ATLANTIS PRESS

Advances in Computer Science Research, volume 59

7th International Conference on Education, Management, Computer and Medicine (EMCM 2016)

Overview of System Reliability Modeling Tools

Jiacong Zhao[1, a *], Yanxia Lu[1, b], Qisong Zhang and Xiaoyan Zhang[1, c]

[1]Dalian Neusoft University of Information, Dalian, China

[a]zhaojiacong@neusoft.edu.cn, [b]luyanxia@neusoft.edu.cn, [c]zhangqisong@neusoft.edu.cn, zhangxiaoyan@neusoft.edu.cn

Keywords: Reliability; Dynamic fault tree; Dynamic reliability block diagrams; Dynamic behavior; Colored petri nets

Abstract. System reliability is a critical aspect of a system, which incurs the proposing of related modeling tools. Tools like Reliability Block Diagrams (RBD) and Fault Tree (FT) provide static representation of system reliability. RBD is a graphical representation that depicts a network of system components and connections. FT is a logical and diagrammatic method to evaluate an accident's probability that results from faults and failure events. The increasing complexity of systems drives the demands for analyzing system dynamic behaviors like dynamics, dependencies, redundancy and load sharing. Dynamic models like Dynamic Reliability Block Diagrams (DRBD) and Dynamic Fault Tree (DFT) are proposed. Dynamic tools define frameworks for modeling dynamic reliability behavior of systems. DRBD is an extending and enhancing of RBD by adding a state-events working mechanism, which permits to model dynamic reliability behaviors of the system. DFT is an extension of FT by adding the sequential notion which can meet time requirements. It is also important to verify these tools for locating and identifying deadlock and faulty states. The efficiency of verification techniques to model these modeling tools and the assessment of dynamic behaviors are taken as evaluation criteria. Based on current research, DRBD performs better than alternatives.

Introduction

Currently, there is a significantly increasing reliance on computer-based systems, especially systems for controlling critical infrastructures like banking systems. It leads to the system reliability summarized under the concept of dependability is receiving increasing attention. However, there is lack of suitable tools to analyze reliability critically. This fact motivates the construction of reliability modeling tools like Reliability Block Diagrams (RBD) [1,2], Fault Tree (FT) [1,2]. However, since the stochastic independence assumption stands, these static-based modeling tools offer no capabilities to model dynamic aspects like reliability interactions among components or subsystems [3], which means that static formalisms have restrictions in achieving dynamic properties like dynamics, dependencies, redundancy and load sharing [4]. Then dynamic modeling tools were proposed like Dynamic FT (DFT) and Dynamic RBD (DRBD). DFT can model a functional dependency in a system, where a component's failure will cause the inaccessibility or un-usability of other dependent components [5]. Although DFT works well in modeling dynamic reliability, it cannot adequately represent dynamic behaviors previously listed and model general state-based dependencies relationship between components [6]. DRBD which is an extending and enhancing of RBD is also proposed. DRBD consists of a state-based RBD (SRBD) and controller blocks that support modeling dynamic relationships between components in a computer-based system. Although DRBD models dynamic reliability properties effectively, it introduces subtle flaws easily due to its modeling complexity [6]. In order to deal with the drawbacks of these modeling tools, verification methodologies are introduced to avoid the certain limitation of each tool. Therefore, the assessment of dynamic behaviors and efficiency of verification methodologies become important criteria for evaluating modeling tools. Based on these criteria, this report takes the position that compared with mentioned alternatives of this paper, DRBD is the best tool to model system reliability. The rest of

this review gives a brief introduction to reliability modeling tools, evaluates and compares these tools and finally draws a conclusion.

System Reliability Modeling Tools

Generally, reliability modeling tools is divided into static-stage and dynamic-stage.

Static Tools. Static tools regard system components' state as either active or failed. The typical examples are RBD and FT. RBD is a graphical representation that depicts a network of system components and connections which can use the given reliability of each component to determine the overall system reliability [8, 9]. Generally, series and parallel are two main types of connections, they can be established between more than one components [6,11]. Fig.1 is a simple hybrid RBD model, according to each component's reliability, the overall system reliability is calculated as 88.4%. The combinations of working components that keep the entire system operational are represented by blocks, a component's failure represented by removing the component and its connections with other components from the network [6, 7, 11]. In addition, the network has an input and output point that are linked by multiple paths which represent successful system operations. Therefore, if there are enough working components guaranteeing only one linking path between the input and output points the system is still under the working condition. The main advantage of RBD is that it can be modeled and read easily [7]. This virtue enables system design and test engineers and managers who make decisions on system configuration work more effectively. For example, with RBD model design engineers can understand the system productively and comprehensively, then they can easily construct and modify the corresponding RBD model and exchange ideas with people from different disciplines. However, RBD only can represent the static topology of system's reliability and be applied to non-repairable system configuration and behaviors. It cannot model the dependencies and dynamic behaviors of systems, especially for analyzing the computer-based and complex systems.

