Automated extraction of rules and knowledge from risk analyses: a ventilation unit demo

Bram Steenwinckel, Pieter Heyvaert, Dieter De Paepe, Olivier Janssens, Sander Vanden Hautte, Anastasia Dimou, Filip De Turck, Sofie Van Hoecke, and Femke Ongenae

Ghent University - imec, IDLab, Ghent, Belgium
`bram.steenwinckel@ugent.be`

Abstract. Assessing upfront the causes and effects of failures is an important aspect of system manufacturing. Nowadays, these analyses are performed by a large number of experts. To enable semantic unification and easy operationalization of these risk analyses, this paper demonstrates an approach to automatically map the captured information into an ontology and accompanying rules. The approach is demonstrated with a use case to identify anomalies and their causes within a ventilation unit.

1 Introduction

Analysing upfront the risks and their potential causes are of high importance to let manufacturers predict & prevent system failures. However, defining unwanted behaviour, i.e. anomalies, and pinpointing its cause requires much expert knowledge. Experts possess the contextual background information needed to model these anomalies & effects, using existing risk analysis tools, such as Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) [2].

This process can be time-consuming when applied thoroughly. Due to a large number of people involved, each having expertise in other parts of the system, ambiguities are common. Ontology-based approaches have been proposed to resolve this, [3,4] to structure the risk analyses, impose a common vocabulary and semantically link the faults and causes. However, most system experts are not familiar with ontology design, which makes these approaches difficult to implement and maintain.

This paper demonstrates an approach to automatically extract the knowledge from FMEA tables and FTA trees into domain-specific ontologies and rules. These can be used to easily analyse and spot the ambiguities, duplicates and inconsistencies in a structured format. The proposed approach enables incremental knowledge incorporation by multiple domain experts while enabling some primary Anomaly Detection (AD) analysis.

2 Mapping risk analyses to ontologies & rules

A FMEA is performed per system component to get a full view of all the possible failures that can occur, their effects and their possible causes. This analysis results in an FMEA table, as visualised in Figure 1a. In contrast, a FTA identifies the link between a particular observation and a possible failure, as shown
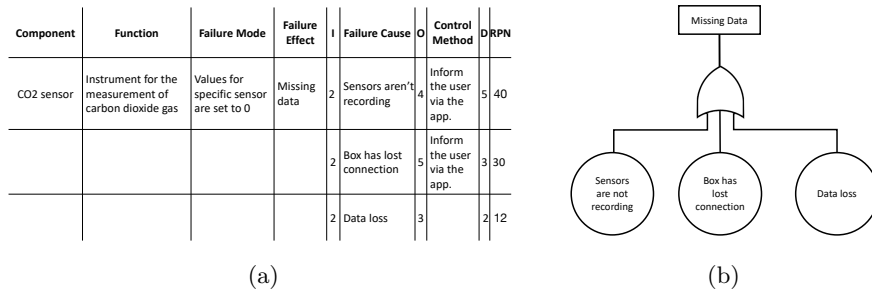
Component	Function	Failure Mode	Failure Effect	I	Failure Cause	O	Control Method	D	RPN
CO2 sensor	Instrument for the measurement of carbon dioxide gas	Values for specific sensor are set to 0	Missing data	2	Sensors aren't recording	4	Inform the user via the app.	5	40
				2	Box has lost connection	5	Inform the user via the app.	3	30
				2	Data loss	3		2	12

(a)	(b)

Fig. 1: Example of a (a) FMEA and (b) FTA

in Figure 1b. Our mapping approach allows to easily combine the two, by offering methods that automatically translate the FMEA tables and FTA trees to ontologies and accompanying rules that correlate the different observations, system components, faults and causes. Data generated by the systems can then be semantically annotated using these ontologies, while a reasoner can be used to automatically derive the correlated anomalies and faults with the generated rules. The full mapping approach is given in Figure 2. It consists of two main parts, domain knowledge transformation and rule generation.

The **domain knowledge transformation** process is shown on the bottom of Figure 2. Entries from FMEA tables can be mapped to RDF using a mapping language, resulting in a domain-specific ontology with risk analysis information. As such, anomaly knowledge can be extracted from the FMEA, and the causes of these anomalies can be derived by following the semantic links.

To enable this, an upper ontology, called Folio[1], was designed. It captures and correlates all application-independent concepts that occur within FMEA tables and AD methods, such as `FailureCause`, `FailureEffect`, `Criticality` and `DetectionMethod`. The Semantic Sensor Network (SSN) ontology[2] is used to describe the various system components. Relationships were defined in Folio to correlate the SSN concepts with possible failures and effects. The Folio ontology functions as a blueprint for further domain-specific ontology designs.

