Automated Traceability between Requirements and Model-Based Design

Maria Bonner[1], Marc Zeller[2], Gabor Schulz[1], Dagmar Beyer[2] and Mihaela Olteanu[3]

[1]Siemens AG, Gleiwitzer Str. 555, 90475 Nürnberg, Germany
[2]Siemens AG, Otto-Hahn-Ring 6, 81739 München, Germany
[3]Siemens S.R.L., Bulevardul 15 Noiembrie 78, 500097 Brasov, Romania

Abstract

Traceability is an important aspect in system development, since it helps to ensure that the developed system fulfills all requirements and prevents failures. When developing safety-critical systems, traceability is mandatory to demonstrate that the system is implemented correctly. Unfortunately, establishment and maintenance of trace links are hard to achieve manually in today's complex systems. Tools and methods for automated trace link generation, which have been proposed so far, have no broad adoption in industry and there is lack of evidence of their effectiveness. To address these challenges we introduce a tool, which allows establishing bi-directional traceability links between requirements and model-based designs using Artificial Intelligence (AI). Our tool is an extension of the Siemens toolchain for Application Lifecycle Management (ALM), systems engineering, and embedded software design. It creates trace links between requirements written in natural language and Capital™ software, AUTOSAR, SysML, UML, or Arcadia models for system/software design. This paper describes the implemented use-cases of tracing system/SW/HW requirements to system architecture models and provides an overview of the tool architecture.

Keywords

Requirement traceability, Model-Based Design, NLP, Ontology, Semantic Web

1. Introduction

Modern Electrical/Electronical (E/E) systems are characterized by a huge growth in complexity. One of the big challenges is to assure that all requirements are implemented correctly in software, electrical, electronic, and network designs. Especially, when developing safety-critical system, safety standards, e.g., ISO 26262 [1] in automotive, ISO 17894 [2] in marine, or ARP 4754/4761 [3] in avionics, demand a proof that the system

In: A. Ferrari, B. Penzenstadler, I. Hadar, S. Oyedeji, S. Abualhaija, A. Vogelsang, G. Deshpande, A. Rachmann, J. Gulden, A. Wohlgemuth, A. Hess, S. Fricker, R. Guizzardi, J. Horkoff, A. Perini, A. Susi, O. Karras, A. Moreira, F. Dalpiaz, P. Spoletini, D. Amyot. Joint Proceedings of REFSQ-2023 Workshops, Doctoral Symposium, Posters & Tools Track, and Journal Early Feedback Track. Co-located with REFSQ 2023. Barcelona, Catalunya, Spain, April 17, 2023.

maria.bonner@siemens.com (M. Bonner); marc.zeller@siemens.com (M. Zeller); gabor.schulz@siemens.com (G. Schulz); dagmar.beyer@siemens.com (D. Beyer); mihaela.olteanu@siemens.com (M. Olteanu)

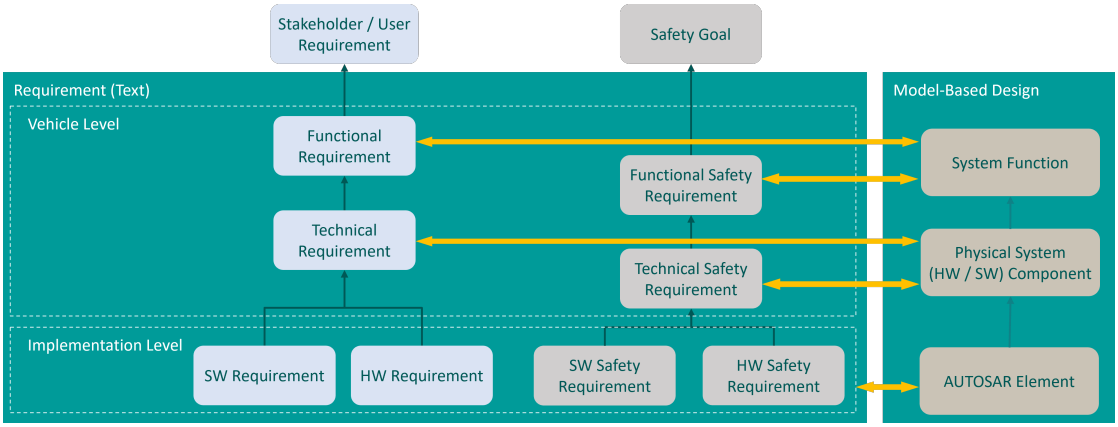

Figure 1: Traceability information model showing traced artifacts.

is built correctly. Hence, it is required to show vertical traceability between (safety) requirements, the system/software design, and the test cases.