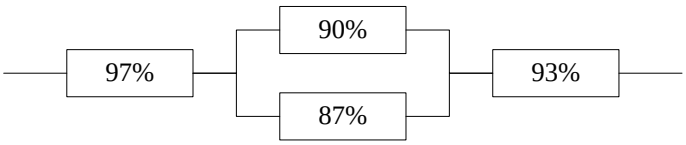

Figure 1. RBD

FT is a logical and diagrammatic method to evaluate an accident's probability that results from faults and failure events [14], which means that it provides a compact, graphical and intuitive method to analyze system reliability [13]. Compared with RBD, the working mechanism of FT is simpler. FT starts with the top event and then deduces all possible ways for this event systematically. Specifically, the model is based on three assumptions and two analysis steps. The assumptions describe as events are binary ones, they are statically independent and their relationship are depicted by AND, OR, and Voting gates (Fig.2) [12]. As to the steps, one is qualitative step for expressing the top event and the other is quantitative step for calculating the occurrence of top event [6, 12]. Therefore, FT is widely used for the quantitative reliability and detailed safety analysis. However, this mechanism indicates that failure rates and probabilities of individual component are difficult to estimate. Additionally, it cannot manage time-variant features. For example, the top event cannot be monitored if it acts as a function of time to follow the changes of the system.

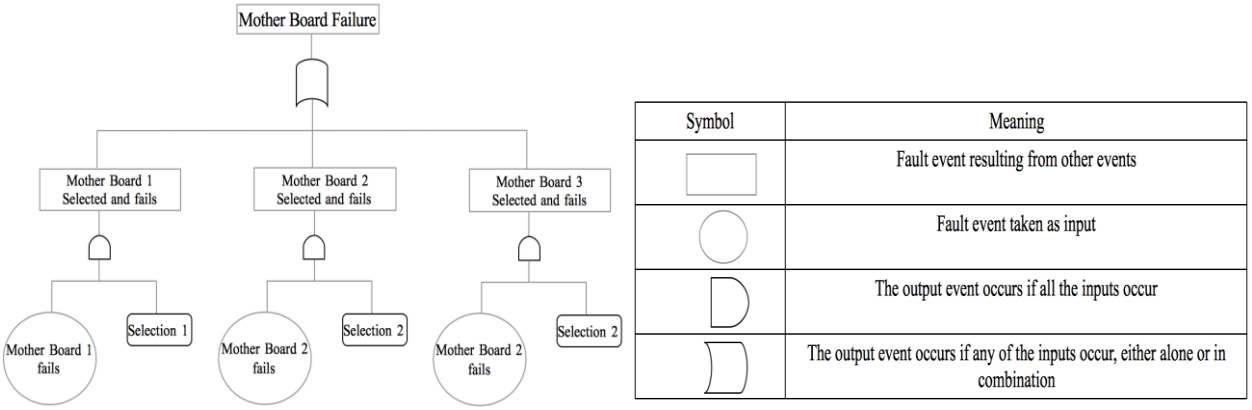

Figure 2. FT Structure

Dynamic Tools. DRBD is an extending and enhancing of RBD by adding a state-events working mechanism, which permits to model dynamic reliability behaviors of the system [4,11]. Briefly, a DRBD model consists of SRBD and controller blocks [6]. The SRBD not only inherits advantages of RBD, but extends RBD by associating with a state representing the activeness of system's components. Additionally, controller blocks are for dynamic behaviors modeling. Specifically, in DRBD the topology and/or configuration of the target system is considered as time-variant. Then, each component of DRBD is characterized by a variable state to identify its operational condition during a certain period and the evolution of these states is characterized by events [11]. Fig.3 [4] summarizes all the possible states (rounded rectangles), events (directed arcs) and their relationships. Each state and event has its specific meaning. As to states, Active means the component works without problem, Failed represents the component reaches this state following its failure, Standby describes an unworkable component [4,6,11]. Additionally, Standby is divided into Hot-standby, Warm-standby and Cold-standby with the meaning of energized, partially energized and not energized by order. Events demonstrate the transition of a component from a state to another. For example, Sleep represents the transition from Active to a Standby. Therefore, the implementation of DRBD is based on two main points: characterizing components by system dynamics and specifying dependency as the dynamic reliability modeling's building block [11]. The main enhancement of DRBD is to model subsystems' dependencies. For example, with DRBD modelers can easily schematize the system modeling approach into ordered steps: system specification, subsystems identification, structural linking, dynamic linking and reiteration, which detail all components of subsystems. However, the modeling process is complex and introduces subtle flaws easily [6].

