmegenceFailure*** *class*

5 The DAO evaluation

The proposed ontology evaluation process is performed using SABiO verification and validation methods guided by the raised CQs. This process allows a dynamic CQs-driven validation and verification of the DAO behaviour regarding a set of test cases. For the verification step, the concepts management table is established in order to check the ontology ability to answer the competency questions (CQs) mentioned before. Then, the validation aspect will be ensured by the ontology instantiation in order to illustrate real-world situations such as a real railway accident scenario and a case study from a rail remotely-operated task.

5.1 Ontology Verification

This technique aims to ensure the completeness of the proposed ontology by proving the satisfiability regarding its requirement elicitation. Table 2 illustrates the verification results according to the predefined CQs.

CQ	Concepts and Relations
CQ1	A **Failure** is a *subtype of* **Event**. It *brings about* a **Failure State** and a **Hazardous State** *triggers* a **Failure**. As an **Event**, a **Failure** *causes* an other **Failure** (cascading failure). A **Stakeholder-caused failure**, as a *subtype of* **Failure**, is a *manifestation of* a **Hazardous Usage** and is *caused by* an **Erroneous Stakeholder Action**. This failure *is classified into* **Non-intentional** and **ill-Intentional** that are respectively *lead by* **Stakeholder False Beliefs** and **Stakeholder ill-intentions**. A **Fault emergence Failure**, as a *subtype of* **Failure**, is a *manifestation of* a **Fault**. As a *subtype of* **Defect**, a **Fault** can be a **System Equipment Fault** and an **Environment Object Fault**
CQ2	A **Hazardous State**, as a *subtype of* **Situation**, *triggers* a **Failure** and *activates* an **Exposure**, which is a *subtype of* a **Disposition**. This **Exposure** *inheres in* a **Hazard** and is manifested by a **Failure**. This **Exposure** *subsumes* a **Hazardous Usage** and a **Defect**.
CQ3	A **Failure State** is *a subtype of* **Situation** and is *brought by* a **Failure**.
CQ4	A **Safety Measure** is a *subtype of* **Action**. It *is composed into* sub-measures.
CQ5	A **Task** denotes the *realization of* **Safety Measures**. It is associated to a **Context**, which *is composed into* sub-contexts and validates the validity of the **Safety Measure** realization. A **Safety Measure** *satisfies* a **Safety Goal**, which *is composed into* sub-goals.

Table 2: *Verification table: Ontology's CQs and how to fulfil them*

This table may be used as an ontology management tool or as a traceability support in order to deal with the ontology changes made for other domains needs. The proposed ontology provides a complete, non-ambiguous and reusable set of concepts that satisfy the defined ontology purpose.

An automated proof of the ontology consistency has been generated by the *Pellet* reasoner [Sirin et al. 2007]. The ontology reasoning is used to check the consistency of the proposed taxonomy and to obtain the inferred hierarchy of DAO. Furthermore, the expressiveness and clarity qualities are considered as relevant criteria in the verification step. They show how the ontology objectively communicates the meaning of its taxonomy and how this one is expressed with highly-expressive languages in each phase of its development process. The DAO is grounded in UFO in order to provide real world semantics and is represented using the well-founded language OntoUML, which increases its syntax and semantics quality. The OWL formalization increases its understandability and reusability and allows the query answering for the data extraction and consistency

checking. Finally, the verification results show that the proposed ontology fulfils all its raised CQs and covers all intended domain aspects.

5.2 Validation: railway case studies

Railway systems as socio-technical systems require some explanation of the technical aspects. The aim of the evaluation is to demonstrate the DAO relevance to cover complex issues of the railway domain. In this paper, the choice of two case studies has been made and the essential information is detailed in order to improve the understandability and the clarity of the proposed ontology and its relevance for the domain application. However, readers may find further details in the references to our previous work and to some railway literature. A good domain ontology should validate its adaptability criterion to represent several real-world situations, to annotate different data sets and to support the decision-making process in a specific task. For the ontology validation, we refer to a real accident scenario in Longueville (France) in order to illustrate the dysfunctional analysis aspect of the proposed ontology (DAO). Then, we rely on a case study from a remotely operated task representing two different scenarios according to the context related to the **Task** in order to illustrate the goal-oriented safety decisions perspective.

