Recommendation-Based Decision Support for Hazard Analysis and Risk Assessment

Kerstin Hartig and Thomas Karbe

Institute for Software Engineering - TU Berlin
10587 Berlin, Germany
Email: {kerstin.hartig, thomas.karbe}@tu-berlin.de

***Abstract*—Since 2011, automotive companies have to adhere to the functional safety standard ISO 26262. One important safety activity described in the standard is the hazard analysis and risk assessment, which is strongly expert-driven, and therefore expensive, time consuming, and dependent from the individual expert's opinion. In this paper, we present a decision support system for hazard analyses in order to increase their consistency and efficiency. The system automatically combines results from finished analyses and supporting information in a knowledge base and searches it for useful recommendations during a new hazard analysis and risk assessment.**

Keywords–Decision Support; Advisory System; Recommendation System; Spreading Activation; Hazard Analysis.

I. INTRODUCTION

In 2011, the automotive functional safety standard ISO 26262 "Road Vehicles - Functional Safety" [1] was published. Since then, the individual safety processes of automotive companies were adapted and now each new system for a car is developed according to the ISO 26262. The *hazard analysis and risk assessment* (HARA) is one of the first activities of the safety lifecycle. In this analysis, experts examine the systems with respect to its functions, possible malfunctions, and the consequences of those malfunctions in different situations. For many systems in the automotive domain nearly identical systems exist for other series vehicles. However, a simple copy-paste approach is not feasible. Even small changes in a system could lead to completely different analysis results.

Since the ISO 26262 is still a very young standard, there are not many tools to support it appropriately. According to [2], the experience of experts is still the main means to conduct a proper HARA. In order to reduce the workload of the domain experts and to increase the consistency of HARA projects for similar systems, we propose a recommendation system that bases its recommendations on already completed analyses, and that therefore makes optimal use of the reuse potential. The system automatically creates a knowledge base that combines information from other HARA projects with complementary information, e.g., synonym dictionaries. When an expert is working on a new HARA, the system proposes knowledge artefacts that could be useful for the actual or next analysis step, together with an explanation. Relevance in the knowledge base is determined by a mechanism called *spreading activation* that leverages the relationships between concepts in a semantic network. In Section II of this paper, we cover the basics and the related work for the topics HARA, spreading activation, and semantic web technologies. In Section III, we discuss the two phases of our proposed recommendation system. Finally, in Section IV, we summarize our results and present multiple possibilities to continue research in this area.

II. BASICS AND RELATED WORK

In this section, we shortly introduce the main concepts and tasks for conducting a HARA. Furthermore, we describe spreading activation and its application as semantic search technique. In a third part, we present selected applications of semantic web technologies that have been applied in non-web environments and are related to our approach.

A. Hazard Analysis and Risk Assessment (HARA)

According to ISO 26262, HARA is a method for identifying and assessing hazards and specifying safety goals in order to reduce risks down to an acceptable level [1]. The HARA workflow consists of several steps, which can be tailored individually.

The initial input is a collection of documents related to an item of interest, e.g., description, interfaces, architecture. In subsequent steps, the item functions to be examined are defined, their potential malfunctions are identified, relevant driving situations are assigned, and hazardous situations are derived. The impact and consequences of each hazardous situation are determined and their risk is classified by the specific parameters. Their evaluation leads to the assignment of an Automotive Safety Integrity Level (ASIL) and results in appropriate safety goals. Higher ASILs usually require higher efforts in providing functional safety. HARA strongly relies on expert knowledge, usually involving several experts from different departments and is usually a very complex and time-consuming analysis.

B. Spreading Activation

Spreading activation has its origin in the fields of psychology and psycholinguistics. It was used as a theoretical model to explain semantic memory search and semantic preparation or priming [3]–[5]. A semantic network was defined as an explanatory model of human knowledge representation. In such a network, concepts are represented by nodes and the associations between concepts as links [4]. Over the years, spreading activation evolved into a highly configurable semantic search algorithm and found its application in different fields [6]. Spreading Activation is capable of both identifying and ranking the relevant environment in a semantic network.

The processing of spreading activation is usually defined as a sequence of one or more iterations, so-called pulses. Each node in a network has an activation value that describes its current relevance in the search. In each pulse, activated nodes spread their activation over the network towards associated concepts, and thus mark semantically related nodes [6]. If a termination condition is met, the algorithm will stop. Each pulse consists of different phases in which the activation values are computed by individually configured activation functions.

