[1] M. Adachi, Y. Papadopoulos, S. Sharvia, D. Parker, and T. Tohdo. An approach to optimization of fault tolerant architectures using HiP-HOPS. *Software: Practice and Experience*, 2011. [bib | DOI]

New processes for the design of dependable systems must address both cost and dependability concerns. They should also maximize the potential for automation to address the problem of increasing technological complexity and the potentially immense design spaces that need to be explored. In this paper we show a design process that integrates system modelling, automated dependability analysis and evolutionary optimization techniques to achieve the optimization of designs with respect to dependability and cost from the early stages. Computerized support is provided for difficult aspects of fault tolerant design, such as decision making on the type and location of fault detection and fault tolerant strategies. The process is supported by HiP-HOPS, a scalable automated dependability analysis and optimization tool. The process was applied to a Pre-collision system for vehicles at an early stage of its design. The study shows that HiP-HOPS can overcome the limitations of earlier work based on Reliability Block Diagrams by enabling dependability analysis and optimization of architectures that may have a network topology and exhibit multiple failure modes.

Keywords: dependability analysis, fault tolerance, active safety, multi-objective optimization, genetic algorithms

[2] R. Adler, D. Domis, K. Höfig, S. Kemmann, T. Kuhn, J.-P. Schwinn, and M. Trapp. Integration of component fault trees into the UML. In *Models in Software Engineering*, volume 6627 of *Lecture Notes in Computer Science*, pages 312-327. 2011. [bib | DOI]

Efficient safety analyses of complex software intensive embedded systems are still a challenging task. This article illustrates how model-driven development principles can be used in safety engineering to reduce cost and effort. To this end, the article shows how well accepted safety engineering approaches can be shifted to the level of model-driven development by integrating safety models into functional development models. Namely, we illustrate how UML profiles, model transformations, and techniques for multi language development can be used to seamlessly integrate component fault trees into the UML.

[3] ARINC Report 664P7-1. *Aircraft Data Network, Part 7: Avionics Full Duplex Switched Ethernet (AFDX) Network*, Sept. 2009. [bib]

The purpose of this document is to define a deterministic network: Avionics Full Duplex Switched Ethernet (AFDX). AFDX is a trademark of Airbus and is used with permission. This document also highlights the additional performance requirements of avionics systems within the context of AFDX.

[4] R. W. Butler. A primer on architectural level fault tolerance. Technical Report TM-2008-215108, NASA, Feb 2008. [bib]

[5] M. Chérèque, D. Powell, P. Reynier, J.-L. Richier, and J. Voiron. Active replication in delta-4. In *FTCS*, pages 28-37, 1992. [bib | http]

Keywords: dblp

[6] E. M. Clarke and P. Zuliani. Statistical model checking for cyber-physical systems. In *Automated Technology for Verification and Analysis*, pages 1-12. Springer, 2011. LNCS 9669. [bib]

[7] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007. [bib]

[8] M. Clavel and J. Meseguer. Reflection and strategies in rewriting logic. *Electr. Notes Theor. Comput. Sci.*, 4:126-148, 1996. [bib]

[9] S. Distefano and A. Puliafito. Dependability evaluation with dynamic reliability block diagrams

and dynamic fault trees. *IEEE Transactions on Dependable and Secure Computing*, 6(1):4-17, 2009. [bib | DOI]

Dependability evaluation is an important often-mandatory step in designing and analyzing (critical) systems. Introducing control and/or computing devices to automate processes increases the system complexity, with an impact on the overall dependability. This occurs as a consequence of interferences, dependencies, and other similar effects that cannot be adequately managed through formalisms such as reliability block diagrams (RBDs), fault trees (FTs), and reliability graphs (RGs), since the statistical independence assumption is not satisfied. In addition, more enhanced notations such as dynamic FTs (DFTs) might not be adequate to represent all the behavioral aspects of dynamic systems. To overcome these problems, we developed a new formalism derived from RBD: the dynamic RBD (DRBD). DRBD exploits the concept of dependence as the building block to represent dynamic behaviors, allowing us to compose the dependencies and adequately manage the arising conflicts by means of a priority algorithm. In this paper, we explain how we can use the DRBD notation by specifying a practical methodology. Starting from the system knowledge, the proposed methodology drives to the overall system reliability evaluation through the entire phases of modeling and analysis. Such a technique is applied to an example taken from the literature, consisting of a distributed computing system.