Case 1: The rail accident occurred at Longueville (France) on February 16^{th} 2005 and consisted of a side collision when the train 117710 from Provins (Seine-et-Marne) hit the train 117578 sidelong at Longueville station (Seine-et-Marne). According to [Longueville accident BEA-TT report. 2005], there were no human losses and only material damage was suffered. The damage was to the front carriage over 5 meters, to the impacted locomotive and its chassis, and to the deformation of both the line and the adjacent platform. Figure 9 depicts the accident scenario in the Longueville station.

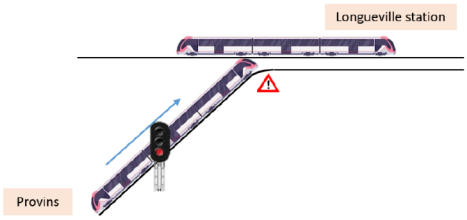

Figure 9: *Illustration of the Longueville accident scenario*

The accident was due to two principal factors:

1. The switch system failure (a *subtype of* **Fault emergence Failure**) is a *manifestation of* **System Equipment Fault**. This consists in the unstable position of the reversibility lock that switches locomotives and activates brakes. Indeed, it was not locked in the "leading" locomotive position (the operating position). Consequently, the vibrations from the engine to the switch system produces the inhibition of the normal brake lever in the locomotive cab (Figure 10).

2. The **ill-intentional stakeholder caused Failure** denotes the lack of the driver's behaviour knowledge in emergency situations. It *is led by* the **driver ill Intention** who only used the locomotive's handbrake to stop at both stations (he did not use the emergency brake control). Consequently, the driver was not able to stop at the intended station and he crossed a closed signal. Unfortunately, the handbrake acted on a single axle (rather than four), which made it impossible to stop the train before the shunting where the train 117578 was stationary (Figure 11).

Figures 10 and 11 show respectively the Resource Description Framework (RDF) graphs of the principal two factors related to the accident scenario occurrence using the DAO pattern. These graphs are generated in order to visualize the instantiation of the proposed ontology and the integration of different kinds of data sets with the design pattern. Furthermore, they allow the data query by the Simple Protocol and RDF Query Language (SPARQL). In RDF graphs, rectangles represent entities and circles represent classes of DAO.

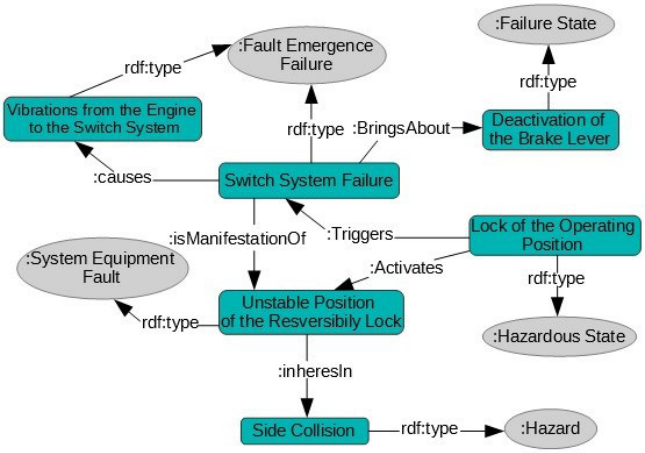

Figure 10: *RDF graph of the first factor related to the occurrence of the Longueville accident scenario*

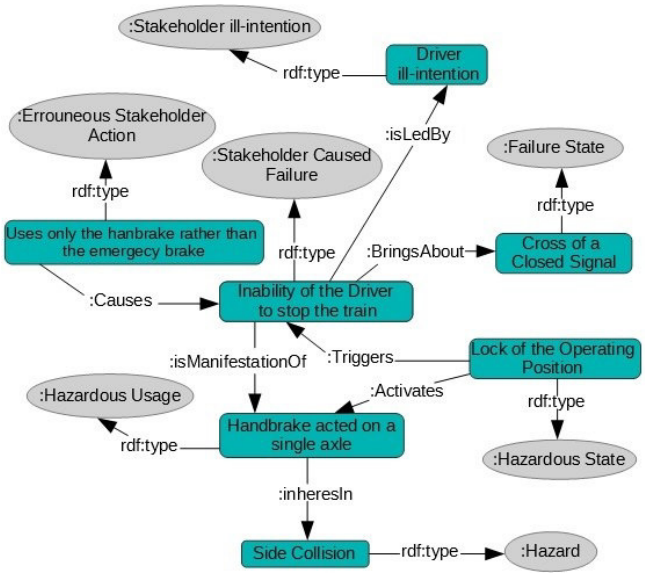

Figure 11: *RDF graph of the second factor related to the occurrence of the Longueville accident scenario*

