2017 24th Asia-Pacific Software Engineering Conference

An Ontological Approach to Elicit Safety Requirements

Luciana Provenzano
Bombardier Transportation
Västerås, Sweden

Kaj Hänninen, Jiale Zhou, Kristina Lundqvist
School of Innovation, Design and Engineering
Mälardalen University
Sweden

***Abstract*—Safety requirements describe risk mitigations against failures that may cause catastrophic consequences on human life, environment and facilities. To be able to implement the correct risk mitigations, it is fundamental that safety requirements are defined based on the results issued from the safety analysis. In this paper, we introduce a heuristic approach to elicit safety requirements based on the knowledge about hazard's causes, hazard's sources and hazard's consequences (i.e. hazard's components) acquired during the safety analysis. The proposed approach is based on a Hazard Ontology that is used to structure the knowledge about the hazards identified during the safety analysis in order to make it available and accessible for requirements elicitation. We describe how this information can be used to elicit safety requirements, and provide a guidance to derive the safety requirements which are appropriate to deal with the hazards they mitigate.**

***Keywords*—Safety requirements; safety requirements elicitation; ontologies; hazards; hazard's components.**

I. Introduction

Safety requirements are the safety measures taken to mitigate hazards in safety-critical systems. This implies that safety requirements describe means of avoiding, reducing or limiting failures [6][7] that may cause a system to fail with catastrophic consequences on human life, environment or facilities. Safety requirements are typically identified based on a list of "*categorized hazards and associated safety risk analysis*" [8]. Safety requirements describe measures against failures. So, the information issued from safety analysis, i.e. the knowledge about hazards and their sources, causes and consequences, is essential to elicit the appropriate safety requirements since it describes how system's failures happen [18]. This knowledge, discovered and owned by the safety team, is not always easy to access in order to be used to elicit and specify safety requirements. As a result, efforts done by the safety team "*are usually not integrated into the requirements specifications, and this makes it difficult to ensure that the architecture incorporates the appropriate safeguards*" [8].

Requirements elicitation concerns "*getting the right requirements*" [19]. This means that the main focus is not just to collect requirements but to have a better understanding of the requirements [20]. This is not a straightforward task due to the different requirements' sources and system needs that have to be managed, and the different techniques that can be used [20]. Requirements elicitation for safety-critical systems further implies that explicit safety analysis is incorporated into the requirements process [21]. So, requirements elicitation must take into account potential hazards and risks, and ensures that the final specification does not violate the safety behavior of the system [22].

We think that the information needed to elicit safety requirements should be searched among the three hazard's components, as described by the hazard's triangle [10], and their relations. Based on this observation, we propose a new Safety Requirements Elicitation (SARE) method based on a Hazard Ontology (HO). The HO is used to capture the knowledge about hazards identified during the safety analysis, and structured it as hazard's components and relations among them. This information is then used by the SARE approach to create a list of questions which guide the analyst in the requirements elicitation. Answering the questions is a part of the elicitation process to capture the safety requirements that are correct with respect to the hazard they are supposed to mitigate. The rest of this paper is structured as follows: Section II presents background information for this work. Section III describes the proposed approach for safety requirements elicitation. Section IV introduces the related work of this study. Conclusions and future work are presented in section V.

II. Background

In system safety, a hazard consists of three components, i.e. hazard source, initiating mechanism (causes), and target/threat outcome (consequences). The three components of the hazard form the so-called hazard triangle. "*All three sides of the triangle are essential and required in order for a hazard to exist*" [10]. The three hazard's components are fundamental to determine where to mitigate a hazard. In fact, by removing any one of the triangle sides the hazard becomes incomplete and, as a such, the hazard is eliminated since it cannot produce a mishap. Also, by reducing the probability of the initiating mechanism the mishap probability is reduced. Finally, by reducing a hazard element or the target/threat outcome than the mishap severity is reduced.

