Reliability Engineering and System Safety 120 (2013) 150–162

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

OASIS: An automotive analysis and safety engineering instrument

Roland Mader [a,b,*], Eric Armengaud [a,c], Gerhard Grießnig [a], Christian Kreiner [b], Christian Steger [b], Reinhold Weiß [b]

[a] AVL List GmbH, Austria
[b] Graz University of Technology, Austria
[c] Virtual Vehicle Competence Center, Austria

ARTICLE INFO

Available online 6 July 2013

Keywords:
FMEA
FTA
Functional safety
PHA
System architecture

ABSTRACT

In this paper, we describe a novel software tool named OASIS (AutOmotive Analysis and Safety EngIneering InStrument). OASIS supports automotive safety engineering with features allowing the creation of consistent and complete work products and to simplify and automate workflow steps from early analysis through system development to software development. More precisely, it provides support for (a) model creation and reuse, (b) analysis and documentation and (c) configuration and code generation. We present OASIS as a part of a tool chain supporting the application of a safety engineering workflow aligned with the automotive safety standard ISO 26262. In particular, we focus on OASIS' (1) support for property checking and model correction as well as its (2) support for fault tree generation and FMEA (Failure Modes and Effects Analysis) table generation. Finally, based on the case study of hybrid electric vehicle development, we demonstrate that (1) and (2) are able to strongly support FTA (Fault Tree Analysis) and FMEA.

1. Introduction

Powertrain[1] electrification of vehicles transforms powertrains into complex, mechatronic systems. Due to the safety-criticality of the embedded system, sensors and actuators, they are developed according to safety standards like the automotive functional safety standard ISO 26262 [1] that provides a risk-based development process relying on quality management.

The standard requires the application of a concept phase. This phase aims at the functional description of a newly developed vehicle, the early application of PHA (Preliminary Hazard Analysis) and the derivation of safety requirements (top-level and functional). The standard also requires a subsequent system level development phase. In this phase, the system architecture (architecture of embedded system, connected sensors and controlled actuators) is designed and analyzed using FTA (Fault Tree Analysis) and FMEA (Failure Modes and Effects Analysis). Furthermore, technical safety requirements are defined. In addition, safety requirements and ASILs (Automotive Safety Integrity Levels) are allocated to the components of the system architecture. ISO 26262 also requires a subsequent software level development phase including the definition of software requirements and the design, implementation and integration of software.

Tool support for the application of the required phases exists. However, due to lacking support for safety engineering, (1) the concept phase and the system level development phase are difficult to apply. Another difficulty is (2) the effective transition from the system level development phase to the software level development phase. Both hinder the efficient creation of a complete and consistent set of work products. To overcome these problems, adequate tool support for safety engineering is required.

The contribution of this paper is a novel tool named OASIS (AutOmotive Analysis and Safety EngIneering InStrument). OASIS supports a safety engineering workflow aligned with ISO 26262's concept phase, system level development phase and software level development phase. OASIS's features allow (a) model[2] creation and reuse, (b) analysis and documentation as well as (c) configuration and code generation. These features allow the creation of consistent and complete work products and to simplify and automate workflow steps. In particular, this paper focuses on OASIS' (1) support for property checking and model correction as well as its (2) support for fault tree generation and FMEA table generation and describes how (1) and (2) aid the application of FTA and FMEA.

The paper is organized as follows. Section 2 describes approaches supporting FTA or FMEA and how OASIS' support

* Corresponding author at: AVL List GmbH, Hans-List-Platz-1, 8020 Graz, Austria. Tel.: +43 6648509202.
E-mail address: roland.mader@avl.com (R. Mader).

[1] *Powertrain* denotes the group of vehicle components (e.g. engine, transmission or driving shafts) that generate power and deliver it to the road surface.

[2] *Model* denotes a structured, computerized artifact used to support communication, documentation, analysis and synthesis as part of the safety engineering.

0951-8320/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2013.06.045

for applying FTA and FMEA is different. Section 3 illustrates OASIS as well as its use as part of a tool chain for applying a safety engineering workflow. OASIS' support for property checking and model correction as well as its support for fault tree generation and FMEA table generation are described in Sections 4 and 5. Section 6 explains how OASIS' features aid the application of FTA and FMEA based on the case study of HEV (Hybrid Electric Vehicle) development. Section 7 concludes the paper.