The 117710 train agent had a **False Belief** concerning the drivers behaviour when the train did not stop at the intended Sainte-Colombe-Septveille station. Indeed, the train agent believed that the train would stop since he perceived the train slowing (he did not imagine the critical situation of the drivers behaviour). Nevertheless, there was no communication between the driver and the train agent (the train was not equipped with an inter-comm locomotive-train). Consequently, the lateral collision occurred at a low speed of 20 km/h for the train 117710. This is a secondary factor that indirectly contributes to the accident occurrence. This accident scenario is considered in order to validate the adaptability of the proposed ontology and it shows that the related taxonomy is flexible and able to represent critical situations. Then, the complete set of the proposed concepts and the consistent relations between them show that the DAO conceptual model can cover and analyse several critical situations.

After analysing the scenario description and the accident investigation detailed in [Longueville accident BEA-TT report. 2005], we may intuitively propose some **Safety Measures** such as the deployment of an electric control of the reversibility system in order to *satisfy* the switch of locomotives and correctly activate the brakes (**Safety Goal**).

Figure 12 depicts the RDF graph of the safety decisions management regard-

ing this accident. Furthermore, a good knowledge of the professional behaviour could have been efficient in order to prevent this critical situation. Consequently, there is a need to maintain a good communication between involved actors to have an overall safe system view. Here, we invoke the significance of requirements traceability, particularly for safety functions. This aspect will be considered in future works.

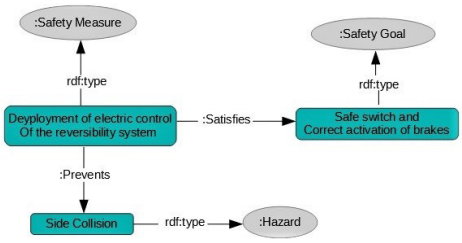

Figure 12: *RDF graph of the goal-oriented safety measures developpment*

SPARQL queries are performed on RDF graphs generated from DAO in order to investigate data query. Figure 13 shows an example of a SPARQL query which extracts the technical factor that causes the side collision.

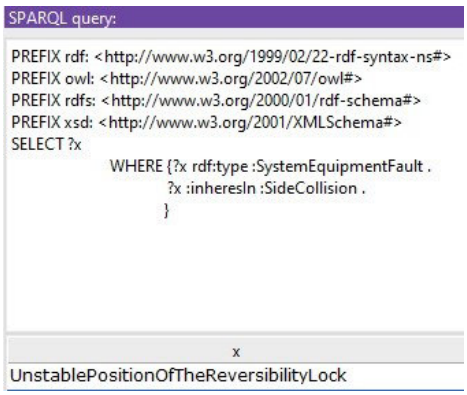

Figure 13: *SPARQL query and result form RDF dataset related to the Longueville accident*

The consistency of DAO and its instantiation is tested by reasoning on dysfunctional analysis. This allows the datasets centralisation in order to assist the safety decisions management process. Furthermore, it systemises the experience

feedback (REX) for the development of future systems. As shown in Figure 14, the SPARQL query checks the safety measure that satisfies the safety goal stated as "stop the train". The obtained result refers to the **Safety Measure** that satisifies this **SafetyGoal**. The *use of emergency brakes* may be applied in all contexts to prevent any **Hazard**. Furthermore, this is recommended by the frame of reference of main-line train drivers.

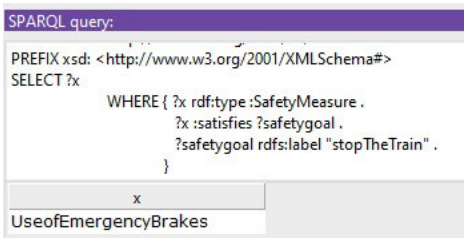

Figure 14: *SPARQL query and result about a specific safety measure*

Case 2:

The case study from a remotely-operated **Task** preparation is considered in order to illustrate the relevance of the context concept and to show how the task performance can be adapted to a specific context. The scenario is described in the study report [Debbech et al. 2018d] and it denotes a nominal scenario defining the task performance in classic circumstances after being aware of the **Context**. The degraded scenario represents the **Task** performance based on the the related **Context** and by considering a set of **Safety Measures**. Furthermore, **Safety Measures** (as a *subtype of* **Action**) are based on **Train driver intentions** since they are associated to a plan to be performed (**Task**). The considered **Safety Measures** should satisfy the whole **Safety Goal** which consists in the safe crossing of the incline.