The knowledge about hazard's sources, hazard's mechanisms and hazard's outcomes, issued from safety analysis, is therefore fundamental for the safety requirements elicitation. Indeed, it provides a deep understanding of how and why hazards happen and, as a consequence, gives an insight of the correct system behavior that should be described in the safety requirements to mitigate a hazard. The main difficulty associated with it is to understand exactly what is or may be a hazard source, or an initiating mechanism or an outcome, and where to find this information. This situation is made more difficult by the fact that

978-1-5386-3681-7/17 $31.00 © 2017 IEEE
DOI 10.1109/APSEC.2017.91

hazards can be expressed ambiguously and lack precise information about their components (i.e. source, causes and consequences). Some research works [4][5][9], mostly based on ontologies, have been done to provide a precise definition of "what a hazard is", to remove the ambiguity in the hazard description.

To overcome this problem, we proposed in our previous works [1][2][3] a Hazard Ontology (HO) which aims at providing a conceptualization of the hazard to increase the knowledge of the details and circumstances which result in hazards, and the operating context in which hazards can occur. For each hazard's component, the HO defines additional entities to describe in-depth the logic and mechanisms that lead to causes, sources and consequences. This contributes to improve the understanding and definition of what hazard's components are. So, the HO can be used to capture thoroughly the knowledge about hazards acquired during the safety analysis. This knowledge is clearly structured through the HO entities to make it accessible and usable in the sub-sequent development phases. The approach to elicit safety requirements that we propose aims at using the knowledge about hazards captured in the HO to guide the safety requirements elicitation, as explained in section III.

III. The Ontological Approach to Elicit Safety Requirements

A. Description of the application scenario

To describe the approach, we use the Parking Brake (PB) function implemented to mitigate the "Collision Hazard" (C-H) for a high-speed train. The PB is designed to prevent the train from moving away when at standstill. The PB can be manually activated by pushing a PB push button in the driver's cabin. The PB is then applied if previously released, and released if previously applied. The application of the PB causes a bi-stable electro-valve installed on each bogie to open the relevant pneumatic brake line to apply the parking brake. The unintended release of the PB can cause a collision hazard since the train may roll away and collide with surrounding persons (personnel or passengers) and/or objects, such as other trains or depot's walls.

The example we consider, concerns the behavior of the PB lamp installed on the PB button, and used to indicate the PB status. The PB lamp is controlled by the Train Control System (TCS) which is a real-time on-board system in charge of the execution of the train control functions. The PB lamp indication is one of the risk mitigations at software system level for the C-H. The PB lamp has the following behavior: 1) it is turned on when the PB is applied; 2) it is turned off when the PB is released; 3) it flashes when the PB status is unknown.

B. From hazard's components to safety requirements

Based on the findings from our previous works [1][2][3], we observed that hazard's components concern "objects" in both the application environment and the safety-critical system, that participate to a hazard due to their properties and roles. Identifying these objects and classifying them as causes, source or consequences leads to the understanding of how a hazard may occur. As a result, the identified objects, their properties and their roles will be the basis for the safety requirements elicitation. The objects that compose hazards are typically identified and discussed during safety analysis, and can also be obtained from the safety analysis documents, the system description and architecture, the software system description and architecture, the domain description, and the experts' knowledge. In the PB example, possible objects are the PB, the PB lamp, the TCS, the train, etc.

We suggest some insights based on our experience concerning the objects, their properties, roles and relations as a guidance to identify and classify them:

- Each object has some properties. For example, the PB lamp can be burnt out, or the TCS is subjected to input failures. Object's properties provide a first understanding of which hazard's component an object belongs to. The PB lamp, for instance, is more likely to be part of a hazard's source or a hazard's cause than part of the hazard's consequences. The maintainer, on the contrary, is more likely to be a victim, i.e. to belong to the hazard's consequences, since he/she can be injured if hit by a train that moves. Properties can be weaknesses or strengths for a given object with respect to a hazard. Objects which show a weakness can be considered as vulnerable objects with respect to the hazard, while objects with strengths may be dangerous objects.
- Each object plays one or more roles in the hazard. For example, one of the possible roles played by the TCS is the "PB lamp controller" role, while the PB lamp's role is "PB status indicator".
- Objects contribute to a hazard with their role and properties. Roles and properties are strictly dependent on each other. In fact, an object's property can determine the object's role, and vice versa, the object's role can influence the object's property. For example, the property "subjected to input failures" depends on the fact that the TCS plays the role of "PB lamp controller". On the other hand, the fact that the PB lamp plays the role of "PB status indicator" whose behavior is determined by the PB lamp controller, makes it have the property of "being passive" about the information it conveys. This implies that the weaknesses of an object depend on the role played by the object itself. The TCS, which controls the PB lamp (object's role), is subjected to erroneous inputs (object's weakness) coming from the PB valve sensors.
- Understanding how objects participate to a hazard due to their weaknesses and roles provides the explanation to the hazard's causes. The computation of the PB status based on incorrect inputs may be a possible cause of a wrong indication of the PB status through the PB lamp. The wrong indication of the PB status may lead to an unintended release of the PB which, as a consequence, may cause a train to roll away.
- Discovering that objects with weaknesses (vulnerable objects) interact with objects with strengths (dangerous object) explains the hazard's consequences. The maintainer (vulnerable object) can be injured by the train (dangerous object) that moves.
- Identifying dangerous objects means to discover the hazard's source. The fact that the train cannot be standstill without braking, implies that the train becomes a danger in case the brake is not applied.

- Interactions between two or more objects are realized through a relation. The maintainer can be hit by the train because of the relation "Move towards" which exists between the object "train" and object "maintainer".

It's worth noting that the hazard's causes are indeed complex mechanisms (Initiating Mechanism) which consist of unfortunate events that happen in a particular situation. The situation is originated by the interactions of some objects in the application environment and/or in the safety system that present some weaknesses related to their role. As an example, the "indication of a faulty PB status" is the situation originated by a weakness of the object "TCS". The "unintended release of the PB" is the unfortunate event that may happen due to this situation.

C. The heuristic Safety Requirements Elicitation (SARE) approach

Based on the above considerations, the safety requirements shall be elicited by reasoning on how to: 1) overcome the properties that are weaknesses for a given object with respect to the hazard (SARE-ACT1 activity); 2) change, add or remove the object's role (SARE-ACT2 activity); 3) cut off existing relations in order to remove the connections among two or more objects (SARE-ACT3 activity). The steps to be followed and the questions that are proposed to the analyst for the safety requirements elicitation are explained in Fig. 1.

D. SARE-ACT1: overcome an object's weakness

We define the object's weakness as the property of an object to be vulnerable to something which implies that the object does not behave as expected when playing its role. In the PB example, the TCS might compute a wrong PB status since it may receive incorrect inputs. So, "incorrect inputs" is a weakness related to the object's role "PB lamp controller". Based on this definition, overcoming an object's weakness implies that we need to find a way to eliminate it or, if it is not possible, reduce its effects, as shown in Fig. 2. The way in which the object's weakness is overcome represents the safety requirement/s to be elicited.

Pre-condition
The hazards shall be structured as objects, objects' roles, objects' properties, and relations among hazard's components, i.e. the HO is available.
Trigger
<Hazard's description>, i.e. a sentence which describes the hazard
Main flow
1. The analyst obtains all the HO instances which correspond to the hazard's description, i.e. HAZARD = <hazard's description>
2. For each HO instance, the analyst obtains the object along with its role and its property which belong to the hazard's causes, i.e. ENVIRONMENT OBJECT=<object>, INITIATING ROLE=<role>, INITIATOR FACTOR=<weakness> from HO
3. The analyst performs the SARE-ACT1
4. The analyst performs the SARE-ACT2
5. The analyst obtains the objects with weakness and strength and their relation, and the relation between event and hazard, i.e. MISHAP VICTIM=<victim>, EXPOSURE=<exposure>, HAZARD ELEMENT=<danger>, HARM TRUTHMAKER=<source>, INITIATING EVENT=<event>, and HAZARD=<hazard> from HO
6. The analyst performs the SARE-ACT3
7. Go to step 2