Keywords: dependability evaluation;distributed computing system;dynamic fault trees;dynamic reliability block diagrams;system reliability evaluation;distributed processing;fault trees;software reliability

[10] M. Forster and D. Schneider. Flexible, any-time fault tree analysis with component logic models. In *Symposium on Software Reliability Engineering (ISSRE)*, pages 51-60, Nov. 2010. [bib | DOI]

This article presents a novel approach to facilitating fault tree analysis during the development of software-controlled systems. Based on a component-oriented system model, it combines second-order probabilistic analysis and automatically generated default failure models with a level-of-detail concept to ensure early and continuous analysability of system failure behaviour with optimal effort, even in the presence of incomplete information and dissimilar levels of detail in different parts of an evolving system model. The viability and validity of the method are demonstrated by means of an experiment.

Keywords: any time fault tree analysis;component logic models;component oriented system;failure behaviour;incomplete information;optimal effort;probabilistic analysis;software controlled systems;fault trees;formal logic;probability;safety-critical software;

[11] X. Ge, R. Paige, and J. McDermid. Probabilistic failure propagation and transformation analysis. In *Computer Safety, Reliability, and Security (SAFECOMP)*, volume 5775 of *Lecture Notes in Computer Science*, pages 215-228. 2009. [bib | DOI]

A key concern in safety engineering is understanding the overall emergent failure behaviour of a system, i.e., behaviour exhibited by the system that is outside its specification of acceptable behaviour. A system can exhibit failure behaviour in many ways, including that from failures of individual or a small number of components. It is important for safety engineers to understand how system failure behaviour relates to failures exhibited by individual components. In this paper, we propose a safety analysis technique, failure propagation and transformation analysis (FPTA), which automatically and quantitatively analyses failures based on a model of failure logic. The technique integrates previous work on automated failure analysis with probabilistic model checking supported by the PRISM tool. We demonstrate the technique and tool on a small, yet realistic safety-related application.

Keywords: failure, safety analysis, probabilistic analysis, component-based system

[12] X. Ge, R. Paige, and J. McDermid. Analysing system failure behaviours with PRISM. In *International Conference on Secure Software Integration and Reliability Improvement Companion*, pages 130-136, June 2010. [bib | DOI]

The verification of safety-critical systems using formal techniques is not something new. Traditionally, safety-critical systems are verified using hazard analysis techniques, e.g., fault tree analysis. As safety-critical systems have become larger and more complex, several analysis techniques with compositional capabilities were developed. However, these techniques were not able to analyse stochastic systems. In this paper, we present a model-based compositional safety analysis technique (i.e., failure propagation analysis) and explore the feasibility of integrating this safety analysis technique with techniques of probabilistic model checking, more precisely the PRISM model checker. By doing so, we make it possible to rigorously verify a model while system failure behaviours are quantitatively analysed.

Keywords: PRISM model checker;compositional capability;failure analysis;fault tree analysis;formal techniques;hazard analysis techniques;model-based compositional safety analysis technique;probabilistic model checking;safety critical system verification;stochastic system;failure analysis;formal verification;safety-critical software;stochastic processes;

[13] L. Gong, P. Lincoln, and J. Rushby. Byzantine agreement with authentication: Observations and applications in tolerating hybrid and link faults. In R. K. Iyer, M. Morganti, W. K. Fuchs, and V. Gligor, editors, *Dependable Computing for Critical Applications-5*, volume 10 of *Dependable Computing and Fault Tolerant Systems*, pages 139-157, Champaign, IL, sep 1995. IEEE Computer Society. [bib | http]

[14] R. C. Hammett and P. S. Babcock. Achieving 10^{-9} dependability with drive-by-wire systems. In *SAE 2003 World Congress and Exhibition, March 2003,Session: Safety Critical Systems*, Detroit, March 2003. [bib | http]

[15] R. Isermann, R. Schwarz, and S. Stolzl. Fault-tolerant drive-by-wire systems. *Control Systems, IEEE*, 22(5):64-81, Oct. 2002. [bib | DOI]

The article begins with a review of electronic driver assisting systems such as ABS, traction control, electronic stability control, and brake assistant. We then review drive-by-wire systems with and without mechanical backup. Drive-by-wire systems consist of an operating unit with an electrical output, haptic feedback to the driver, bus systems, microcomputers, power electronics, and electrical actuators. For their design safety, integrity methods such as reliability, fault tree and hazard analysis, and risk classification are required. Different fault-tolerance principles with various forms of redundancy are considered, resulting in fail-operational, fail-silent, and fail-safe systems. Fault-detection methods are discussed for use in low-cost components, followed by a review of principles for fault-tolerant design of sensors, actuators, and communication. We evaluate these methods and principles and show how they can be applied to low-cost automotive components and drive-by-wire systems. A brake-by-wire system with electronic pedal and electric brakes is then considered in more detail, showing the design of the components and the overall architecture. Finally, we present conclusions and an outlook for further development of drive-by-wire systems.