In this scenario, the **Hazardous State** denotes the low adherence of the train. It *triggers* the inability of the train to move in the middle of the incline as a **Fault emergence Failure**) and it *activates* the **Exposure**, which denotes that the train is carried away by its weight in the middle of the incline as a **Fault**. Consequently, it *inheres in* the **Hazard**, which is in this case the potential drift of the train. Furthermore, this **Fault emergence Failure** is the *manifestation of* the rolling stock capacities and its constituents fault (**System Object Fault**) and/or the weather conditions disturbances such as frost and/or humidity on the rail (Environment Object Fault). The dysfunctional analysis process annotation of the remotely-operated task is represented in Figure 15.

In order to prevent the **Hazard** occurrence, the driver performs this **Task**

by proceeding a set of **Safety Measures** according to the identified **Context**. The **Safety Goal** consists in the safe crossing of the incline. In this scenario, the **Context** awareness C is composed of five elements by the *hasPart* property:

- c_1: The perception of weather conditions;
- c_2: The verification of the train constituents verification such as the device of automatic wedges that is adequate to the considered incline;
- c_3: The verification of the effective capacities of the train;
- c_4: The absence of the device of automatic wedges;
- c_5: The high hazard probability estimation of the non crossing of the incline, namely in the case of a hollow;
- c_6: The perception of the task history of the previous train in the same conditions.

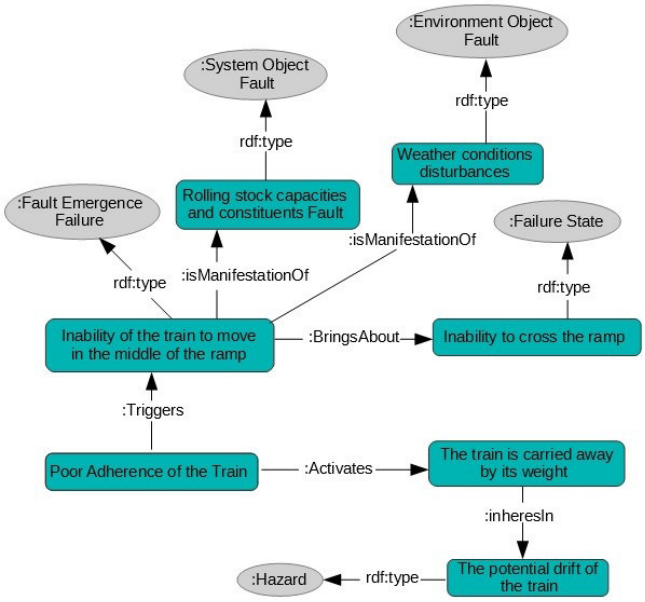

Figure 15: *RDF graph of the dysfunctional analysis process for the remotely operated task*

Consequently, after the perception of the whole **Context** and its parts, the driver carries out his **Task** as follows:

1. The nominal scenario: If the union property of c_1, c_2, c_3 and c_6 is true then the driver ordinarily performs his **Task** in order to *satisfy* the intended **Safety Goal**. The integration of these contexts in the safety decisions process is necessary and sufficient for the classic performance of the **Task**. The set of **Safety Measures** that are respectively adapted to c_1, c_2, c_3 and c_6 are the contact of the warehouse manager (sm_1), the verification of the available documents (sm_2), the brake test (sm_3) and the sm_1 also.

2. The degraded scenario: Else if the perceived context is the union of c_4 and c_5, he contacts the warehouse and the traffic center as the whole **Safety Measure** (sm) in order to ask for either the availability of another locomotive equipped and able to perform the **Task** (sm_4) or the use of a remotely-operated locomotive (sm_5) aiming to safely cross the incline (**Safety Goal**). The sm_5 seems more efficient in terms of physical and logistics constraints since it avoids critical situations caused by **Stakeholder caused Failure**.

