REROUTING FAILURE FLOWS USING LOGIC BLOCKS IN FUNCTIONAL MODELS FOR IMPROVED SYSTEM ROBUSTNESS: FAILURE FLOW DECISION FUNCTIONS

Short, Adam R.; Van Bossuyt, Douglas Lee
Colorado School of Mines, United States of America

Abstract

Functional modelling methods used in the early conceptual phases of complex system design allow system designers to better understand and refine system architecture from a functional perspective. A family of methods exist to model functional failures and failure flows. These failure flow modelling methods provide the opportunity to understand potential system failure sources and redesign systems for more robustness. One area lacking from the family of function failure and flow methodological family is the ability to model failure flow decision-making. This paper presents the Function Flow Decision Functions (FFDF) methodology that allows system designers to model failure flow decision-making where critical functions and flow exports are protected from failure flows by sacrificing less critical functions and flow exports. By sacrificing less critical functions and flow exports, mission-critical functions and flow exports can be preserved in order to accomplish the primary mission objectives of a system. A case study based upon the Mars Exploration Rover platform is presented in this paper.

Keywords: Functional Modelling, Failure Modelling, Failure Flow Decision-Making, Design methodology, Robust Design

Contact:
Dr. Douglas Lee Van Bossuyt
Colorado School of Mines
Mechanical Engineering
United States of America
dvanboss@mines.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

In the early phases of complex system design, functional modelling methods are often used to develop system architecture and functionality. Function failure analysis and design methods such as Function Failure Identification and Propagation (**FFIP**) (Kurtoglu & Tumer 2008) and others have provided methods for system designers to model failures in the conceptual phase of design and change the design of a system to be more robust. However no method of assessing the diversion of failures from critical flow paths and functions to less critical paths and functions currently exists for the early phases of conceptual design. This paper contributes a novel method of representing failure flow path decision-making in complex systems. The Failure Flow Decision Functions (**FFDF**) methodology presented in this paper allows system designers to model diverting failure flows to less critical functions and failure flow export paths thus preserving critical flow paths and functionality for critical mission success at the cost of less critical portions of a system's mission.

1.1 Specific Contributions

The focus of this paper is on a novel method for analysing a system to determine the optimal failure flow path through the system as a whole. FFDF builds upon the existing method of FFIP in order to create a method for determining the optimized failure path. The method can increase the reliability of a system by directing failures away from critical system functions and flows. FFDF contributes to the field by providing a replicable and standardisable method for determining values the effects of sub-system failure on a system. The results of this analysis can be used to inform decision making in the design phase as well as provide important information for managing and controlling a system operating in hazardous conditions.

2 BACKGROUND

The method presented in this paper relies upon several related fields of research and practice including system modelling and failure analysis methods. Of specific interest to this work are the sub topics of functional modelling methods and function failure analysis methods. This section outlines important background material that aids in understanding the methodology.

2.1 System Modelling

System modelling is a powerful tool for developing models of complex systems that can be used in a wide range of analysis and design applications. Functional Flow Block Diagrams (**FFBD**) are used in a variety of systems engineering applications (Blanchard et al. 1990). Integrated Computer Aided Manufacturing (**ICAM)** DEFinition for Function Modelling (**IDEF0**) is another system modelling language developed for use in systems engineering applications originally with special focus on aerospace and defence systems (Ross et al. 1981). Systems Modelling Language (**SysML**) (Friedenthal et al. 2011), developed as a more flexible systems engineering-specific form of the Unified Modelling Language (Rumbaugh et al. 2004) used in software design, was first introduced in 2001 and has had several updates and standardizations (Obj 2012). SysML is useful for complex engineered systems and has been advocated by the International Council on Systems Engineering (**INCOSE**) (Obj 2003). The Functional Basis for Engineering Design (**FBED**), described by Stone and his collaborators (Bryant et al. 2005, Hirtz et al. 2002, Kurtoglu et al. 2005, Stone & Wood 2000, Stone et al. 2000), provides concise definitions of functions and flows that describe all possible engineered systems. Stone's FBED is used in this research due to the significant development of advanced failure analysis methods that are built upon FBED (Kurtoglu & Tumer 2008, Kurtoglu et al. 2010, Lough et al. 2009, O'Halloran et al. 2015, Ramp & Van Bossuyt 2014, Stone, Tumer & Stock 2005, Stone, Tumer & Van Wie 2005, Tumer et al. 2003).