Fig. 1. The Safety Requirements Elicitation (SARE) heuristic approach

In order to perform this activity, the analyst needs to know: a) which is/are the object/s with weakness; b) for each identified object, which is/are its weaknesses; c) for each identified object, which is its role. It's worth noting that the objects with weaknesses that are considered in this activity are those that belong to the hazard's causes. Moreover, due to the fact that objects' properties and roles are strictly dependent on each other, the knowledge about the object's role is a fundamental information to overcome its weakness. The HO (see [3] for details) provides the analyst with these pieces of information through the entities ENVIRONMENTAL OBJECT, INITIATING ROLE and INITIATING FACTOR that belong to the Initiating Mechanism. An example of safety requirement that can be elicited to overcome an object's weakness is **SwSR1** in section G.

E. SARE-ACT2: change, add or remove an object's role

We define the object's role as the task assigned to an object to accomplish a given function. In the PB example, the TCS plays the role of the PB lamp controller, i.e. its task is to control the PB lamp. This task is accomplished by reading some PB status signals in input and producing the output command (to the PB lamp) to turn on, turn off or flash the PB lamp. To be able to act on an object's role, it is necessary to understand how an object's role can be dangerous when accomplishing its task, i.e. why an object does not perform its task as expected. This reasoning may lead to the need to change the current object's role, remove it or add a new role, as shown in Fig. 3. This results in the safety requirements/s to be elicited.

In order to perform this activity, the analyst needs to know: 1) which is/are the object/s whose role can be harmful; 2) for each identified object, which is/are its weakness. It's worth noting that the objects with weaknesses that are considered in this activity are those that belong to the hazard's causes. Moreover, due to the fact that objects' properties and roles are strictly dependent on each other, the knowledge about the object's weakness is necessary to reason about roles. The HO provides the analyst with these pieces of information through the entities ENVIRONMENTAL OBJECT, INITIATING ROLE and INITIATING FACTOR which belong to the Initiating Mechanism.

Trigger
ENVIRONMENT OBJECT=<object>, INITIATING ROLE=<role>, INITIATOR FACTOR=<weakness> from HO
Main flow
1. The analyst answers to the question: "is it possible to remove the weakness <weakness>"?
   - 1.1. [yes] The analyst is asked to describe how the <weakness> can be removed by examining <object>, <role> and <weakness>. Go to step 2.
   - 1.2. [no] The analyst answers to the question: "is it possible to reduce the effects caused by the weakness <weakness>?"
     - 1.2.1. [yes] The analyst is asked to describe how the effects of <weakness> can be reduced by examining <object>, <role>, <weakness>. Go to step 2.
2. The SARE-ACT1 activity ends.

Post-condition
New safety requirements are elicited.

Fig. 2. Steps to overcome an object's weakness (SARE-ACT1)

Once the harmful objects' roles are identified, the analyst can elicit the appropriate safety requirement/s by: 1) adding a new role for the dangerous object; 2) changing the role of the dangerous object; 3) remove the current role for the dangerous object; 4) define a new role for a new or an existing object. An example of safety requirement that can be elicited by acting on the object's role is **SwSR2** in section G. It's worth noting that by overcoming an object's weakness or acting on an object's role, the initiating mechanism (i.e. hazard's causes) is removed. This implies that the hazard cannot occur anymore.

F. SARE-ACT3: cut off existing relations

The relations among objects that are of interest for the safety requirements elicitation are the following:

- The relation between objects with strengths and objects with weaknesses. In fact, this relation describes how an object with weakness becomes a victim of a hazard, and how an object with strength becomes the source of the hazard. In our HO, this relation is called EXPOSURE.
- The relation between the trigger event and the hazard. An event which happens in a specific situation (trigger situation) is the trigger for a hazard to occur. In the PB lamp example, "the maintainer which pushes the PB button to apply the PB" is the event which causes "the train to move", i.e. the hazard. This event happens because the PB lamp shows the PB as released (trigger situation) but the PB is indeed applied. In our HO, this relation is called "bring about" (between the INITIATING EVENT and the HAZARD).