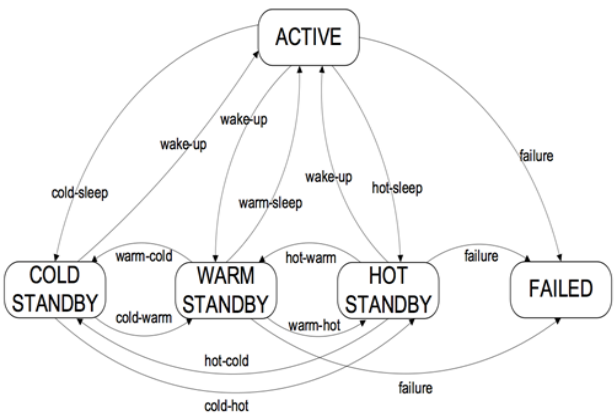

Figure 3. DRBD

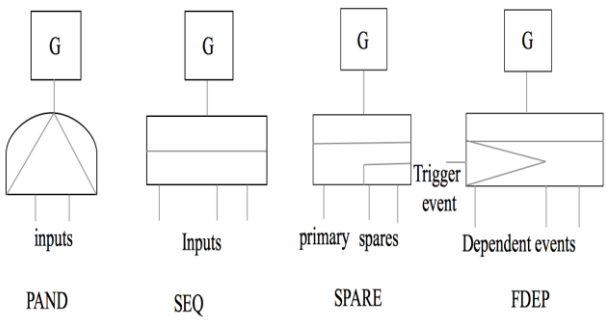

Figure 4. Dynamic Gates of DFT

DFT is an extension of FT with time by adding the sequential notion which can meet time requirements [13, 15, 16]. Then the system failures depend on component failure order and combination [16]. This can be achieved by introducing dynamic gates: PAND, SEQ, SPARE and FDEP to FT. Specifically, according to Fig.4: PAND will reach the failure state after all the input

components have failed in a preassigned order, graphically from left to right. SEQ happens that it inputs to fail in a certain order as the preassigned. If the number of operational powered spares and/or principle components that with the same functionality is less than the minimum required, SPARE gate will fail [17]. The FDEP is consisted by one trigger-input and one or more dependent events, the occurrence of trigger event forces the occurrence of dependent events [13]. Hence, these special gates can model: dynamic replacement of failed components from pools of spares, failures that occur only if others occur in certain orders, dependencies that propagate failure in one component to others, and situations where failures can occur only in a predefined order [13]. Therefore, engineers can control configuration and integrate several modes of equipment operation. However, DFT also shows limitation in modeling systems that involve general state-based dependencies between components. Furthermore, the problems caused by concurrency among dependencies cannot be managed as well.

Evaluation and Comparison

In order to use modeling tools to precisely analyze system reliability, this report chooses efficiency of verification methodologies and assessment of dynamic behavior as the evaluation and comparison criteria. Verification methodologies means to avoid the drawbacks of chosen tool by mapping it onto an analyzable domain. Dynamic behavior specifies to dynamics, dependencies, redundancy and load sharing.

Efficiency of Verification Methodologies. Exploiting Markov Chain (MC) and Petri Nets (PN) are widely used to model RBD and FT [18]. Additionally, the methodologies to analyze these two tools are significantly similar due to their equivalent mathematical characteristic. However, methodologies are proposed to model dynamic tools for their higher functionality. Hence, verification methodologies of DFT and DRBD are discussed, especially automatically modeling DRBD by CPN.

There are several proposed methodologies modeling DFT. Gulati proposed the modular approach which provides a combination of Bryant's Binary Decision Diagram (BDD) for FT and MC for dynamic features coupled with the detection of independent subtrees [19]. Similarly, Boudali used Input/Output Interactive MC to model the DFT, instead of BDD and MC [17]. Dugan pursued Galileo which can analyze DFT model economically by decomposing complex model into small pieces, applying different techniques to sub-models and finally integrating results into a system-level result [13]. Ameri proposed a numerical integration technique for solving the dynamic gates [16]. Bobbio proposed the Bayesian to further reduce the problem of solving DFT [20]. It shows that current methodologies are dividing DFT into sub-modules and analyzing each of them to fill the gap of DFT's unavailability to manage concurrency among dependencies. Specifically, excluding Dugan divided the whole model into pieces, others were trying to analyze the static modules and dynamic-modules separately. Although these methodologies' results are matching with the analytical approaches, what all of them need is manual work which introduces errors easily. Hence, automatic ways to model DFT are needed.