The adaptive task performance based on a specific context is relevant aspect in the safety decisions management process since it deals with the dynamic aspect of safety measures. This process is constrained and inferred by Axioms (27) to (30). Figures 16 and 17 depict RDF graphs of the goal-oriented safety decisions management in the remotely operated task by considering respectively the nominal and the degraded scenario.

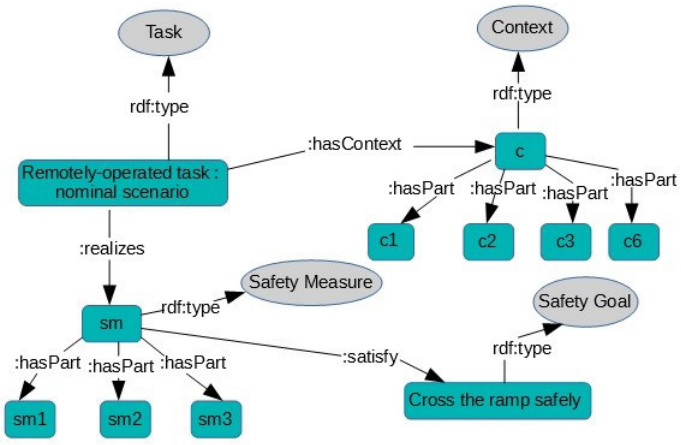

Figure 16: *RDF graph of the nominal scenario of the remotely operated task*

As shown in Figure 18, the SPARQL query tested on DAO asks about dataset (**Safety Measures**) on the RDF graph presented by Figure 17. The data results

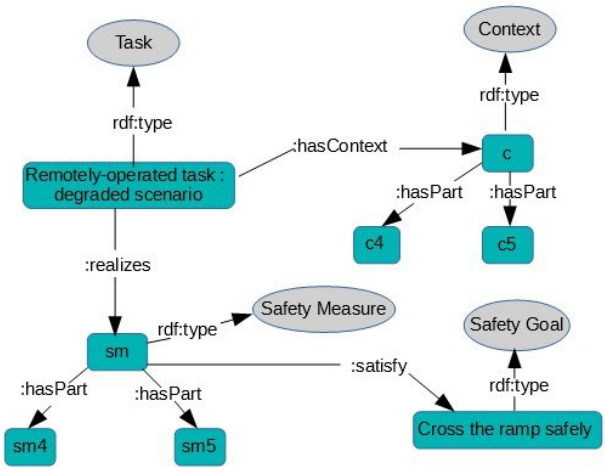

Figure 17: *RDF graph of the degraded scenario of the remotely operated task*

of this query must satisfy two conditions, namely their satisfaction of a specific **Safety Goal** and their applicability in contexts c_4 and c_5.

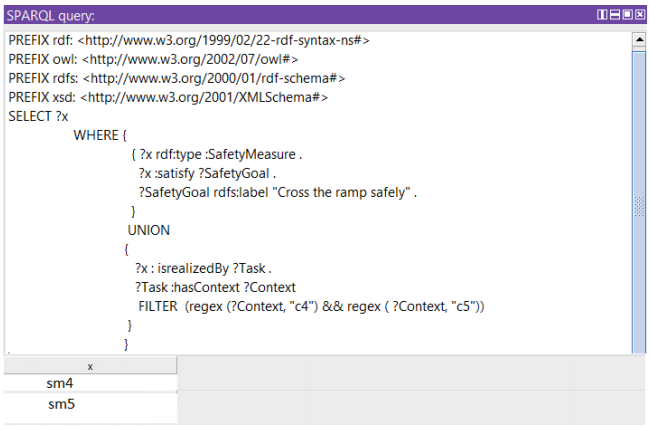

Figure 18: *SPARQL query and result about safety measures that are suitable for specific contexts*

The proposed Dysfunctional Analysis Ontology (DAO) provides a full taxonomy and a conceptual clarification of failures, its causes and effects in terms of technical devices, human operators and the system environment. As founded in

UFO, DAO proposes real-world semantics and covers different aspects of complex critical situations. Then, it shows a capability to represent and analyze real accidents scenarios thanks to its concepts polymorphism and the high level of abstraction of safety analysis. Moreover, the interpretation of safety measures from a goal-oriented view ensures the consistence and the completeness of the safety constraints into the design model. Otherwise, the proposed DAO as a reference domain ontology contributes to the knowledge sharing and the safety decisions management and it can be reused for other safety critical domains.