Keywords: ABS; antilock brake systems; brake assistant; brake pedal; bus systems; design safety; drive-by-wire systems; electrical actuators; electrical output; electronic driver assisting systems; electronic stability control; fail-operational systems; fail-safe systems; fail-silent systems; fault tree analysis; fault-detection methods; fault-tolerance; haptic feedback; hazard analysis; integrity methods; low-cost automotive components; mechanical backup; power electronics; redundancy; reliability; risk classification; steering wheel; traction control; automobiles; automotive electronics; brakes; braking; electric actuators; fault diagnosis; fault tolerance; haptic interfaces; redundancy; safety

[16] B. Kaiser, C. Gramlich, and M. Förster. State/event fault trees - a safety analysis model for software-controlled systems. *Reliability Engineering & System Safety*, 92(11):1521-1537, 2007. [bib | DOI]

Safety models for software-controlled systems should be intuitive, compositional and have the expressive power to model both software and hardware behaviour. Moreover, they should provide quantitative results for failure or hazard probabilities. Fault trees are an accepted and intuitive model for safety analysis, but they are incapable of expressing state dependencies or temporal order of events. We propose to combine fault trees with an explicit State/Event semantics, using a graphical notation that is similar to Statecharts. Our new model, named State/Event Fault Trees (SEFTs),

subsumes both deterministic state machines suited to describe software behaviour, and Markov chains that model probabilistic failures, while keeping the visualisation of causal chains known from fault trees. We allow exponentially distributed probabilistic events, deterministic delays, and triggered events. The model provides a component concept, where components are connected by typed ports. Quantitative evaluation is achieved by translating the component models to Deterministic and Stochastic Petri Nets (DSPNs) and using an existing tool for analysis or simulation. This paper, which is an extended version of , revisits the model elements and the analysis procedure and provides a small case study of a fire alarm system, completed by an outlook on our tool project ESSaRel.

[17] N. Khakzad, F. Khan, and P. Amyotte. Safety analysis in process facilities: Comparison of fault tree and bayesian network approaches. *Reliability Engineering & System Safety*, 96(8):925-932, 2011. [bib | DOI]

Safety analysis in gas process facilities is necessary to prevent unwanted events that may cause catastrophic accidents. Accident scenario analysis with probability updating is the key to dynamic safety analysis. Although conventional failure assessment techniques such as fault tree (FT) have been used effectively for this purpose, they suffer severe limitations of static structure and uncertainty handling, which are of great significance in process safety analysis. Bayesian network (BN) is an alternative technique with ample potential for application in safety analysis. BNs have a strong similarity to FTs in many respects; however, the distinct advantages making them more suitable than FTs are their ability in explicitly representing the dependencies of events, updating probabilities, and coping with uncertainties. The objective of this paper is to demonstrate the application of BNs in safety analysis of process systems. The first part of the paper shows those modeling aspects that are common between FT and BN, giving preference to BN due to its ability to update probabilities. The second part is devoted to various modeling features of BN, helping to incorporate multi-state variables, dependent failures, functional uncertainty, and expert opinion which are frequently encountered in safety analysis, but cannot be considered by FT. The paper concludes that BN is a superior technique in safety analysis because of its flexible structure, allowing it to fit a wide variety of accident scenarios.

Keywords: Bayesian network, Fault tree analysis, Accident analysis, Uncertainty modeling

[18] R. M. Kieckhafer, C. J. Walter, A. M. Finn, and P. M. Thambidurai. The maft architecture for distributed fault tolerance. *IEEE Transactions on Computers*, 37:398-405, 1988. [bib]

[19] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition, 1997. [bib]

[20] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer. The time-triggered Ethernet (TTE) design. In *Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, pages 22-33, May 2005. [bib | DOI]

This paper presents the rational for and an outline of the design of a time-triggered (TT) Ethernet that unifies real-time and non-real-time traffic into a single coherent communication architecture. TT Ethernet is intended to support all types of applications, from simple data acquisition systems, to multimedia systems up to the most demanding safety-critical real-time control systems which require a fault-tolerant communication service that must be certified. TT Ethernet distinguishes between two traffic categories: the standard event-triggered Ethernet traffic and the time-triggered traffic that is temporally guaranteed. The event triggered traffic in TT Ethernet is handled in conformance with the existing Ethernet standards of the IEEE. The design of TT Ethernet has been driven by the requirement of certification of safety-critical configurations and an uncompromising stand with respect to the integration of legacy applications and legacy Ethernet hardware.