Additional constraints control the activation process. Fan-out constraints limit the spreading of highly connected nodes because a broad semantic meaning may weaken the results. Distance constraints reduce activation of distant nodes, because distant nodes are considered to be less associated to each other. There are many other configuration details such as decays, thresholds, and spreading directions. In the survey, Crestani argues that spreading activation is capable of providing good results, but the effectiveness highly depends on the availability of a representative network as well as techniques for automated network building [6]. Therefore, the approach presented in this paper aims at both the automated creation and the semantic enrichment of the network.

C. Applications of Semantic Web Technologies

In 2001, Tim Berners Lee coined the term *Semantic Web* [7], which envisions extensive sharing and reuse of semantically enriched data over the web. To support this vision, organizations and initiatives such as the W3C elaborate on development and standardization of knowledge and semantic technologies, including RDF and OWL. While those technologies are created with the web in mind, they are useful in other domains as well.

One area of application is the *semantic desktop*, which aims at transferring semantic web technologies to the user's desktop [8]. Schumacher et al. even apply spreading activation in semantic desktop information retrieval [9]. Semantic desktop technologies primarily focus on interconnecting different desktop applications for personal or group information management, e.g., implemented in the NEPOMUK Project [10]. Similarly, we want to combine semantic web technologies and spreading activation, but focus on providing recommendations for safety analyses such as HARA. Álvarez et al. examined spreading activation techniques for information retrieval in RDF graphs and ontologies [11]. They introduced the OntoSpread Framework to support configuration and execution of the algorithms and applied it in a medical recommendation system [12]. However, they utilized existing ontolgies whereas our approach includes the overall process of creating and searching semantic networks in order to provide step-by-step guidance through the analysis process by problem-specific recommendations.

III. APPROACH FOR A RECOMMENDATION SYSTEM FOR DECISION SUPPORT

A. Approach

We propose an approach to enhance a HARA tool with semantic technologies in order to provide the user with recommendations. One such analysis tool is medini analyze [13], in which the HARA projects used in this paper were conducted. However, our approach is independent from a concrete tool and applicable to any tool with a known structure, e.g., meta model, class diagram. The approach consists of two phases: the building phase and the search phase, each of which comprises three steps (see Figure 1). The building phase includes building the knowledge base on model and instance level and a post-processing step for semantic enrichment. The search phase includes the identification and evaluation of relevance, generation of recommendations and providing explanations.

Throughout the remainder of this paper, we will make use of the following concrete scenario when explaining each step.

Example: A safety engineer adds a new function, namely "operate directional indicator", during a HARA. The engineer queries the system for functions in order to see, which related functions have been used in earlier HARA projects. Next to finished HARA projects, the system contains knowledge about synonyms. One entry in this synonym collection is, that "directional indicator" and "turn signal" have the same meaning. Therefore, one of the provided recommendations should be the function "activate turn signal right", which has been used in a finished HARA project.

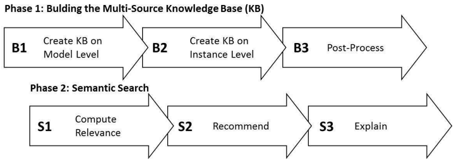

Figure 1. Phases and Steps of the Approach.

B. Building Phase: Multi-Source Knowledge Base

Optimally, recommendation detection should be conducted on a knowledge base containing extensive expert's knowledge. This knowledge originates from different sources, most importantly from already completed analyses. Additional information, such as glossaries, synonyms, feature models, or other domain-specific background knowledge can help to find potentially useful semantic relationships between different artefacts. Therefore, our proposed knowledge base has an extensible modular structure, consisting of multiple so-called *knowledge blocks*. Creating this knowledge base automatically bypasses the main obstacles for successful application of spreading activation, i.e., dependance on the representativeness of networks and automated network building [6].

Each block consists of both the model representation of the knowledge and their instances. Therefore, we require both the XML schema definition and the data provided in XML as input. A block contains relations between concepts within the block, as well as relations to other blocks, stitching multiple blocks to one piece. These so-called cross-block relations are identified and set whenever a new block is included.