6 Conclusion and Future Works

In this study, the main contribution consists in proposing a Dysfunctional Analysis Ontology (DAO) which can be, by considering the different criteria mentioned throughout the paper, a reference ontology domain grounded in UFO. It is developed using the SABiO approach, based on the standards definition for the proposed taxonomy and the alignment with the involved knowledge domains such as safety, railway and a part of GORE. Moreover, it establishes a semantic link between the dysfunctional analysis and requirement engineering aspect in order to analyze failure modes, their causes and effects from both the system and the environment perspectives. The DAO development is driven by its raised competency questions and their refinement until the fulfillment of the DAO's scope. The proposed ontology contributes to the knowledge sharing, the conceptual modeling and the goal-oriented safety decisions identification from several perspectives as summarized below.

Firstly, the DAO provides a conceptualization of the failure type, its causes, its effects and the related hazard. This conceptual analysis is based on the use of UFO's foundational concepts and relations between them. Then, it systematizes the ambiguous use of the term *failure* and its related concepts in the safety critical systems terminology. Moreover, three aspects of the dysfunctional analysis are considered in the knowledge conceptualization: **1)** components failures of both system and environment objects and their cascading by the causality relationship, **2)** human errors caused by both system and environment stakeholders, and the type of the **Mental Moment** defining the origin of the errors, **3)** the situation that causes the failure and the one which is caused by the failure occurrence. Therefore, the conceptualization is based on the interoperability of existing dysfunctional analysis methods. Furthermore, the DAO supports the ontological analysis and the conceptual clarification of real-world critical situations.

Secondly, the safety measures considered to prevent the hazard are semantically linked to the GORE perspective in order to provide a shared view between both safety and design actors. This aspect is relevant since the aim of DAO is

to support dysfunctional analysis as soon as possible in the first design stages of SCSs. Besides, a conceptual analysis of GORE concepts such as goal, context and task is performed through the alignment with the safety aspect. In other words, safety notions are considered and integrated from the first design phases of SCSs in order to obtain a safe system behavior. Then, it contributes to the goal-oriented safety analysis process through the goals conceptualization and the safety measures capture and specification.

Thirdly, the DAO establishes a common vocabulary for the knowledge sharing in order to improve the communication and avoid the semantic heterogeneity between the actors of domains. Moreover, the proposed ontology provides a complete and consistent taxonomy, which is able to represent and analyze several real situations thanks to its flexibility, adaptability and expressiveness qualities. The verification and validation process validate the criteria mentioned above by the CQs verification and the annotation of complex case studies from the railway domain. The proposed concepts can be used interchangeably in order to refer to different aspects and types of phenomena. As a reference domain model, DAO may be reused for other safety critical domains since it is based on well-defined standards and real-world semantics.

Fourthly, the operational version of DAO and its OWL formalization is provided since we believe in the good capabilities of this language in terms of the clarity and the reasoning to help the safety decision-making process. The set of axioms are defined in order to constrain the proposed classes and properties and to query about individuals for a specific application. The OWL implementation allows the semantic annotation of the dysfunctional analysis process, the related components in the design model and the safety measures to be considered from the goal-oriented view. Then, the DAO can be used to semantically annotate dysfunctional analysis data from a range of different domains such as the aviation and the automotive domains. The interoperable view provided by the DAO makes it extendible and reusable since the formalization goes beyond the typical concepts and simple relations. The real-world semantics interpretation and the UFO-driven conceptualization allows the integration of knowledge according to the specific needs of the application domain. Moreover, the DAO can be used to annotate and retrieve data according to the required granularity of the application domain.

In future works, we intend to extend the GORE concepts integration such as requirement, agent and the goal nature. Furthermore, we aim to investigate the full safety decisions management process, which ranges from the safety measures specification by the organization until the safety requirements assignment to stakeholders in order to satisfy safety goals. This process will be performed by the integration of new concepts and the safety-oriented reinterpretation of the Organization-Based Control Access model (Or-BAC), which was initially devel-

oped for the Information Systems (IS) security. Then, the requirements management process must be considered in order to deal with the dynamic context-adaptive aspect of the safety decisions management process. The requirement management process will include several perspectives of the requirement engineering such as the requirements traceability, the requirements hierarchy, their satisfaisability, etc. Finally, we plan to establish the UFO-driven alignment between the DAO and the perspectives mentioned above in order to provide a structured and a consistent safety control model for the SCSs design.