2.2 Failure Analysis Methods

Several failure analysis methods have found widespread use in various industries. Failure Modes and Effects Analysis (**FMEA**) has been used extensively since its development for military applications in the 1950s (Stamanis 2003). Reliability Block Diagrams (**RBD**) are used to determine system reliability using parallel and series flow paths through blocks containing reliability data (Modarres et al. 1999). Probabilistic Risk Assessment (**PRA**) was developed out of several industries including

aerospace, defence, and civilian nuclear power, and sees significant continued use in those industries (Keller & Modarres 2005, Modarres 2008).

Recent developments in academia of relevance to this research focus on function failure methods derived from Stone's FBED methodology. The Function Failure Design Method (**FFDM**) was developed as a method of connecting FMEA with FBED. FFDM allows designers to choose specific functions and their component solutions based upon historical component failure data (Stone, Tumer & Van Wie 2005). The Function-based Analysis of Critical Events (**FACE**) methodology provided a method of modifying functional models based upon critical events during a complex system's lifecycle, such as with the various stages of a spacecraft mission (launch, deployment, cruise, orbital capture, descent and landing, primary science mission, etc.) (Hutcheson et al. 2006). The Function Failure Identification and Propagation (**FFIP**) method models failure flows through FBED models (Kurtoglu & Tumer 2008). The Function Failure Reasoning (**FFR**) method developed a simulation tool to model FFIP across a complex system (Kurtoglu et al. 2010). Flow State Logic (**FSL**) was developed to further refine the FFIP method and provide for a complete representation of the analysed system's state (Jensen et al. 2009). The Uncoupled Failure Flow State Reasoner (**UFFSR**) method was developed to address shortcomings in FFIP related to failure flows that do not follow nominal flow pathways (O'Halloran et al. 2015, Ramp & Van Bossuyt 2014). While function failure methods have seen significant development over the past ten years, many areas remain to be addressed and new areas for potential innovation exist within the current methods and frameworks. This paper develops one such innovation which is described in the following sections.

3 METHODOLOGY

The core idea of the Failure Flow Decision Functions (**FFDF**) methodology is that failure flows can be directed away from critical functions and to less-critical functions in order to preserve core critical functionality while sacrificing some less critical functionality of the system. A simple physical analogy of the core idea is the routing of floodwaters along a sparsely-populated irrigation canal rather than a densely-populated urban area. A functional modelling example is an electrical energy failure flow that is diverted from a critical process signal function to a less critical convert electrical to rotational function. In this case, the critical function would be protected from the electrical energy failure flow and the less critical function would fail due to the failure flow being directed there.

The methodology presented in this paper is specifically useful for engineered complex systems that are existent or for systems that are under development where high reliability of core functionality is desired. Systems that are modelled as largely serial functional relationships will not benefit from this method. Complex systems with many parallel flow paths and diverse functions benefit the most from the implementation of this methodology. The implementation of the methodology requires several steps that are outlined as follows:

1. Using existing FBED methodologies, develop a functional model of the complex system of interest.
2. Implement FFIP methodology (Kurtoglu et al. 2010) to determine failure flow paths and failure probabilities.[1]
3. Determine critical export flows from the system and associated critical functions.
4. Determine hierarchy of export flows and functions based upon criticality to overall system mission success.
5. Insert FFDF functions at points where the system designer desires the ability to choose where a failure flow will be directed. A FFDF is a new addition to the FBED and FFIP family of function flow failure methodologies. The FFDF represents the ability of a system to choose the flow path along which a failure flow will be sent. While a series of functions and flows representing physical components including sensors, automated decision tools such as embedded computing

[1] Note that to avoid statistics problems, we advocate using a Monte Carlo method to determine failure flow path probabilities in the FFIP method. The methodology presented in this paper uses this approach with good success.

systems, and flow direction equipment could be represented using the FBED functional basis, for clarity we represent this series of functions and flows as a single pseudo-function (the FFDF) and with an ovular shape rather than a rectangle, as is traditional with functional models. The FFDF incorporates logic that represents the critical functions and the hierarchy of functions based upon importance to the critical export flows that contribute to overall mission success.