1) Automatic Generation of the OWL Model (B1): The main knowledge block for a tool-based recommendation system is given by the data structure of the tool itself, usually available through meta models or class diagrams. In this paper, the target language for the semantic representation is Web Ontology Language (OWL), a W3C standardized description language with formal semantics for representing and computing knowledge. However, the approach is applicable to any other target structure based on RDF Graph. In OWL, we can describe information as classes, properties, instances, and data values [14]. Given XML schema definitions of a meta model and other information sources, we can apply mapping techniques to create an OWL model. In [15], Bohring and Sauer propose an XSD to OWL mapping to capture the XML schema semantics while translating the schema constructs to OWL. Similar transformation approaches are described in several other publications, e.g., [16][17]. We slightly adapt the existing mappings for our specific transformation.

Example: In our example, we provide, additionally to the tool meta model, a collection of synonyms as second knowledge block. Synonyms are easy enough to explain in the example, but carry semantic meaning, and therefore have a visible impact. Synonyms are represented by a class with a name attribute and a reflexive *synonym* association. In the same beforementioned fashion, we apply our transformation. This results in an *owl:class Synonym* and a symmetric object property *hasSynonym* as well as a datatype property for the synonym name (see upper right side of Figure 2).

2) Automatic Import of OWL Instances (B2): Now, we want to fill the created OWL model with instance data. The import can be technically implemented using an XML to OWL transformation [15].

Example: For the scenario, we import the instances "turn signal" and "directional indicator" of type *Synonym* and connect them by a *hasSynonym* link. Furthermore, we include the instance "activate turn signal right" of type *Function*, among others (see Figure 2).

3) Stitching Multi-Source Knowledge Blocks (B3): Knowledge blocks need to be interconnected in order to capture known semantics. Proper stitching is essential, since it represents the actual semantic enhancement of the knowledge base. Usually, stitching knowledge blocks requires domain knowledge to decide which concrete concepts need to be connected. However, once this decision is made, the linking process can be automated via stitching rules. The resulting OWL representation including the model and instance level consists of an underlying RDF graph which is composed of a set of RDF triples [18]. Each triple consists of a subject, a predicate, and an object which read as a statement, e.g., "Function *hasMalFunction* Malfunction". The OWL to RDF graph mapping is standardized by the W3C [19].

Example: We stitch the HARA block and the synonym block by introducing a new relation *hasSynonymConnection*. This relation links all instance nodes that contain a synonym instance name with that synonym instance (see Figure 2).

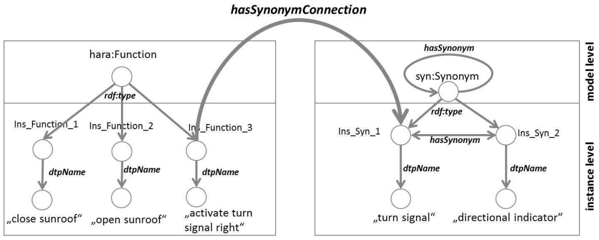

Figure 2. Knowledge Base with Knowledge Blocks for HARA and Synonyms.

C. Search Phase: Semantic Search Concept

Searching the knowledge base is conducted in three steps (see Figure 1). Firstly, we apply a spreading activation algorithm to identify the context of our specific search, i.e., the relevant subnetwork. This step reduces the search space and ranks the visited nodes by their relevance. Secondly, we filter the most relevant nodes in the resulting subnetwork by the sought-after type. As a result, we generate recommendations for the user in order to support their decisions. In a third step, we provide explanations for the recommendations.

1) Spreading Activation (S1): Since spreading activation algorithms are highly configurable and profit from domain- and problem-specific configurations, we apply the following configuration settings: The termination criteria are a specified amount of pulses, the full activation of the graph, as well as a threshold for the total activation value transmission of a pulse. In case of convergence the spreading will stop. We additionally apply fan-out and local distance constraints to limit the activation broadcast of highly connected nodes and decrease the activation depending on the path distance. We apply a pulse constraint to reduce the spreadable activation values over the time in order to achieve convergence with increasing pulse count. Most importantly, we apply path constraints utilizing the semantic relevance of properties.

Example: In our scenario, we privilege the synonym knowledge block because the knowledge of two words meaning the same thing can boost the search. In order to emphasize their importance, we attach higher weights to the associated properties *hasSynonymConnection* and *hasSynonym*. Figure 3 depicts our search scenario. The engineer added the function "operate directional indicator" and now searches for associated functions.