Methodologies to model DRBD like MC and PN [4, 21] introduce subtle flaws easily due to DRBD's complex modeling process. Furthermore, DRBD contains limited static modeling constructs, which probably lead modelers to bring design errors into the model due to the probability of introducing new modeling constructs [7]. What was proposed to model DRBD was CPN due to their similar working mechanism. Specifically, CPN combines by PN and Standard ML [22]. PN can model concurrency communications and synchronization [23]. Standard ML provides the primitives for dealing with data and creating compact and parameter models. Therefore, a CPN model can describe states and events of time-variant systems. However, an interface should be introduced to convert DRBD to CPN. Corresponding to Standard ML, an XML-based Reliability ML is proposed [6]. The main virtue of RML is to mutate DRBD model as an XML documents which support a standard information encoding and allow programmers to use that information in standard way [6, 24]. Hence, RML can nest all DRBD's components and controllers by its elements to describe their properties according to their respected definitions. The conversion procedure of DRBD to CPN is summarized as a four-step process:

Using RML defines SRBD, then adding controllers into the RML file using specific XML tags
Converting DRBD's SRBD into a CPN model
Converting DRBD's controller blocks into PN
Add the converted PN into the converted CPN

As the development of dynamic tools, the methodologies provided to static ones are extremely limited. In terms of dynamic models, compared with the complex and manual approaches to model DFT, CPN can model DRBD automatically and precisely, which is relatively better.

Assessment of Dynamic Behaviors. RBD and FT cannot achieve the dynamic and dependency features. Additionally, generally no specific redundancy aspects and representation possibilities are offered in them, which demonstrate that they cannot achieve redundancy and load sharing. Compared with static techniques, DFT and DRBD can access dynamic reliability behaviors. The following are the evaluation and comparison of their dynamic features:

Dynamics: The main point for their driving from the static models is to achieve the dynamic reliability behaviors. Hence, they can successfully describe dynamic features.

Dependencies: DFT has no management of problems due to the concurrency among dependencies. Therefore, it cannot fully achieve the dependency feature. However, DRBD provides a compositional model mechanism and a concurrency manager to represent dependencies [4].

Redundancy: Redundancy enables components duplication during system design. DFT introduces a redundancy gate (REB) to model the specified duplications [25]. While DRBD applies a multiple component, which is a compact alternative to represent some specific implementations of multi-units redundant components. Hence, exploiting and combining dependencies among redundant.

Load Sharing: DFT provides a set of gates to achieve loading sharing. For example, FDEP gates achieve load sharing by propagating the failure of one unit to others. Initially, DRBD defines that the load is shared between any two components in Warm-standby as effect of the application of the reciprocal dependency [4], when one of them fails the whole load goes to the other that will be fully activated.

Therefore, compared with dynamic tools, static tools are with significantly limitations. What the comparison result of DRBD and DFT shows is that excluding dependencies, DFT performs dynamic behaviors as well as DRBD.

Conclusion

There is a growing demand to build reliable and stable computer systems. Building these kinds of system, a precise and correct reliability modeling tool should be created. There are a number of proposed tools for analyzing system reliability. This paper firstly discusses static tools like RBD and FT and then turns to dynamic tools like DFT and DRBD. Although these dynamic tools enhance the function of static ones, they have certain disadvantages as well. For example, DFT cannot adequately achieve system's dependencies and DRBD can bring subtle flaws easily due to its analysis complexity. Then, existing techniques like MC, PN and CPN are used to verify the behavioral properties of these modeling tools to avoid their drawbacks. This report shows that, compared with other three mentioned alternatives, DRBD is the most productive models. In addition, CPN automatically identify design flaws and faulty states of DRBD by tracing the deadlock states of the converted CPN model from DRBD. With this automatic way, the system reliability can be modeled correctly and effectively. The future work should focus on developing an environment for supporting editing, verification analysis of dynamic models for large and complex computer-based systems.

References

[1] M. Rausand, A. Høyland, System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Edition, New York, USA, Wiley Interscience, 2003. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] B. W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Boston, USA, Addison-Wesley Longman Publishing Co. Inc., 1988.

[3] S. Distefano and A. Puliafito, Dependability Evaluation with Dynamic Reliability Block Diagrams and Dynamic Fault Trees, IEEE Trans. Dependable Secur. Comput., vol. 6(2009) No. 1, pp. 4–17.