Acknowledgements

We thank IRT Railenium for their collaboration in the context of TC-Rail project in order to provide the case study 2 inspired from a remotely operated task.

References

[Arp et al. 2015] Arp, R., Smith, B., and Spear, A."Building Ontologies with Basic Formal Ontology"; MIT Press, (2015).

[Borst 1997] Borst, W.: "Construction of Engineering Ontologies"; PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands (1997).

[Carral et al. 2013] Carral, D., Scheider, S., Janowicz, K., Vardeman, C., Krisnadhi, A. A., & Hitzler, P.: "An ontology design pattern for cartographic map scaling"; Proc. Extended Semantic Web Conference. Springer, Berlin, Heidelberg (2013), 76-93.

[Cheatham et al. 2017] Cheatham, M. A., Ferguson, H., Vardeman, C., & Shimizu, C.: "A Modification to the Hazardous Situation ODP to Support Risk Assessment and Mitigation"; Advances in Ontology Design and Patterns, 32 (2017), 97-104.

[de Almeida Falbo and Bertollo 2009] de Almeida Falbo, R., Bertollo, G.: "A software process ontology as a common vocabulary about software processes"; International Journal of Business Process Integration and Management, 4, 4 (2009), 239-250.

[de Souza et al. 2017] de Souza, E.F., de Almeida Falbo, R., Vijaykumar, N.L.: "ROoST: Reference Ontology on Software Testing"; Applied Ontology, (2017), 1-32.

[Debbech et al. 2018a] Debbech, S., Bon, P., Collart-Dutilleul, S.: "Improving safety by integrating dysfunctional analysis into the design of railway systems"; WIT Transactions on The Built Environment, 181 (2018), WIT press, 399-411.

[Debbech et al. 2018b] Debbech, S., Bon, P., Collart-Dutilleul, S.: "Towards semantic interpretation of goal-oriented safety decision based on foundational ontology"; Journal of Computers, 14(4), (2019), 257-267.

[Debbech et al. 2018c] Debbech, S., Bon, P., Collart-Dutilleul, S.: "A Model-based system engineering approach to manage railway safety-related decisions"; International Journal of Transport Development and Integration, 3(1), (2019), 30-43.

[Debbech et al. 2018d] Debbech, S., Collart-Dutilleul, S., Bon, P.: "Cas d'étude d'une mission ferroviaire télé-opérée, Study report (in french)"; IFSTTAR, November 2018.

[EN-50129 2003] EN-50129: "Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling", (2003).

[Falbo 2014] de Almeida Falbo, R.: "SABiO: Systematic Approach for Building Ontologies"; In: Guizzardi, G., Pastor, O., Wand, Y., de Cesare, S., Gailly, F., Lycett, M., Partridge, C. (eds.): 1^{st} Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. FOIS (2014).

[Firesmith 2004] Firesmith, D.: "A taxonomy of safety-related requirements"; Proc. International Workshop on High Assurance Systems (RHAS'04), 29-30, (2004).

[Guizzardi 2005] Guizzardi, G.: "Ontological Foundations for Structural Conceptual Model"; PhD thesis, University of Twente, The Netherlands (2005).

[Guizzardi et al. 2013] Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R.S.S., Almeida, J.P.A.: "Towards Ontological Foundations for the Conceptual Modeling of Events"; Proc. 32 th International Conference on Conceptual Modeling, Hong Kong, Springer (2013), 327-341.

[Gruber 1993] Gruber, T. R.: "A Translation Approach to Portable Ontologies"; Knowledge Acquisition, 5, 2 (1993), 199-220.

[Heimdahl 2007] Heimdahl, M. P. E.: "Safety and Software Intensive Systems: Challenges Old and New"; Proc. FOSE'07, (May 2007), 137-152.

[Herre et al. 2006] Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., and Michalek, H.: "General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes. Part I: Basic Principles (Version 1.0)"; Pure Collection, (2006), 297-345.

[Hitzler et al. 2009] Hitzler, P., Krotzsch, M., & Rudolph, S.: "Foundations of semantic web technologies"; Chapman and Hall/CRC (2009).