Removing the EXPOSURE relation implies that the hazard's source cannot harm anymore the victim. As a consequence, the hazard cannot occur. On the other hand, cutting off the "bring about" relation means that the trigger event does not end up into a hazard, i.e. the hazard cannot happen. The solution that is proposed to remove the above relations represents the safety requirements to be elicited. The steps and questions for this activity are shown in Fig. 4.

In case of the EXPOSURE relation, the analyst needs to know: 1) which are the objects that can participate to the EXPOSURE relation; 2) for each identified object, which is its weakness; 3) or each identified object, which is its strength. The HO provides the analyst with these pieces of information through the entities EXPOSURE, HAZARD ELEMENT, HARM TRUTHMAKER and MISHAP VICTIM. In particular, the HARM TRUTHMAKER represents the object's strength (the hazard's source) that makes the object become a harmful element with respect to the hazard.

In case of the "bring about" relation, the analyst needs to know: 1) which is the event that ends-up in a hazard; 2) in which situation this event happens; 3) which is the hazard. Once the relations and their components are identified, the analyst can elicit the appropriate safety requirements. Two examples of safety requirements that can be elicited by cutting off relations are **SwSR3** and **SwSR4** in section G.

G. The SARE approach applied to the PB lamp control

In this section, we apply the SARE approach to the PB lamp example. For the sake of simplicity, we focus on the following scenario: SCEN1 "A train is parked in a depot. The PB lamp is turned off, showing that the PB is released. The driver, before leaving the cabin, pushes the PB button to apply the parking brake". In case the PB lamp indication is faulty, this scenario may lead to the collision hazard (C-H). Let's suppose that the PB is applied but the PB lamp is turned off (i.e. PB released). This misleads the driver who, by pushing the PB button to apply the PB, unintentionally releases it.

To understand which mitigation is needed to avoid that a fault in the PB lamp leads to the C-H, it is necessary to understand how it is possible that the PB lamp shows an incorrect PB status. We assume that some of the possible reasons are: 1) faulty contact between the lamp holder and the lamp bulb (a hardware problem); 2) error in the communication of the "turn on" command sent from the TCS to the PB lamp; 3) a PB status signal in input to the TCS is invalid or incorrect; 4) an error occurred in the computation of the PB status (TCS).

Trigger
ENVIRONMENT OBJECT=<object>, INITIATING ROLE=<role>, INITIATOR FACTOR=<weakness> from HO
Main flow
1. The analyst answers to the question: "should the role <role> be changed?"
   1.1. [yes] The analyst is asked to describe how the <role> should be changed by examining <object>, <role> and <weakness>. Go to step 2.
   1.2. [no] The analyst answers to the question: "should the <role> be removed?"
      1.2.1. [yes] The analyst is asked to describe how the role <role> should be removed by examining <object>, <role> and <weakness>. Go to step 2.
      1.2.2. [no] The analyst answers to the question: "should another object and/or a new role for <object> be added?"
         1.2.2.1. [yes] The analyst is asked to describe how to add a new object and/or a new role by examining <object>, <role> and <weakness>. Go to step 2.
2. The SARE-ACT2 activity ends.
Post-condition
New safety requirements are elicited.

Fig. 3 Steps to change, add or remove an object's role (SARE-ACT2)

Trigger
MISHAP VICTIM=<victim>, EXPOSURE=<exposure>, HAZARD ELEMENT=<danger>, HARM TRUTHMAKER=<source>, INITIATING EVENT=<event>, and HAZARD=<hazard> from HO
Main flow
1. The analyst answers to the question: "is it possible to cut off the relation <exposure>?"
   1.1. [yes] The analyst is asked to describe how the <exposure> can be removed by examining <victim>, <danger> and <source>. Go to step 3.
2. The analyst answers to the question: "is it possible to cut off the relation bring about between <event> and <hazard>?
   2.1. [yes] The analyst is asked to describe how to remove this relation by examining <event> and <hazard>. Go to step 3.
3. The SARE-ACT3 activity ends.
Post-condition
New safety requirements are elicited.