To translate the information inside the FMEA table, usually in the CSV format, to ontological Folio concepts, i.e. RDF, the mapping language RML [1] is used. To create the mappings, i.e. RML rules, the graphical user interface RMLEditor was used. The RML rules map the different columns in the FMEA table on concepts of the Folio and SSN ontologies. Afterwards, the RMLMapper, a tool to execute RML rules, is used to generate the domain-specific ontologies for the mapped FMEA tables. The mappings ensure that for each cell in the FMEA table, a new concept is created in the ontology, which is a subconcept of the concept on which the column is mapped according to the rules. For example, if we consider the 4th cell on the first row of Figure 1a, the RMLMapper will create a new class `Missingdata` in the ontology, which has as superclass the `FailureEffect` class. The rules defined by the RMLEditor have to be specified only once. As such, these mappings[3] can be re-used to translate any FMEA table
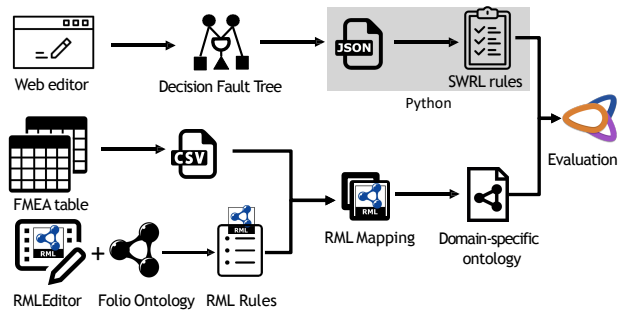

Fig. 2: Overview of the full mapping approach

that is created according to the standard FMEA structure. If a new column is added, a new rule can easily be created to map this column to the Folio ontology by using the RMLEditor. Updating or incorporating new information inside the tables can be performed without defining additional RML rules. Rerunning the RML mapper suffices.

The **rule generation** process is visualized on the top of Figure 2. Observations generated by the system's sensors will be the main entry point for further analyses of unwanted behaviour. This analysis can be expressed as a FTA tree that determines how new observations should be interpreted in the context of failures and effects. Classical FTA does not allow complex analysis of observations. Therefore, our approach supports an extension, namely decision fault trees (DFTs), which allow tests on the edges of the tree to get a combination and more complex interpretation of the system observation values. A user interface was designed to build such DFTs. In this editor, descriptions of the observation and failure nodes can be given. These different node concepts should align with the concepts defined in the FMEA. Tests describing the relations between these observations and failures can be added or adapted. To transfer these tests to decisions, a script[4] was constructed that derives Semantic Web Rule Language (SWRL) rules from the DFTs. Similarly to the construction of FMEA tables, additions and updates to the trees do not require changes to the scripts but rerunning them suffices.

The ontologies and rules generated by using the full translation approach can be incorporated in a knowledge-based monitoring system to identify anomalies and their causes continuously. The monitoring system can semantically annotate new observations using the domain-specific ontologies, generated by mapping the FMEA tables. A reasoner can process the generated SWRL rules and links defined in the ontologies to determine whether failures are occurring and what their possible causes are.

3 Demonstrator: a ventilation use case

The Healthbox 3.0 is a ventilation unit for residential buildings, offered by Renson Ventilation NV, which controls the airflow 24/7 to enhance indoor air quality.

---

[4] Script: https://github.com/IBCNServices/Folio-Ontology/blob/master/swr_builder.py

It extracts polluted air from different rooms in a house. One central fan is responsible for creating the requested airflow. The Healthbox is equipped with a series of sensors of which the observations can be used to detect possible failures. When for instance the requested airflow could not be achieved, the valves could be malfunctioning, or the fan speed can be too low. The described approach was used to automatically generate ontologies and rules that capture the knowledge of the Renson experts for the detection of ventilation anomalies and their causes. Renson performed a FMEA analysis for each component of the Healthbox to get a full view on all the possible failures, their effects and causes. The demo will show how the designed RML rules, see the previous section, can be used to easily map the contents of this FMEA to a Ventilation ontology, which is an extension of the SSN en Folio ontologies. The user will also be able to add new rows and column to the FMEA. It will be demonstrated how the RMLEditor can be used to map the new columns to the Folio ontology in a user-friendly manner. It will be shown that the new rows automatically get transformed to the ventilation ontology without requiring to adapt the mappings.

The designed tree editor was used by experts to construct DFTs that link the system observations to possible failures of the Healthbox. The example DFT describes a possibly malfunctioning CO_2 sensor because the measured values are not in the acceptable range provided by the Renson experts. This sensor is important for the correct functioning of the valves, enabling the unit to ventilate when needed. It will be shown that this tree editor automatically exports JSON files that can be translated to SWRL rules by using the script detailed in the previous section. The user will be able to adapt or construct DFTs during the demonstration with the user-friendly editor. It will be shown that the script is also able to translate these new or adapted DFTs to SWRL rules.

The demonstration ends by combining the Ventilation ontology and SWRL rules in Protégé with some generated sensor observation instances. The reasoner is executed to illustrate that the rules extract the correct failures from these observations and link them to the possible causes through the ontology [5].

References

1. Heyvaert, P., et al.: Rmleditor: a graph-based mapping editor for linked data mappings. In: International Semantic Web Conference. pp. 709–723. Springer (2016)
2. Peeters, J., et al.: Improving failure analysis efficiency by combining fta and fmea in a recursive manner. Reliability engineering & system safety **172**, 36–44 (2018)
3. Rehman, Z., et al.: An ontology to support semantic management of fmea knowledge. International Journal of Computers, Communications & Control **11**(4) (2016)
4. Venceslau, A., et al.: Ontology for computer-aided fault tree synthesis. In: Emerging Technology and Factory Automation (ETFA), 2014 IEEE. pp. 1–4. IEEE (2014)

---

[5] The whole demo is also available on https://youtu.be/S3pe47Sn2Qs.