Keywords: communication architecture; data acquisition systems; event-triggered Ethernet traffic; fault-tolerant communication service; legacy Ethernet hardware; multimedia systems; safety-critical real-time control systems; time-triggered Ethernet design; IEEE standards; data acquisition; fault tolerant computing; local area networks; multimedia systems; open systems; real-time systems; telecommunication traffic;

[21] P. Lincoln and J. Rushby. Formal verification of an interactive consistency algorithm for the draper FTP architecture under a hybrid fault model. In *Compass '94 (Proceedings of the Ninth*

Annual Conference on Computer Assurance), pages 107-120, Gaithersburg, MD, jun 1994. IEEE Washington Section. [bib | http]

[22] N. Mahmud, Y. Papadopoulos, and M. Walker. A translation of state machines to temporal fault trees. In *International Conference on Dependable Systems and Networks Workshops*, pages 45-51, July 2010. [bib | DOI]

State Machines (SMs) are increasingly being used to gain a better understanding of the failure behaviour of safety-critical systems. In dependability analysis, SMs are translated to other models, such as Generalized Stochastic Petri Nets (GSPNs) or combinatorial fault trees. The former does not enable qualitative analysis, whereas the second allows it but can lead to inaccurate or erroneous results, because combinatorial fault trees do not capture the temporal semantics expressed by SMs. In this paper, we discuss the problem and propose a translation of SMs to temporal fault trees using Pandora, a recent technique for introducing temporal logic to fault trees, thus preserving the significance of the temporal sequencing of faults and allowing full qualitative analysis. Since dependability models inform the design of condition monitoring and failure prevention measures, improving the representation and analysis of dynamic effects in such models can have a positive impact on proactive failure avoidance.

Keywords: Pandora technique;combinatorial fault trees;condition monitoring measurement;dependability analysis;failure prevention measurement;generalized stochastic Petri Nets;proactive failure avoidance;safety-critical systems;state machines;temporal fault trees;temporal logic;Petri nets;finite state machines;temporal logic;trees (mathematics);

[23] G. Merle. *Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis*. PhD thesis, Ecole Normale Supérieure de Cachan, France, June 2010. [bib | www:]

[24] G. Merle, J.-M. Roussel, J.-J. Lesage, and A. Bobbio. Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events. *IEEE Transactions on Reliability*, 59(1):250-261, Mar. 2010. [bib | DOI]

Keywords: Boolean operators;Markov model;failure distribution;priority dynamic fault trees;priority dynamic gates;probabilistic algebraic analysis;repeated events;Boolean algebra;Markov processes;fault trees;probability;

[25] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73-155, 1992. [bib]

[26] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28(1):71-87, 1986. [bib]

[27] M. J. O'Donnell. Equational logic as a programming language. In Parikh [31], page 255. [bib]

[28] Y. Papadopoulos and C. Grante. Evolving car designs using model-based automated safety analysis and optimisation techniques. *Journal of Systems and Software*, 76(1):77-89, 2005. [bib | DOI]

Development processes in the automotive industry need to evolve to address increasing demands for integration of car functions over common networked infrastructures. New processes must address cost and safety concerns and maximize the potential for automation to address the problem of increasing technological complexity. In this paper, we propose a design process in which techniques for semi-automatic safety and reliability analysis of systems models are combined with multi-objective optimisation techniques to assist the gradual development of designs that can meet reliability and safety requirements and maximise profit within pragmatic development cost constraints. The proposed process relies on tools to automate some aspects of the design that we believe could be automated and thus simplified without loss of the creative input brought in the process by designers.