Fig. 4 Steps to cut off existing relations (SARE-ACT3)

The HO instance (HO-I1) in Fig. 5 models the scenario SCEN1 in which the reason for the faulty PB lamp indication is "Wrong PB status signal in input to the TCS". By applying the SARE approach to HO-I1 as input, some of the software safety requirements that we can obtain are the following:

- to overcome the TCS's weakness "Incorrect inputs": **SwSReq1** "The TCS shall receive the PB status signal from two different sources and check that both of them report the same value" (from SARE-ACT1, step 1.2.1)
- to add a new role for the TCS: **SwSReq2** "In case of failures of the primary TCS, the PB status indication shall be computed by the redundant TCS" (from SARE-ACT2, step 1.2.2.1) OBS! This requirement is appropriate to overcome the weakness "Computation faults" of the PB lamp controller.
- to cut-off the EXPOSURE relation: **SwSReq3** "The TCS shall monitor the train unintended movements and apply the emergency brake when the train starts moving" (from SARE-ACT3, step 1.1).
- to cut-off the "bring about" relation: **SwSReq4** "The TCS shall trigger an acoustic alarm if the PB is released when the train is in Parking Mode" (from SARE-ACT3, step 2.1).

IV. RELATED WORK

A. Safety requirements elicitation based on safety analysis

Safety requirements strictly depend on the hazards they mitigate. For this reason, a lot of efforts have been devoted to exploit the hazard knowledge collected during the safety analysis to derive safety requirements.

In particular, different publications address the problem of using the result of the Fault Tree Analysis (FTA), one of the main safety analysis techniques, to derive functional safety requirements. Hansen, Ravn and Stavridou [11] propose a method to create safety software requirements based on the failures description provided by the FTA. For each fault tree identified by the FTA, the software safety requirements describe what the system should not do in order to prevent a failure from happening. Martins and Oliveira [12] present a protocol to help requirements engineers to derive functional safety requirements from FTA. The idea behind this proposal is that FTA provides a means

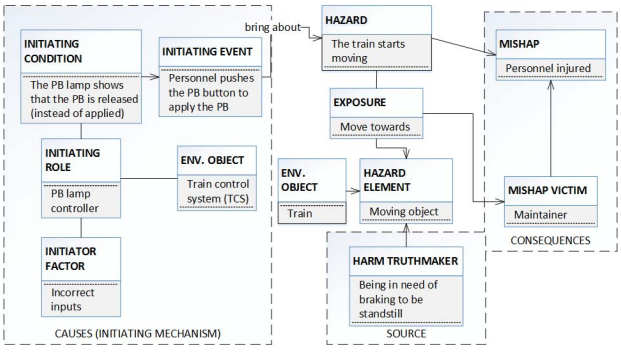

Fig. 5 An instance of the Hazard Ontology (HO-I1)

of analyzing the risky events in order to understand under which environmental conditions a system becomes unsafe. This knowledge is then used to write functional safety requirements that define how the system should or should not behave to maintain it in a safe state. Gorski and Wardzinski [13] extend the FTA results by adding time properties to the relationships among events in order to derive software safety requirements suitable for real-time systems. Vyas and Mittal [14] apply the Software FTA technique to requirements specified as use cases to elicit additional safety requirements. In their work, hazards are expressed as incompatible states of the system components, and FTA are built for each use case after having identified the action steps that can contribute to a hazard. Safety requirements are then obtained from the minimal cut sets for the fault trees.

There are also publications concerning safety requirements elicitation and specification based on other safety analysis techniques, such as Hazard and Operability Study (HAZOP) and Failure Mode and Effects Analysis (FMEA). Allenby and Kelly [15] use a subset of the HAZOP guided-words along with functional requirements expressed as scenario in order to derive safety requirements that capture the hazardous deviations from the core intent defined in the scenarios. Goddard [16] combines Petri-nets and FMEA to identify incomplete, inconsistent and incorrect requirements with respect to their safety implications.