[4] S. Distefano and L. Xing. A new approach to modeling the system reliability: dynamic reliability block diagrams. In RAMS'06 proceedings, 2006, pp.189-195.

[5] R. Manian, J. Dugan, D. Coppit, and K. Sullivan, "Combining various solution techniques for dynamic fault tree analysis of computer systems," in Proc. 3rd Int. Symp. High-Assurance Systems Engineering (HASE'98), Washington, D.C., USA, 1998, pp. 21–28.

[6] R. Robidoux, H. P. Xu, L. D. Xing, and M. C. Zhou, "Automated modeling of dynamic reliability block diagrams using colored Petri nets," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 40(2010)No. 2, pp. 337–351.

[7] H. Xu, L. Xing, and R. Robidoux, "DRBD: Dynamic reliability block diagrams for system reliability modeling," Int. J. Comput. Appl. (IJCA), vol. 31(2009)No. 2, pp. 132–141.

[8] A. Abd-Allah, "Extending Reliability Block Diagrams to Software Architectures," Technical Report USC-CSE-97-501, Dept. of Computer Science, Univ. Southern California, 1997.

[9] Rausand & A. Høyland, System reliability theory: models, statistical methods, and applications , New York, USA, Wiley-Interscience, 2003.

[10] W. Wang, J. M. Loman, R. G. Arno, P. Vassiliou, E. R. Furlong, & D. Ogden, Reliability block diagram simulation techniques applied to the IEEE std. 493 standard network, IEEE Transactions on Industry Applications, 40(3), May/June 2004, pp. 887-895

[11] Distefano S, Puliafito A. Dynamic reliability block diagrams: overview of a methodology. In: [1] Proceedings of the safety and reliability conference (ESREL07),2007.

[12] Durga RK, Gopika V, Sanyasi RV, et al. Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. Reliab Eng Syst Safety, vol. 94(2009)No. 4, pp.872–83.

[13] J.B. Dugan, K.J. Sullivan, D. Coppit, "Developing a low-cost high-quality software tool for dynamic fault-tree analysis", IEEE Transactions on Reliability, vol 49(2000), pp 49-59.

[14] Suresh PV, Babar AK, Raj VV. Uncertainty in fault tree analysis: a fuzzy approach. Fuzzy Sets Syst 1996;83:135–41.

[15] Cepin M, Mavko B. A dynamic fault tree. Reliab Engng Syst Safety 2002;75:83 – 91.

[16] Amari S, Dill G, Howald E. A new approach to solve dynamic fault trees. Reliab Maintainability Symp 2003;27–30 Jan:374–9.

[17] H. Boudali, P. Crouzen, and M. Stoelinga. Dynamic fault tree analysis using input/output interactive Markov Chains. In Proc. ofthe 37th Annual IEEE/IFIP International Conference on DSN, pages 708-717. IEEE, 2007.

[18] Bobbio A, Franceschinis G, Gaeta R, Portinale L. Exploiting Petri nets to support fault tree based dependability analysis. In: Proceed- ings of the 8th international Workshop on Petri Net and Performance models (PNPM'99), 8–10 October 1999, Zaragoza, Spain, 1999. p. 146–155.

[19] Gulati R, Dugan JB. A modular approach for analyzing static and dynamic fault trees. Reliab Maintainability Symp. Annu Proc 1997; 13–16 Jan:57–63.

[20] Bobbio A, Daniele CR. Parametric fault trees with dynamic gates and repair boxes. In: Proceedings of the annual IEEE reliability and maintainability symposium, 2004, pp. 459–465.

[21] Murata, T. Petri Nets: Properties, analysis and applications. Proc. of the IEEE, 77(4), 1989, pp. 541-580.

[22] A. V. Ratzer, L. Wells, H. M. Lasen, M. Laursen, J. F. Qvortrup, et al., "CPN Tools for editing, simulating and analyzing colored Petri nets," in Proc. 24th Int. Conf. Application and Theory of Petri Nets, Eindhoven, Netherlands, Jun. 2003, pp. 450-462.

[23] Kristensen, L.M., Christensen, S., Jensen, K.: The practitioner's guide to coloured Petri nets. International Journal on Software Tools for Technology Transfer 2 (1998) 98–132

[24] C. Goldfarb and P. Prescod, The XML Handbook, Upper Saddle River, NJ, Prentice Hall, 2000.

[25] Ren Y, Dugan JB. Design of reliable systems using static and dynamic fault trees. IEEE Trans Reliab 1998:234±4