Following the ISO 26262 guideline as well as established standards for the engineering of automotive E/E systems such as ASPICE [4], functional/safety system requirements are typically derived from either stakeholder/user requirements or safety goals and provide the input to create a functional system design. In the next step, the functional system architecture is defined in a model-based way, e.g., as an Internal Block Diagram in SysML [5], as a function chain in Capella [6], or as a functional architecture in Capital™ Systems Architect [7]. Afterwards, the functional requirements are refined in a set of technical (safety) requirements which are the basis to create a technical/physical system architecture. The physical system architecture describes the technical components (HW & SW) which realize the functional architecture. In a next design step, the technical (safety) requirements are refined into a set of SW (safety) requirements and a set of HW (safety) requirements which are the basis for the implementation of the E/E system by a set of interconnected electronic control units (ECUs) which run a set of SW-based functions.

In the automotive domain, the SW of an ECU is realized using the AUTOSAR standard[8]. Hence, the HW & SW (safety) requirements can be linked to AUTOSAR elements which realize these requirements. However, creating the necessary trace links is cumbersome and in case of changes of either the requirements or the design, the links must be adjusted accordingly. Since engineers fear to create wrong trace links which may lead to confusion during the entire development, often no trace links are created at all.

We propose a tool for semi-automating the task of creating and maintaining trace links between requirements and model-based design (see Figure 1). The tool is aligned with the Siemens development flow for the design of E/E systems [7]. For our tool we use a semantic web layer, which extends the definition of Siemens ontology libraries [9] for Siemens tools with a traceability model between the different tools.

In this tool demo, we present a use-case, in which safety requirements are defined and

stored in the Polarion^{TM} tool and the engineer is conducting the model-based design of the E/E system in the Capital^{TM} software. The task of the engineer is to establish trace links between safety requirements to the model-based designs to illustrate that the safety features are implemented correctly. The proposed trace link recommender tool suggests either trace links to a model element in the model-based design, or trace links to a safety requirement in Polarion^{TM}. In addition to the use-case, we provide an abstract architecture of the tool with its underlying technologies.

The rest of the paper is organized as follows: In Section 2, we provide a brief overview of the foundations used for our tool. Section 3 describes the abstract architecture of our tool. Afterwards, we present different use cases in which our tool automates the trace link creation. At the end, we summarize the main results of the paper and provide an outlook on future research work in Section 5.

2. Foundations

There are a number of approaches in the area of information retrieval which allow to extract information from different sources. In ontology-based approaches, requirements are extracted into a knowledge graph conformed to a predefined ontology. This extraction happens with a high level of granularity, additional reasoners allow to identify broader semantic meaning [10]. Finally, the rules over ontologies can be defined to automate the trace link generation or the availability of training data to adapt models for domain specific entity recognition [11]. As opposed to granular approaches, which assume easy recognizable structure, there are vector space based approaches [12], which assume text conversion into a vector space model, and application of a distance measure to calculate the semantic similarity between artifacts. Despite fundamental problems with these techniques, such as recognition of different words in the same contexts (synonymy) or identification of the same word for multiple purposes (polysemy), vector space methods show high applicability in practice.

Recent approaches for automated trace link generation are using machine learning techniques [13, 14, 15, 16]. These techniques assume existence of predefined traces, which are used as training data. A machine learning model learns from these data by noticing the connection between artifacts and then proposes new traces. Machine learning algorithms have big potential, since they can discover features which are not always obvious to humans. The drawback is that machine learning techniques require large, labeled datasets, which can hardly be shared outside of organizations. Additionally, there is no guarantee that an algorithm trained on one dataset will perform well on another dataset with the same structure.

Our goal is to propose a tool, which (1) allows to aggregate information from different sources in a semantic data layer, and (2) allows to generate trace links based on vector-based methods between the retrieved data.

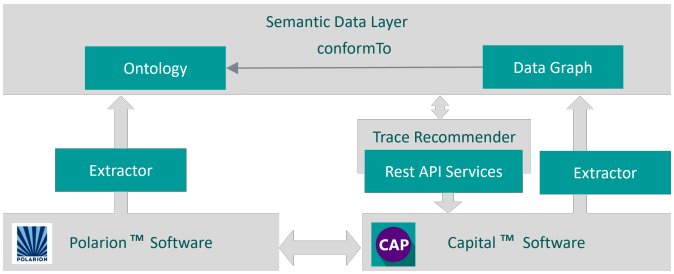

Figure 2: Traceability recommender tool - architecture.

3. Architecture

The traceability recommender tool is an extension of existing Siemens tools and adds an additional semantic data layer for data aggregation and communication of heterogeneous sources (see Figure 2). This layer allows customizing traced artifacts for better data organization by the definition of ontologies. Siemens is actively working on the Siemens ontology database [9], which allows to unify product lifecycle management systems for the creation of a digital twin over several domains.