All these works agree upon the fact that safety analysis techniques "*provide the mechanism for identification of the safety related requirements*" [15] since the knowledge about how hazards may happen is essential to establish the countermeasures expressed in the safety requirements for preventing or mitigating the effects of failures. However, these works also show that the information provided by safety analysis, which are techniques originated as hazard identification and hazard analysis rather than requirements elicitation methods, is not complete to elicit the safety requirements. The interesting and fundamental information for safety requirements elicitation that can be extracted from safety analysis primarily concerns the identification of the different failure cases. However, other methods, which vary from interviews to domain experts to system functionalities described as scenarios, have been employed to complement and complete the failure knowledge in order to elicit the safety requirements. Tiadjo and Jamboti [17] study, in particular, the applicability of the most common safety analysis techniques for the elicitation of safety requirements for Ambient Assisted Living (AAL) systems. Their study shows that the most of these techniques cannot support requirements specific to these systems, such as context-awareness. Another problem, discussed in [14] and [15], is the fact that failures identified during safety analysis are not directly linked to the systems functional behavior, i.e. it is difficult to deduce how a function could contribute to a failure.

Our work shares, with the above cited works, the assumption that safety requirements elicitation shall be based on the hazard knowledge that the safety analysis techniques produce. However, our approach differs mainly for the following reasons: 1) the approach is independent of any particular safety analysis technique. It only relies on the knowledge about hazards which is obtained by applying any of the available safety analysis techniques; 2) the approach is based on the assumption that the information needed to elicit the safety requirements shall be searched among the three hazard's components (refer to section

II); 3) the approach focuses on the failures which lead to a hazard as a guidance to elicit safety requirements.

B. Requirements elicitation based on ontologies

Ontologies have been widely used in requirements engineering [23] and, specifically, for requirements elicitation. Ontologies aim at modeling a domain of knowledge or discourse. Ontologies are therefore used to depict the knowledge about an application domain to guide the analyst for eliciting the appropriate requirements [24][25][26][27] and more.

Our approach also uses an ontology to support requirements elicitation. However, our approach differs from other works in this area because: 1) the ontology is not a domain ontology about a real-world application but a hazard ontology, i.e. it captures the information about the hazard's components and their relations to provide the conceptualization of the hazard (refer to section II); 2) the HO is used to elicit safety requirements. In our previous works [1][2][3], we proposed the HO and structured methods to populate it with information about hazards. In this paper, we extend our previous work by presenting a method to elicit safety requirements based on the HO.

V. Conclusion and Future Work

In this paper, we described an ontological approach for safety requirements elicitation based on a Hazard Ontology. The approach provides the analyst with a guidance to elicit the safety requirements driven by the knowledge of how hazards occur. The approach is based on the assumption that the information needed to discover safety requirements must be searched among the hazards' components and their relations.

We are currently evaluating this method in an industrial setting to elicit safety requirements for a construction plant consisting of a mix of both autonomous and manually operated vehicles.

Acknowledgment

This research is supported by the Knowledge Foundation (KK-stiftelsen) project DPAC – Dependable Platforms for Autonomous Systems and Control.

References

[1] J. Zhou, K. Hänninen, K. Lundqvist, and L. Provenzano, "An Ontological Approach to Identify the Causes of Hazards for Safety-Critical Systems," ICSRS'17, in press.

[2] J. Zhou, K. Hänninen, K. Lundqvist, and L. Provenzano, "An ontological approach to hazard identification for safety-critical systems", in Proc. ICRSE'17, July, 2017.

[3] J. Zhou, K. Hänninen, K. Lundqvist, and L. Provenzano, "An ontological interpretation of the hazard concept for safety-critical systems", ESREL'17, June, 2017.

[4] R. Winther, and W. Marsh, "Hazards, accidents and events – A land of confusing terms", presented at the ESREL, Amsterdam, NL, Sept. 2013.

[5] A. Lawrynowicz, and I. Lawniczak, "The hazardous situation ontology design pattern", in WOP:2015, October 2015, pp. 1-4.

[6] J. Hatcliff, A. Wassyng, T. Kelly, C. Comer and P. Jones, "Certifiably safe software-dependent systems: challenges and directions", in Proc. FOSE 2014, 2014, pp. 182-200.

[7] Nancy G. Leveson, Engineering a safer world. System thinking applied to safety, Ed. Cambridge, MA: MIT Press, 2011, ch. 9, sec. 9.3, pp. 263-265.