2. Related work

The approaches described in [2–12] use models describing structure and/or behavior of a system (typically of a computer-based system that shall be dependable, reliable or safe). These models are complemented with quantitative or qualitative information concerning the behavior of the system in the presence of faults (typically about faults and failures and their propagation). These underlying models are used by all approaches as input to fault tree generation and/or FMEA table generation, supporting the application of FTA and/or FMEA. In contrast to the reviewed approaches, OASIS does not only support the application of FTA and FMEA by generating fault trees (FTA Generator) and FMEA tables (FMEA Generator) from a system model. Instead, OASIS goes one step further and also supports the creation of the system model by the automatic identification of imperfections (Property Checker) and the automatic identification and application of corrective measures (Model Corrector). In the following, the related works are described individually.

An approach that combines system architecture modeling and FTA is described in [2]. The approach allows the assessment of an evolving system design. A system model is input to HAZOP (Hazard and Operational Studies). Each component of the system model is analyzed, and component failure modes are determined. The HAZOP results in a model defining failure modes that can be observed at the component outputs as results of internal component malfunctions as well as deviating component inputs. In [3], an extension of [2] is presented that allows FMEA table generation. In [4], the extended approach is integrated with an EAST-ADL (Electronics Architecture and Software Technology-Architecture Description Language) modeling tool using a model transformation technique. This allows the generation of fault trees and FMEA tables from EAST-ADL models.

In [5], an approach to fault tree generation from system models is presented. The approach foresees performing reliability analysis in parallel with system design and using UML to model fault-tolerant, software-intensive systems. A modeling methodology is presented requiring the use of stereotypes to express concepts such as hardware, redundancy, spares, dependencies and reconfiguration. A three-pass algorithm is presented that allows generating fault tree code.

A methodology combining safety analyses and a component-oriented, model-based software engineering approach is described in [6]. The authors aim at supporting safety analyses in the earlier stages of development. A hierarchical model for component-based software engineering is available. The model allows defining a failure specification and a failure realization as well as a functional specification and a functional realization for each software component. Fault trees can be generated from the component model.

In [7] tool support for automated FMEA generation is presented. Input to the presented method is a component model of a system including so-called safety interfaces that can automatically be generated. Safety interfaces can be seen as formal descriptions of the components in terms of failures affecting the components. From the safety interface descriptions, cFMEAs (Component Failure Modes and Effects Analyses) can be created for each component. Subsequently, the cFMEAs are input to the generation of a system-level FMEA.

The authors of [8] present a novel methodology for the construction of fault trees from system Simulink models. The methodology foresees the manual creation of a Simulink model and the according complementation of the model with other information required for fault tree generation. This model is input to fault tree generation.

The authors of [9] describe a tool set named COMPASS making use of a formal semantics for AADL (Architecture Analysis & Design Language). The approach foresees the creation of a hierarchical system model describing the behavior of the system under normal conditions. This model is complemented with an error model expressing how the system can fail. The presented tool set includes the capability of generating fault trees and FMEA tables from the system model and the error model. In [10], the COMPASS tool set was evaluated using the case study of a satellite platform under development. Among other activities, a fault tree consisting of 66 nodes and an FMEA table were automatically generated in course of this case study.

In [11], an approach to fault tree generation from UML models is presented. The approach aims at supporting reliability analyses in early design stages, when the overall system architecture is still subject to refinement. They separate application-independent information from application-dependent information to sustain reuse and to avoid remodeling. To achieve this, separate UML-profiles for architectural models and application models are used. These models are input to fault tree generation.

In [12], an approach to the automatic generation of static fault trees from system models that are specified with SysML (Systems Modeling Language) is described. The authors use internal block diagrams and sequence diagrams to describe a system. These diagrams are an input to the automatic generation of an RCM (Reliability Configuration Model). Then, an SFTM (Static Fault Tree Model) is developed to generate static fault trees from the RCM specifications. The approach is experimentally evaluated using the case study of a fault-tolerant parallel processor.

3. OASIS

OASIS is a novel software tool for automotive safety engineering (see Fig. 1). It provides features that allow simplifying and automating workflow steps from early analysis through system development to software development (an overview of the proposed workflow is presented in Fig. 2). OASIS can be combined with other tools to form a tool chain supporting an automotive safety engineering workflow aligned with ISO 26262. An overview of OASIS' architecture and features is presented in Section 3.1. The combination of OASIS with other software tools in order to form a tool chain is presented in Section 3.2. The safety engineering workflow supported by the resulting tool chain is described in Section 3.3.

3.1. Architecture and features

OASIS is designed as a plug-in for the software tool Papyrus for UML [13]. Thus, OASIS is no standalone tool and must be used in combination with Papyrus for UML. This relation is illustrated in Fig. 1.

Papyrus for UML is an Eclipse-based open-source tool allowing EAST-ADL modeling. EAST-ADL [14] is a domain-specific language and adapted to the needs of the automotive domain. It is *diagrammatic* [15] such as UML (Unified Modeling Language). It consists of syntactic elements such as boxes, ovals, lines or arrows.