6. Use the FSL and FFR simulation methods augmented with FFDF from Step 5 to determine new failure flow paths and probabilities.
7. Adjust FFDF placement and logic as needed to reduce the probability that a critical flow export will deliver a failure flow. This can be done either by hand or iteratively and automatically using simulation software.

By following the above steps and implementing FFDFs, a system designer can rapidly determine where the system will benefit from directing failure flows to less important functions. This will result in higher critical export flow reliability and higher overall mission success probability. The next section presents a case study based upon a simplified Mars rover.

4 CASE STUDY

In order to illustrate the utility of the FFDF, we present a simplified Mars rover case study based loosely upon the National Aeronautics and Space Administration (**NASA**) Mars Exploration Rover class (Figure 1) exemplified by the Opportunity and Spirit rovers that are currently located on Mars. Figure 2 presents a highly simplified functional diagram of the power systems aboard the generic rover model. Included are the energy production system (Accumulate Energy), the energy delivery system (Deliver Energy), the computer system (Process Signal), a generic vision system (Record Visual), motor controllers (Control Magnitude Electrical), motors and wheels (Convert Electrical to Rotational, and Convert Rotational to Translational), and computer processing of information (visual, rotation counter, etc.) to determine distance driven by the rover (Record Position).

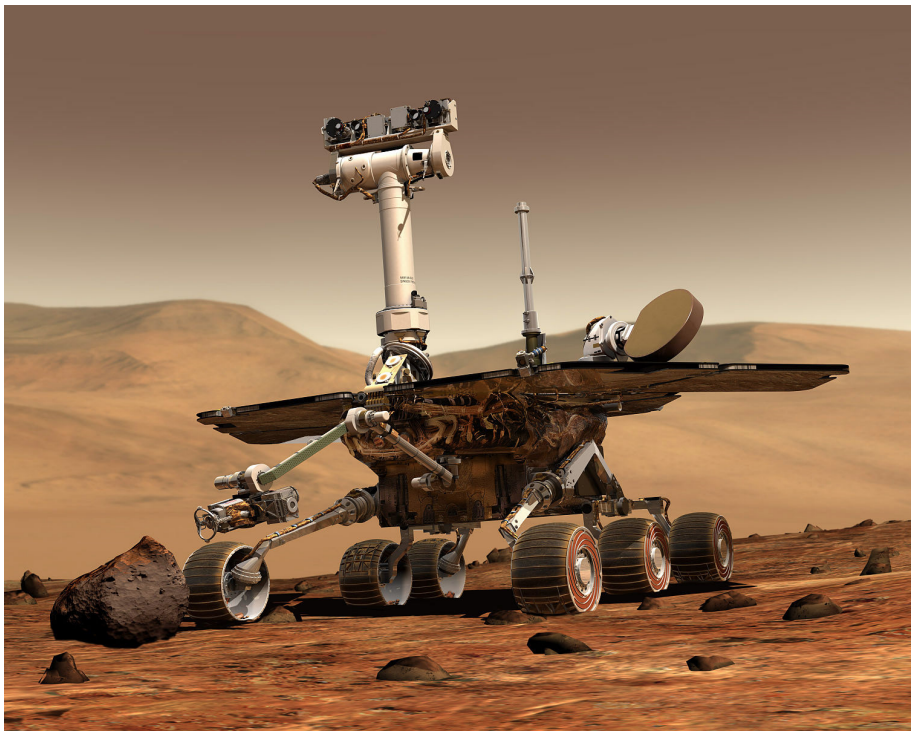

Figure 1. An artist's rendering of a Mars Exploration Rover situated on typical Martial terrain.