[Hoinaru et al. 2013] Hoinaru, O., Mariano, G., & Gransart, C.: "Ontology for complex railway systems application to ERTMS/ETCS system"; Proc. FM-RAIL-BOK Workshop in SEFM'2013 11th International Conference on Software Engineering and Formal Methods, (2013, September).

[IEC61508 2010] IEC61508: "Functional safety of electrical/electronic/programmable electronic safety-related systems", (2010).

[IEEE 610.12 1990] IEEE: IEEE 610.12: "IEEE Standard Glossary of Software Engineering Terminology", (1990).

[IEEE 1012 2016] IEEE: IEEE 1012: "Standard for System, Software, and Hardware Verification and Validation"; Tech. rep., Institute of Electrical and Electronics Engineers, Inc (2016).

[ISO/DIS 26262-1 2009] ISO/DIS 26262-1: "Road vehicles - Functional safety - Part 1 Glossary"; Tech. rep., (July 2009).

[Johnson 2003] Johnson, C. W.: "Failure in Safety Critical Systems: A Handbook of Incident and Accident Reporting"; Glasgow University Press, (2003).

[Lewis 2012] Lewis, R.,: "A semantic approach to railway data integration and decision support"; Thesis, University of Birmingham, (2012).

[Longueville accident BEA-TT report. 2005] The Longueville accident BEA-TT report (in french), Rapport d'enquête technique sur l'accident ferroviaire survenu à Longueville le 16 février 2005, Rapport technique, Ministère de l'Équipement, des Transports, de l'Aménagement du Territoire, du Tourisme et de la Mer, METATTM. English summary online: http://www.bea-tt.developpement-durable.gouv.fr/longueville-english-summary-a178.html : Accessed on: 1. Oct. 2018.

[Maalel et al. 2012] Maalel, A., Mejri, L., Hadj Mabrouk, H., Ben Ghezela, H.: "Toward a knowledge management approach based on an ontology and Case-based Reasoning (CBR): Application to railroad accidents"; Proc. Sixth International Conference on Research Challenges in Information Science (RCIS), (2012), 1–6.

[Masolo et al. 2003] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A.,: "Ontology Library"; in WonderWeb Deliv. D18, (2003).

[Negri et al. 2017] Negri, P.P., Souza, V.E.S., de Castro Leal, A.L., de Almeida Falbo, R., Guizzardi, G.: "Towards an ontology of goal-oriented requirements"; Proc. CIbSE. Argentina (2017), 469-482.

[Rehman and Kifor. 2016] Rehman, Z., & Kifor, C. V.: "An Ontology to Support Semantic Management of FMEA Knowledge"; International Journal of Computers, Communications & Control, 11, 4 (2016).

[Schön et al. 2014] Schön, W., Larraufie, G., Moens, G., Poré, J.,: "Signalisation et automatismes ferroviaires-Tome3"; Vie du rail, (2014).

[Sigwarth et al. 2015] Sigwarth, T., Loewe, K., Beck, E., Pelchen, L., & Schrader, T.: "Conceptual Ontology of Prospective Risk Analysis in Medical Environments-the OPT-Model-Ontology"; Proc. ICICIS.'15, (Dec 2015), 88-93.

[Sirin et al. 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y.: "Pellet: A practical owl-dl reasoner"; Journal of Web Semantics, 5, 2 (2007), 51-53.

[Swain and Guttmann 1983] Swain, A. D., and Guttmann, H. E.: "Handbook of human reliability analysis with emphasis on nuclear power plant applications"; Tech. Rep., US Nuclear Regulatory Commission, Washington, D.C. (1983).

[Tutcher 2014] Tutcher, J.: "Ontology-driven data integration for railway asset monitoring applications"; Proc. 2014 IEEE International Conference on Big Data (Big Data), IEEE, (2014, October), 85-95.

[Van Gulijk et al. 2015] Van Gulijk, C., Hughes, P., FigueresEsteban, M., Dacre, M., and Harrison, C.,: "Big Data Risk Analysis for Rail Safety?"; Proc. Safety and Reliability of Complex Engineered Systems (ESREL), (2015).

[Zhou et al. 2017] Zhou, J., Hänninen, K., Lundqvist, K., & Provenzano, L.: "An ontological approach to hazard identification for safety-critical systems"; Proc. Second International Conference on Reliability Systems Engineering (ICRSE), IEEE, (2017, July), 1-7.