The traceability recommender tool covers different modelling domains. The E/E architecture is defined either in the tool Capital^{TM} Systems Architect or using SysML or the Arcadia methodology. While the software system is defined using UML or the AUTOSAR standard for embedded software in the automotive domain. For each modelling domain there is a separate ontology. Traceability information are contained in cross-domain ontologies. Additionally, ontologies and extractors for requirements defined in the Polarion ISO 26262 templates [17] are created. These extractors preprocess and label data. The preprocessing steps during extraction allow to add semantic context to the textual information about requirements and elements from the model-based design. Moreover, further ontologies can be added any time to extend the functionality of the traceability recommender tool.

Based on the information gathered in the semantic data layer, the traceability recommender tool allows users to choose a model to represent requirements and elements from model-based design into a common vector space, to compute their similarities, e.g., using the cosine similarity measure [18]. The most similar items are taken by the tool as candidate trace elements and presented as recommendations to the user. One of the possible semantic layers can be GraphDB [19], an RDF database for knowledge graphs. In the next section, an example of the usage of the built-in semantic vector package [12] will be shown.

4. Use-Case

In this section, we present two possible outputs of the traceability recommender tool. Here, an exemplary functionality of a *Seat Heater* in a car is taken as a use case. It consists of a set of requirements specified in textual form (requirement types are presented

	entity	id	title	description	score
1	http://ontology.siemens.com/plm/iso-26262/data/SeatHeater_ISO26262/SHIS-202	"SHIS-202"	"SWSafReq18 - activate seat heater degraded mode"	"When the seat heater is powered, the seat heater supervisor shall activate the seat heater Degraded Mode (via power cutoff) if any failure is detected continuously for 200ms."	"0.7616024157497582"^^xsd:double
2	http://ontology.siemens.com/plm/iso-26262/data/SeatHeater_ISO26262/SHIS-195	"SHIS-195"	"SWSafReq11 - determine seat heater mode as specified"	"The seat heater supervisor shall determine the Seat Heater Mode status as specified."	"0.7476084310847307"^^xsd:double
3	http://ontology.siemens.com/plm/iso-26262/data/SeatHeater_ISO26262/SHIS-196	"SHIS-196"	"SWSafReq12 - evaluate seat heater mode request"	"The seat heater supervisor shall evaluate the mode request conditions based on Seat Heater Mode and their timing as specified."	"0.7257084120587818"^^xsd:double

Figure 3: Output example of the traceability recommender tool: list of requirements to trace for an AUTOSAR element.

in Figure 1), a functional design, a physical design, and an AUTOSAR model. The relevant information from all these artifacts is successfully extracted and preprocessed into our semantic data layer.

In the first use case, the traceability recommender tool is used to find requirements which are implemented by a specific element in the AUTOSAR model. Assuming that an engineer is working on the AUTOSAR model and defined an element of the type *RPortPrototype*, which is named **pt_seatheaterMode**. The traceability recommender tool returns three SW requirements from the Polarion specification, which are related to the *RPortPrototype* **pt_seatheaterMode**, as depicted in Figure 3. The first column of the table contains the full path to the SW requirement in the semantic data layer. The second column contains the id of the SW requirement in the Polarion project. The third and the fourth columns represent the title and the description of the requirement. Finally, the fifth column contains the similarity score, which tells how similar is the provided element in the model-based design to the suggested requirement. The information in the table can now be used by the engineer to create trace links between the suggested requirements and the AUTOSAR element. Hence, consistency and completeness in the mapping of the HW & SW (safety) requirements to AUTOSAR elements, which realize these requirements, can be achieved.

In a second use case, the engineer wants to verify that all safety requirements from a given specification are addressed in the model-based design. In this scenario, our tool recommends trace links for a specific SW requirement to the elements in which implement the requirement the AUTOSAR model. For instance, for the requirement with the title: *"Activate seat heater degraded mode"* and the description: *"When the seat heater is powered, the seat heater supervisor shall activate the seat heater Degraded Mode (via power cutoff) if any failure is detected continuously for 200ms."* the traceability recommender tool suggests trace links to the ports, which include such words as *mode*, *cutoff*, *temperature*. These words are either exisiting in the requirement name or description or have a similar meaning. Moreover, our tool suggests trace links to software component types, which are either connected with the mentioned ports or have similar words in their description. Some of the recommended AUTOSAR elements for which a trace link should be created

	matchedPortID	score
1	http://ontology.siemens.com/plm/autosar/data/SeatHeater_2206/SWC/co_seatheaterSupervisor/pt_seatheaterMode	"0.5319052413931838"^^xsd:double
2	http://ontology.siemens.com/plm/autosar/data/SeatHeater_2206/SWC/co_seatheaterSupervisor	"0.5085040265284687"^^xsd:double
3	http://ontology.siemens.com/plm/autosar/data/SeatHeater_2206/SWC/co_seatheaterSupervisor/pt_cutoff	"0.48731921250619803"^^xsd:double
4	http://ontology.siemens.com/plm/autosar/data/SeatHeater_2206/SWC/co_seatheaterSupervisor/pt_seatTemperature	"0.4823829703307479"^^xsd:double
5	http://ontology.siemens.com/plm/autosar/data/SeatHeater_2206/SWC/co_seatheaterSupervisor/pt_temperature	"0.43138115485747036"^^xsd:double