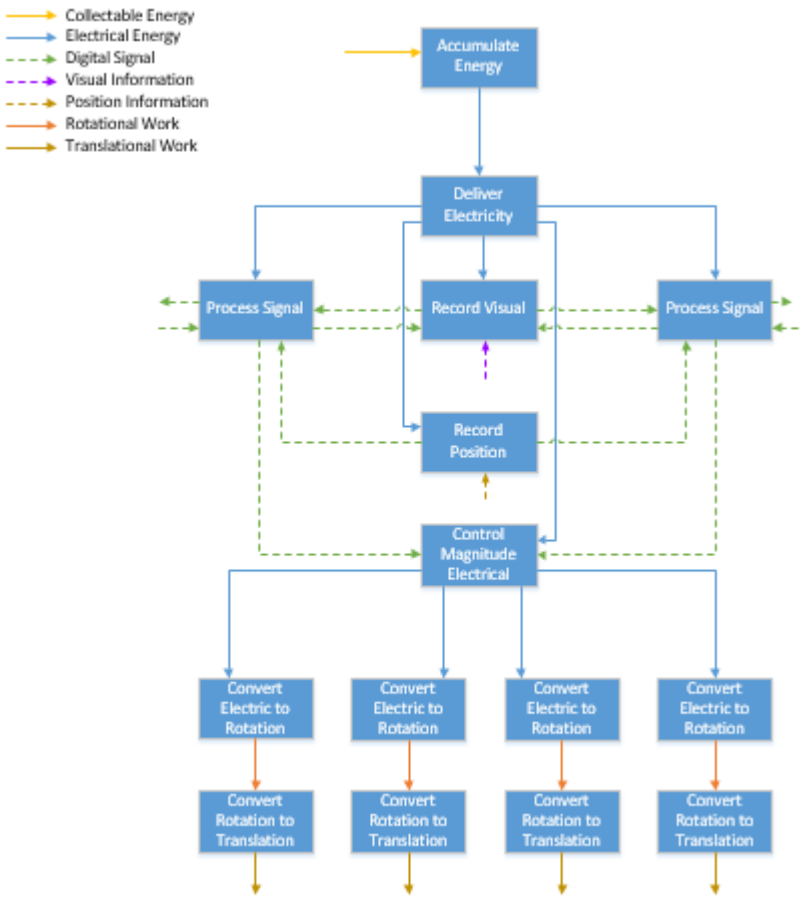

Figure 2. Functional model of simplified rover system.

In this case study, we assume that a major cause of failure on the rover system is an Electrical Energy failure flow that takes the form of too much energy being available to the system. An example of this failure mode is a situation where excess energy is generated from the rover power source and must be consumed by the powered systems of the rover. In this case study, we assume systems such as electric heaters and coolers, instruments that can be turned on and off, batteries, and other potential excess energy sinks are not available. The simplified rover system only has the previously identified functions (shown in Figure 1) available to accept the Electrical Energy failure flow.

As per the FFDF methodology, a FFIP implementation was developed of the simplified rover. The critical export flows were then identified which include in rank order: 1) Export Visual, representing sending data back to Earth for scientific study and analysis, and 2) Export Translation, representing mobility of the rover platform in order to drive from one scientific target to another. It should be noted that only two of the four Export Translation export flows are critical and further that one Export Translation flow from each physical side of the rover is required to allow full mission success. Partial mission success can be achieved with only Export Visual. An example of this partial mission success behaviour comes from the NASA Spirit Rover's quagmire situation where the rover became stuck in a sandy area after several wheel and motor pairs were disabled[2].