Keywords: Fault tree synthesis, Automated safety analysis, Software hazard analysis, Fault tolerance, Multi-objective optimization

[29] Y. Papadopoulos, J. McDermid, R. Sasse, and G. Heiner. Analysis and synthesis of the

behaviour of complex programmable electronic systems in conditions of failure. *Reliability Engineering & System Safety*, 71(3):229-247, Mar. 2001. [bib | DOI]

This paper introduces a new method for safety analysis which modifies, automates and integrates a number of classical safety analysis techniques to address some of the problems currently encountered in complex safety assessments. The method enables the analysis of a complex programmable electronic system from the functional level through to low levels of its hardware and software implementation. In the course of the assessment, the method integrates design and safety analysis and harmonises hardware safety analysis with the hazard analysis of software architectures. It also introduces an algorithm for the synthesis of fault trees, which mechanises and simplifies a large and traditionally problematic part of the assessment, the development of fault trees. In this paper, we present the method and discuss its application on a prototypical distributed brake-by-wire system for cars. We argue that the method can help us rationalise and simplify an inherently creative and difficult task and therefore gain a consistent and meaningful picture of how a complex programmable system behaves in conditions of failure.

Keywords: Automated safety analysis, Mechanical fault tree synthesis, Software hazard analysis, Safety cases

[30] Y. Papadopoulos and J. A. McDermid. Hierarchically performed hazard origin and propagation studies. In *Computer Safety, Reliability and Security*, volume 1698 of *Lecture Notes in Computer Science*, pages 688-688. 1999. [bib | DOI]

This paper introduces a new method for safety analysis called Hi-PHOPS (Hierarchically Performed Hazard Origin and Propagation Studies). HiP-HOPS originates from a number of classical techniques such as Functional Failure Analysis, Failure Mode and Effects Analysis and Fault Tree Analysis. However, it extends, automates and integrates these techniques in order to address some of the problems currently encountered in complex safety assessments. The method enables integrated assessment of a complex system from the functional level through to the low level of component failure modes. It mechanises and simplifies a large part of the analysis, the development of fault trees, and can guarantee the consistency of results. HiP-HOPS is currently supported by a tool called the Safety Argument Manager (SAM). In this paper we introduce the method and we show how it has helped us analyse and improve the safety of a distributed brake-by-wire system for cars.

[31] R. Parikh, editor. *Logics of Programs, Conference, Brooklyn College, June 17-19, 1985, Proceedings*, volume 193 of *Lecture Notes in Computer Science*. Springer, 1985. [bib]

[32] http://www.prismmodelchecker.org/. [bib]

[33] W. Steiner. *TTEthernet Specification*. TTA Group, 2008. Available at http://www.ttagroup.org. [bib]

[34] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch. TTEthernet: Time-Triggered Ethernet. In R. Obermaisser, editor, *Time-Triggered Communication*. CRC Press, Aug 2011. [bib]

[35] M. Walker, L. Bottaci, and Y. Papadopoulos. Compositional temporal fault tree analysis. In *Computer Safety, Reliability, and Security (SAFECOMP)*, volume 4680 of *Lecture Notes in Computer Science*, pages 106-119. 2007. [bib | DOI]

HiP-HOPS (Hierarchically-Performed Hazard Origin and Propaga-tion Studies) is a recent technique that partly automates Fault Tree Analysis (FTA) by constructing fault trees from system topologies annotated with component-level failure specifications. HiP-HOPS has hitherto created only classical combinatorial fault trees that fail to capture the often significant temporal ordering of failure events. In this paper, we propose temporal extensions to the fault tree notation that can elevate HiP-HOPS, and potentially other FTA techniques, above the classical combinatorial model of FTA. We develop the formal foundations of a new logic to represent event sequences in fault trees using Priority-AND, Simultaneous-AND, and Priority-OR gates, and present a set of temporal laws to identify logical contradictions and remove redundancies in temporal fault trees. By qualitatively analysing these temporal trees to obtain ordered minimal cut-sets, we show how these extensions to FTA can enhance the safety of dynamic systems.

Keywords: temporal fault trees, formal FTA, automated FTA, fault tree synthesis, formal safety

analysis

[36] M. Walker and Y. Papadopoulos. Qualitative temporal analysis: Towards a full implementation of the fault tree handbook. *Control Engineering Practice*, 17(10):1115-1125, Oct. 2009. [bib | DOI]

The Fault Tree Handbook has become the de facto standard for fault tree analysis (FTA), defining the notation and mathematical foundation of this widely used safety analysis technique. The Handbook recognises that classical combinatorial fault trees employing only Boolean gates cannot capture the potentially critical significance of the temporal ordering of failure events in a system. Although the Handbook proposes two dynamic gates that could remedy this, a Priority-AND and an Exclusive-OR gate, these gates were never accurately defined. This paper proposes extensions to the logical foundation of fault trees that enable use of these dynamic gates in an extended and more powerful FTA. The benefits of this approach are demonstrated on a generic triple-module standby redundant system exhibiting dynamic behaviour.