Figure 4: Output example of the traceability recommender tool: list of AUTOSAR elements to trace for an SW requirement.

are depicted in Figure 4. Based on this information the engineer can create trace links between the requirement and the suggested AUTOSAR elements. Especially, when the requirements of the AUTOSAR model are modified, the recommender tool helps engineers to detect inconsistencies and eases the adaptation of the trace links to the performed changes.

5. Conclusion and Further Research

Ensuring that all requirements are implemented correctly, is a major challenge in the development of modern E/E systems. Establishing and maintaining trace links between requirements and the model-based design are hard to achieve manually due to the increasing system complexity. In this paper, we introduce an AI-based recommender tool. This tool extends the Siemens toolchain for the development of automotive E/E systems and supports engineers to create bi-directional trace links between requirements written in natural language and Capital^{TM} software, AUTOSAR, SysML, UML, or Arcadia models for the system/software design.

As next steps, we will generalize the existing approach, since the development of the recommender is performed on specific datasets. Moreover, we plan sophisticated empirical studies on the datasets and developed similarity algorithms. To increase support for the engineers, we will incorporate reinforcement learning approaches in the recommender tool. Hence, the recommender can self-improve based on confirmed generated traces. Moreover, we plan to implement data obfuscation methods in the data extraction process, since it is often not possible to share data between different companies.

References

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

[2] Int. Organization for Standardization (ISO), ISO/TC 8/SC 8 Ship design, ISO 17894:2005 Ships and marine technology – General principles for the development and use of programmable electronic systems in marine applications, 2005.

[3] Society of Automotive Engineers Inc. (SAE), ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, 1996.
[4] Automotive SPICE, Standards Development Organisation, 2022. URL: https://www.automotivespice.com/.
[5] OMG, Systems Modeling Language (SysML), 2017. URL: http://www.omg.org/spec/SysML/.
[6] P. Roques, MBSE with the ARCADIA Method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), Toulouse, France, 2016, pp. 1–8. URL: https://hal.science/hal-01258014.
[7] Siemens, Architecture-driven E/E Systems Development Flow, White Paper, 2021. URL: https://resources.sw.siemens.com/en-US/white-paper-ee-systems-development-flow.
[8] AUTOSAR initiative, AUTOSAR Website - Standards, 2018. URL: https://www.autosar.org/standards/classic-platform.
[9] S. T. Maja Milicic Brandt, Industrial Ontologies at Siemens, 2022. https://ontocommons.eu/sites/default/files/20210607_MajaMilicicBrandt_Ontocommons.pdf.
[10] M. Vierlboeck, D. Dunbar, R. Nilchiani, Natural language processing to extract contextual structure from requirements, 2022, pp. 1–8. doi:10.1109/SysCon53536.2022.9773855.
[11] Y. Li, J. Cleland-Huang, Ontology-based trace retrieval, in: 2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE), 2013, pp. 30–36. doi:10.1109/TEFSE.2013.6620151.
[12] D. Widdows, T. Cohen, The semantic vectors package: New algorithms and public tools for distributional semantics, in: 2010 IEEE Fourth International Conference on Semantic Computing, 2010, pp. 9–15. doi:10.1109/ICSC.2010.94.
[13] R. Rasiman, F. Dalpiaz, S. España, How effective is automated trace link recovery in model-driven development?, in: Requirements Engineering: Foundation for Software Quality: 28th International Working Conference, REFSQ 2022, Birmingham, UK, Proceedings, Springer-Verlag, 2022.
[14] J. Guo, J. Cheng, J. Cleland-Huang, Semantically enhanced software traceability using deep learning techniques, in: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 2017, pp. 3–14. doi:10.1109/ICSE.2017.9.
[15] N. Ali, Y.-G. Gueheneuc, G. Antoniol, Trust-based requirements traceability, in: 2011 IEEE 19th International Conference on Program Comprehension, 2011, pp. 111–120. doi:10.1109/ICPC.2011.42.
[16] J. Cleland-Huang, B. Berenbach, S. Clark, R. Settimi, E. Romanova, Best practices for automated traceability, Computer 40 (2007) 27–35. doi:10.1109/MC.2007.195.
[17] Siemens, Polarion Extensions, 2022. URL: https://extensions.polarion.com/.
[18] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK, 2008. URL: http://nlp.stanford.edu/IR-book/information-retrieval-book.html.
[19] GraphDB, Semantic Graph Database, 2023. URL: https://graphdb.ontotext.com/.