A FFDF was manually inserted in between the Deliver Energy function and subsequent functions, as shown in Figure 3. This could be physically represented by a current-limiting device with intelligent switching capabilities. The Electrical Energy failure flow could be successfully channelled to one of

[2] Note that the Spirit Rover did succeed at its primary and extended mission objectives. It was only several years after the rover was expected to fail that the rover became stuck in the sand and eventually lost contact with earth (Munroe 2010).

the less critical functions, thus protecting the critical functions from the failure flow. Physically, this could manifest as the decision to sacrifice one electric wheel motor in order to save the critical portions of the rover related to collecting scientific data and overall mobility of the rover. In the simplified functional model, this is represented by channelling the Electrical Energy failure flow to either one or two out of the four Convert Electrical to Rotational functions. It should be noted that the simplified case study assumes that the Control Magnitude Electrical function will pass through the Electrical Energy failure flow to a specific Convert Electrical to Rotational Function from the FFDF.

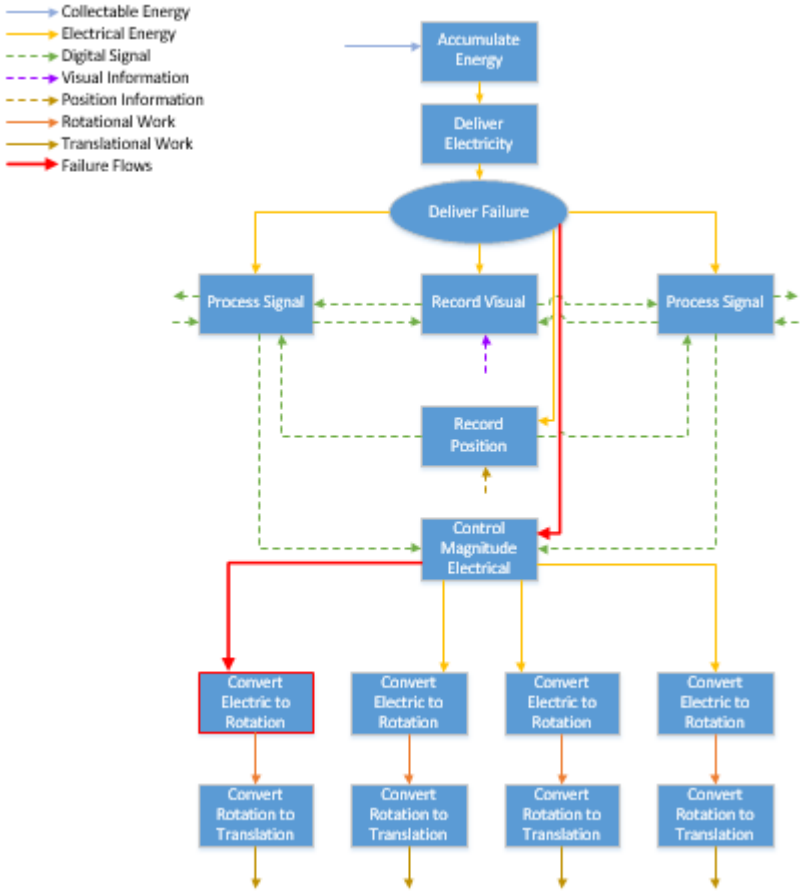

Figure 3. FFDF implementation in simplified rover system with red line indicating intentional failure flow direction from FFDF logic. In this case, the Convert Electric to Rotation function was sacrificed in order to preserve critical functions and critical export flows.

A simulation of the FFDF was created in MATLAB and run using a Monte Carlos sampling method to determine the likelihood of failure. The simulation looks at the initiating event and then calculates the probability of the failure propagating to connected sub-systems using a method that enhances traditional FFIP analysis. The simulation then examines failure propagation from the new sub-systems until all possible paths have been explored.

Using the FSL and FFR simulation methods augmented with FFDF, probabilities of failure of the two critical export flows was determined. Results and discussion of the FFDF implementation are provided in the next section.

5 RESULTS AND DISCUSSION

The FFDF implementation resulted in a significant reduction of critical export flow failure. The FFIP analysis produced an 85% probability of critical export flow failure with an initiating event of excess electrical energy being introduced to the Accumulate Energy function. Using FFDF and directing the failure flow to one of the Convert Electrical to Rotational functions, the critical export flow failure probability is reduced to 13%. Table 1 shows the probability of critical export flow failure for a variety of potential logic choices within the FFDF.