[8] D. Firesmith, "A taxonomy of safety-related requirements", in RHAS (workshop), 2004.

[9] A. Parisaca-Vargas, and R. Bloomfield, "Using ontologies to support model-based exploration of the dependencies between causes and consequences of hazards", in Proc. Int. KEOD, 2015.

[10] C. A. Ericson, "Hazard analysis techniques for system safety", 2nd ed., Ed. New York: Wiley, 2005, pp. 28-44.

[11] K. M. Hansen, A. P. Ravn, and V. Stavridou, "From safety analysis to software requirement", IEEE Trans. Softw. Eng., vol. 24, pp. 573-584, July, 1998.

[12] L. E. Galvao Martins, and T. de Oliveira, "A case study using a protocol to derive safety functional requirements from fault tree analysis", in Proc. Int. RE'14, Aug. 2014, pp. 412-419.

[13] J. Gorski, and A. Wardzinski, "Deriving real-time requirements for software from safety analysis", in Proc. EURWRTS'96, 1996, pp. 9-14.

[14] P. Vyas, and R. K. Mittal, "Eliciting additional safety requirements from use cases using SFTA", in Proc. RAIT, March 2012, pp. 163-169.

[15] K. Allenby, and T. Kelly, "Deriving safety requirements using scenarios", in Proc. Symp. Requirement Engineering, 2001, pp. 228-235.

[16] P. L. Goddard, "A combined analysis approach to assessing requirements for safety critical real-time control systems", in Proc. Symp. Reliability and Maintainability, 1996, pp. 110-115.

[17] A. M. Tiadjio, and K. Jamboti, "Requirements and evaluation of safety analysis techniques for ambient assisted living systems", in Proc. Symp. Software Reliability Engineering Workshops, Nov. 2012, pp. 319-324.

[18] R. Riehle, "Failure-driven software safety", ACM SIGSOFT SEN, vol. 32, nr. 5, pp. 1-4, Sept., 2007.

[19] O. Dieste, N. Juristo, and F. Shull, "Understanding the customer: what do we know about requirements elicitation?", IEEE Softw., vol. 25, nr. 2, pages 11-13, 2008.

[20] H. Bani-Salameh, and N. Al Jawabreh, "Towards a comprehensive survey of the requirements elicitation process improvements", in Proc. IPAC, 2015.

[21] G. Kotonya, and I. Sommerville, Requirements Engineering, Processes and Techniques, Ed. New York: Wiley, 2000.

[22] C. H. N. Lahoz, J. B. Camargo, M. A. D. Abdala, and L.A. Burgareli, "A software safety requirements elicitation study on critical computer systems", in Proc. IET Int. System Safety, London, UK, 2006, pp. 47-53.

[23] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva, "Applications of ontologies in requirements engineering: a systematic review of the literature", Requirements Engineering, vol. 21, pp. 405-437, February, 2015.

[24] I. Omoronyia, G. Sindre, T. Stålhane, S. Biffl, T. Moser, and W. Sunindyo, "A domain ontology building process for guiding requirements elicitation", in Proc. REFSQ'10, June 2010, pp. 188-202.

[25] S. Farfeleder, T. Moser, A. Krall, T. Stålhane, I. Omoronyia, and H. Zojer, "Ontology-driven guidance for requirements elicitation", The semantic web: research and applications, Ed. Antoniu G, Grobelnik M, Simperl E, Parsia B, Plexousakis D, De Leenheer P, Pan J (eds), vol. 6644, Springer, Berlin, Heidelberg, pp. 212-226, 2005.

[26] M. Kitamura, R. Hasegawa, H. Kaiya, and M. Saeki, "A supporting tool for requirements elicitation using a domain ontology", Software and data technology, Ed. Filipe J, Shishkov B, Helfert M, Maciaszek L (eds), pp. 128-140, 2009.

[27] D. V. Dzung, and A. Ohnishi, "Ontology-based reasoning in requirements elicitation", in Proc. Int. Software Engineering and Formal Methods, Nov. 2009, pp. 263-272.