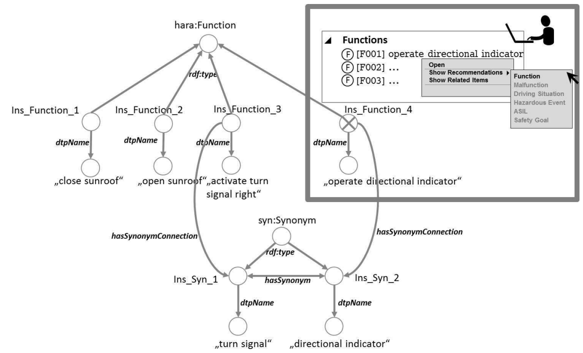

Figure 3. Recommendation Query.

Figure 4 depicts the semantic network before (a) and during five pulses (b-f) of the spreading in our network. Starting point is the crossed node, which stands for the newly added function. Since synonym property edges receive higher weights, they are represented by darker color. Activation spreads in pulses over the network whereas higher activation of nodes is represented by darker color. Over the pulses, the faster activation over priorized edges and limitations by fan-out constraints at nodes with lots of branches can be observed. The result is a semantic network with nodes ranked by relevance.

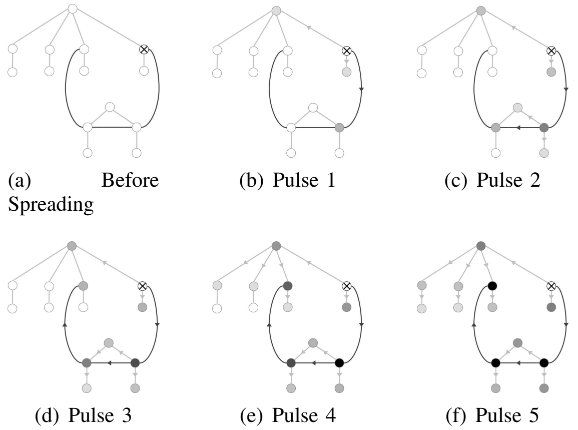
(a) Before Spreading (b) Pulse 1 (c) Pulse 2
(d) Pulse 3 (e) Pulse 4 (f) Pulse 5

Figure 4. Semantic Network before and during spreading activation pulses.

2) Recommendations through Type-Specific Filtering (S2): Recommendation requests are specific to a concrete artefact type. Therefore, we filter the relevant subnetwork resulting from the spreading step by the sought-after type sorted by their assigned activation value representing their relevance regarding the specific query.

Example: The filtered subnetwork, depicted in Figure 5, only contains instances of the artefact type *Function*. The node that represents the function “activate turn signal right” has the highest relevance, and therefore is the first recommendation generated for our scenario.

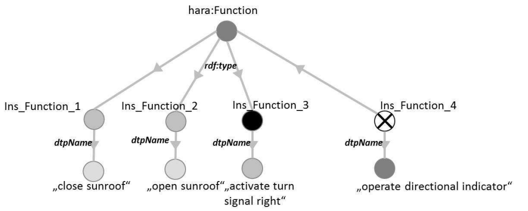

Figure 5. Filtering by Types for Identification of Recommendation.

3) Explanations (S3): For user acceptance, the origins of the resulting recommendations must be transparent. Thus, decision support for HARA can profit from appropriate explanations. Explanation can be derived by evaluating the activation history and find the path sequence that contributed most to the activation of a specific node. Optimizing the explanation given for each recommendation is work in progress and will be examined in our future research work.

Example: In the presented example, the explanation is obvious: The function “activate turn signal” is the highest ranked recommendation, because “turn signal” and “directional indicator” are synonyms, and therefore have the same meaning. In our case, the shortest and highest activated path determines this explanation (see Figure 4(f)).

D. Implementation

The proposed recommendation system is implemented in a prototype called *HARvESTer (Hazard Analysis and Risk assessment dEcision Support Tool)*. We examined different scenarios, generating recommendations for functions, malfunctions and safety goals. First experiments in a safety expert environment led to positive feedback regarding usefulness and showed promising results. Expected recommendations have been found in most cases.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented a decision support system for hazard analysis and risk assessment which aims at increasing efficiency and more consistent and reliable results. The system has two main capabilities: automated construction of a knowledge base from different information sources and finding related information for deriving recommendations during the HARA steps. Since these recommendations are based on already finished analyses, the experts have fast access to decisions that have been made before and can decide to reuse them. Although our first results are very promising, we see much potential for future research.