Table 1. Results of FFIP and FFDF methods.

Failure Path	Probability of Critical Export Flow Failure (%)
FFIP base case	85
Failure flow directed to Process Signal Function	30
Failure flow directed to Record Visual Function	100
Failure flow directed to Convert Electrical to Rotational Function	13
Failure flow directed to Record Position Function	26

Analysing the potential failure flow pathways shows that the best place for the FFDF to send the Electrical Energy failure flow is to one of the Convert Electrical to Rotational functions. The Convert Electrical to Rotational function then fails, thus failing its individual Export Translation flow but not failing enough of the Export Translation flows to cause the critical Export Translation flow (defined as two out of the four Export Translation flows) to fail. Conversely, sending the Electrical Energy failure flow to the Record Visual function leads to 100% failure of the critical export flow of the Export Signal flow. This is because the most critical mission of the simplified rover model is to produce scientific data, as represented by a generic visual sensor that has data processed by the redundant on-board computers and returned to earth via a radio signal represented by the Record Visual function, the Process Signal functions, and the Export Signal flow. Other potential flow paths that the FFDF could be directed to send the failure flow to have intermediate levels of probability of critical export flow failure. Based upon this information, A logic table for the physical implementation of the FFDF can be built where the failure flow is directed first to the Convert Electrical to Rotational function; second to the Record Position function; third to the Process Signal function; and finally if no other options are available, fourth to the Record Visual function.

In a more complexly-modelled system such as a nuclear power plant or a petroleum refinery, many redundant flow paths exist for reasons of safety through redundancy or online maintenance. The FFDF methodology allows for better function failure modelling of these and other complex systems while also locating potential new methods of protecting critical flow exports and functions from failure flows. It is expected that as the FFDF methodology is further refined and fully automated that new and innovative ways of protecting critical flow exports and functions will be discovered that were previously not available to the functional modelling community.

6 CONCLUSION AND FUTURE WORK

The FFDF method presented in this paper is a novel method of inserting logic blocks into function failure models with the goal of channelling failure flows away from critical flow exports and functions, and toward less important functions and flow exports that can be sacrificed in order to preserve core functionality. There previously was no straightforward way to model failure flow decision making in functional modelling approaches to failure analysis and design. With FFDF, designers can actively choose how failure flows will propagate through a system to prevent critical flow exports and critical functions from being compromised by failure flows. Further development of the FFDF method includes expanding the logic available to the FFDF blocks, fully automating the FFDF methodology to allow for automated design optimization and analysis, and investigate differentiating between different failure flow magnitudes in order to better route failure flows toward functions that can withstand failure flows of various magnitudes.

REFERENCES

Blanchard et al. 1990Blanchard, B. S., Fabrycky, W. J. & Fabrycky, W. J. (1990), Systems engineering and analysis, Vol. 4, Prentice Hall Englewood Cliffs, New Jersey.

Bryant et al. 2005Bryant, C. R., Stone, R. B., McAdams, D. A., Kurtoglu, T. & Campbell, M. I. (2005), Concept generation from the functional basis of design, in 'ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy', Engineers Australia, p. 1702.