Keywords: Fault trees, Temporal logic, Safety critical, Safety analysis, Reliability

[37] M. Wallace. Modular architectural representation and analysis of fault propagation and transformation. *Electronic Notes in Theoretical Computer Science*, 141(3):53-71, Dec. 2005. [bib | DOI]

This paper describes a modular representation and compositional analysis of a system's hardware and software components, called Fault Propagation and Transformation Calculus (FPTC). We show, given an architectural description of how components are combined into a whole system, together with an FPTC expression of each component's failure behaviour, how the failure properties of the whole system can be computed automatically from the individual FPTC expressions.

From a safety point of view, this provides some idea of robustness: the system's capability to withstand certain types of failures in individual components. It also provides a way to understand how and where to develop fault accommodation within an architecture.

Keywords: components, architecture, safety-critical, validation

[38] Y. Wei, M. Rodriguez, and C. S. Smidts. Probabilistic risk assessment framework for software propagation analysis of failures. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 224(2):113-135, 2010. [bib | DOI | http]

Probabilistic risk assessment (PRA) is a methodology consisting of techniques to assess the probability of failure or success of a system. It has been proven to be a systematic, logical, and comprehensive methodology for risk assessment. However, the contribution of software to risk has not been well studied. To address this shortcoming, recent research has focused on the development of an approach to systematically integrate software risk contributions into the PRA framework. The latter research has identified as key the need to quantify various major software-failure-related contributions to risk. Of these contributions, the quantification of input failures is the topic of this paper. An input failure consists of a failure of a system component directly or indirectly connected to a software component, which reaches the software input and propagates through the software component. The paper studies and quantifies the impact of input failures on the software component and then further on in the system, and outlines a framework to systematically conduct such an analysis. An application to a safety-critical system is also provided that illustrates the application of the concepts introduced in the paper.

[39] M. Wirsing. Algebraic specification. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 675-788. North-Holland, 1990. [bib]

[40] I. Wolforth, M. Walker, L. Grunske, and Y. Papadopoulos. Generalizable safety annotations for specification of failure patterns. *Software: Practice and Experience*, 40(5):453-483, Apr. 2010. [bib | DOI]

Components in programmable systems often exhibit patterns of failure that are independent of

function or system context. In this paper, we show that it is possible to capture, and reuse where appropriate, such patterns for the purposes of system safety analysis. We describe a language that enables abstract specification of failure behaviour and define the syntax and semantics of this language. The language extends concepts originally defined in HiP-HOPS, a technique that enables a largely automated form of compositional system safety analysis. The paper describes how this language can be used to describe component failure patterns and demonstrates how it can be applied using a simple fuel system example. The approach is evaluated on a set of retrospective industrial case studies, where data-mining and reverse engineering techniques are applied in order to identify hidden patterns in legacy safety analyses. Results show clear potential for practical use of patterns in HiP-HOPS. We argue that careful specification and reuse of failure patterns in conjunction with a tool that automates Fault Tree and Failure Modes and Effects Analysis can help to simplify complex safety assessments

Keywords: compositional safety evaluation, dependability, failure patterns, generalizable safety annotations

[41] I. Wolforth, M. Walker, Y. Papadopoulos, and L. Grunske. Capture and reuse of composable failure patterns. *International Journal of Critical Computer-Based Systems*, 1(1/2/3):128-147, Feb. 2010. [bib | DOI]

Emerging safety analysis techniques use composition of failure models or fault simulation in formal models of a system to determine relationships between the causes and effects of failure. Most recent work has focused on developing system modelling and algorithms for automatic safety analysis. However, little work has focused on developing principles to improve reuse of safety analyses in the context of these techniques. In this paper, we describe a generalised failure logic (GFL) that can capture abstract reusable characteristics of failure behaviour and show how the GFL can be used with templates for the specification of reusable and inheritable component failure patterns. Finally, we illustrate how such patterns can be used with HiP-HOPS, an automated fault tree and FMEA synthesis tool, in order to simplify safety analysis while formalising and improving reuse. Benefits of this approach are discussed in the light of a case study on a brake-by-wire example.

Keywords: safety patterns, reuse in safety analysis, automated FMEA, automated FTA

[42] Y. Yeh. Triple-Triple Redundant 777 Primary Flight Computer. In *IEEE Aerospace Applications Conference*, pages 293-307. IEEE, 1996. [bib]