Our method focuses on HARAs, but could be easily adapted to other analyses of ISO 26262, or even outside of the safety domain. A challenging idea is the automatic configuration of the spreading algorithm to improve results. User feedback could be a useful addition for the recommendation system such that it could learn which recommendations were actually useful, and which were not. Furthermore, an extensive case study is planned to evaluate the overall approach and its usability as well as the effects of different configurations.

REFERENCES

[1] ISO 26262 - Road vehicles - Functional safety, International Organization for Standardization, Nov. 2011.

[2] C. Maier, A. Schloske, and S. Bothe, “Studie zur Funktionalen Sicherheit in der Automobilbranche [Survey of Functional Safety in the Automotive Domain (ISO 26262)],” Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), Tech. Rep., Mar. 2013.

[3] M. R. Quillian, “Semantic Memory,” in Semantic Information Processing, M. Minsky, Ed. MIT Press, 1968, pp. 216–270.

[4] A. M. Collins and E. F. Loftus, “A spreading activation theory of semantic processing,” Psychological Review, vol. 82, no. 6, Nov. 1975, pp. 407–428.

[5] J. R. Anderson, “A Spreading Activation Theory of Memory,” Journal of Verbal Learning and Verbal Behavior, vol. 22, 1983, pp. 261–295.

[6] F. Crestani, “Application of Spreading Activation Techniques in Information Retrieval,” Artificial Intelligence Review, vol. 11, 1997, pp. 453–482.

[7] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Scientific American, vol. 284, no. 5, May 2001, pp. 34–43.

[8] L. Sauermann, A. Bernardi, and A. Dengel, “Overview and Outlook on the Semantic Desktop,” in Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference, ser. CEUR Workshop Proceedings, S. Decker, J. Park, D. Quan, and L. Sauermann, Eds., vol. 175. CEUR-WS, Nov. 2005.

[9] K. Schumacher, M. Sintek, and L. Sauermann, “Combining Fact and Document Retrieval with Spreading Activation for Semantic Desktop Search,” in ESWC, ser. Lecture Notes in Computer Science, S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, Eds., vol. 5021. Springer, 2008, pp. 569–583.

[10] T. Groza, S. Handschuh, K. Moeller, G. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, and R. Gudjonsdottir, “The NEPOMUK Project – On the way to the Social Semantic Desktop,” in Proceedings of the Third International Conference on Semantic Technologies (I-SEMANTICS 2007), Graz, Austria, 2007, pp. 201–211.

[11] J. M. Álvarez, D. Berrueta, L. Polo, and J. E. Labra, “ONTOSPREAD: A Framework for Supporting the Activation of Concepts in Graph-Based Structures through the Spreading Activation Technique,” in Information Systems, E-learning, and Knowledge Management Research, ser. Communications in Computer and Information Science, vol. 278. Springer Berlin Heidelberg, 2013, pp. 454–459.

[12] J. M. Álvarez, L. Polo, W. Jimenez, P. Abella, and J. E. Labra, “Application of the spreading activation technique for recommending concepts of well-known ontologies in medical systems,” in Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine - BCB ’11, 2011, pp. 626–635.

[13] “KPIT - medini analyze - Functional Safety Tool,” 2016, URL: http://www.kpit.com/engineering/products/medini-functional-safety-tool [accessed: 2016-03-11].

[14] “OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition),” W3C, W3C Recommendation, Dec. 2012.

[15] H. Bohring and S. Auer, “Mapping XML to OWL Ontologies,” in Computer Science Days Leipzig, ser. LNI, K. P. Jantke, K.-P. Fähnrich, and W. S. Wittig, Eds., vol. 72. GI, 2005, pp. 147–156.

[16] I. Bedini, C. Matheus, P. F. Patel-Schneider, A. Boran, and B. Nguyen, “Transforming XML Schema to OWL Using Patterns,” in Proceedings of the 5th IEEE International Conference on Semantic Computing, ICSC 2011. IEEE Computer Society, 2011, pp. 102–109.

[17] N. Yahia, S. A. Mokhtar, and A. Ahmed, “Automatic Generation of OWL Ontology from XML Data Source,” International Journal of Computer Science Issues, vol. 9, Mar. 2012, pp. 77–83.

[18] “RDF 1.1 Concepts and Abstract Syntax,” W3C, W3C Recommendation, Feb. 2014.

[19] “OWL 2 Web Ontology Language. Mapping to RDF Graphs (Second Edition),” W3C, W3C Recommendation, Dec. 2012.