Friedenthal et al. 2011Friedenthal, S., Moore, A. & Steiner, R. (2011), A practical guide to SysML: the systems modeling language, Elsevier.
Hirtz et al. 2002Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S. & Wood, K. L. (2002), 'A functional basis for engineering design: reconciling and evolving previous efforts', Research in engineering Design 13(2), 65–82.
Hutcheson et al. 2006Hutcheson, R. S., McAdams, D. A., Stone, R. B. & Tumer, I. Y. (2006), A function-based methodology for analyzing critical events, in 'Proceedings of the IDETC/CIE'.
Jensen et al. 2009Jensen, D., Tumer, I. Y. & Kurtoglu, T. (2009), Flow state logic (fsl) for analysis of failure propagation in early design, in 'ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference', American Society of Mechanical Engineers, pp. 1033–1043.
Keller & Modarres 2005Keller, W. & Modarres, M. (2005), 'A historical overview of probabilistic risk assessment development and its use in the nuclear power industry: a tribute to the late professor norman carl rasmussen', Reliability Engineering & System Safety 89(3), 271 – 285. http://www.sciencedirect.com/science/article/pii/S0951832004002327
Kurtoglu et al. 2005Kurtoglu, T., Campbell, M. I., Bryant, C. R., Stone, R. B. & McAdams, D. A. (2005), Deriving a component basis for computational functional synthesis, in 'ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy', Engineers Australia, p. 1687.
Kurtoglu & Tumer 2008Kurtoglu, T. & Tumer, I. Y. (2008), 'A graph-based fault identification and propagation framework for functional design of complex systems', Journal of Mechanical Design 130(5), 051401.
Kurtoglu et al. 2010Kurtoglu, T., Tumer, I. Y. & Jensen, D. C. (2010), 'A functional failure reasoning methodology for evaluation of conceptual system architectures', Research in Engineering Design 21(4), 209–234.
Lough et al. 2009Lough, K. G., Stone, R. & Tumer, I. Y. (2009), 'The risk in early design method', Journal of Engineering Design 20(2), 155–173.
Modarres 2008Modarres, M. (2008), Probabilistic risk assessment, in 'Handbook of Performability Engineering', Springer, pp. 699–718.
Modarres et al. 1999Modarres, M., Kaminskiy, M. & Krivtsov, V. (1999), Reliability engineering and risk analysis: a practical guide, CRC press.
Munroe 2010Munroe, R. (2010), 'Spirit', XKCD Web Comic. http://xkcd.com/695/
Obj 2003Obj (2003), UML for Systems Engineers Request for Proposal.
Obj 2012Obj (2012), OMG Systems Modeling Language.
O'Halloran et al. 2015O'Halloran, B., Papakonstantinou, N. & Van Bossuyt, D. L. (2015), Modeling of function failure propagation across uncoupled systems, in 'Proceedings of the Reliability and Maintainability Symposium'.
Ramp & Van Bossuyt 2014Ramp, I. J. & Van Bossuyt, D. L. (2014), Toward an automated model-based geometric method of representing function failure propagation across uncoupled systems, in 'Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition IMECE2010', ASME, Montreal, QC.
Ross et al. 1981Ross, D. T., Hori, S., Fieldmann, C. G., Thornhill, D. E., Bravoco, R. R., Connor, M. F., Marca, D. A. & Cornfield, J. (1981), Integrated computer-aided manufacturing (icam) architecture part ii volume iv - function modeling manual (idef0), Technical report, SofTech Inc, Waltham, MA.
Rumbaugh et al. 2004Rumbaugh, J., Jacobson, I. & Booch, G. (2004), Unified Modeling Language Reference Manual, The, Pearson Higher Education.
Stamanis 2003Stamanis, D. H. (2003), Failure Modes and Effects Analysis: FMEA from Theory to Execution, 2nd edn, ASQ Quality Press, Milwaukee, WI.
Stone, Tumer & Stock 2005Stone, R. B., Tumer, I. Y. & Stock, M. E. (2005), 'Linking product functionality to historic failures to improve failure analysis in design', Research in Engineering Design 16(1-2), 96–108.
Stone, Tumer & Van Wie 2005Stone, R. B., Tumer, I. Y. & Van Wie, M. (2005), 'The function-failure design method', Journal of Mechanical Design 127(3), 397–407.
Stone & Wood 2000Stone, R. B. & Wood, K. L. (2000), 'Development of a functional basis for design', Journal of Mechanical Design 122(4), 359–370.
Stone et al. 2000Stone, R. B., Wood, K. L. & Crawford, R. H. (2000), 'Using quantitative functional models to develop product architectures', Design Studies 21(3), 239–260.
Tumer et al. 2003Tumer, I. Y., Stone, R. B., Bell, D. G. et al. (2003), Requirements for a failure mode taxonomy for use in conceptual design, in 'DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm'.

ACKNOWLEDGMENTS

This research was partially supported by United States Nuclear Regulatory Commission Grant Number NRC-HQ-84-14-G-0047. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators.