International Journal of Approximate Reasoning 51 (2010) 179–195

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

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

Supporting reliability engineers in exploiting the power of Dynamic Bayesian Networks ☆

Luigi Portinale *, Daniele Codetta Raiteri, Stefania Montani

Dipartimento di Informatica, Università del Piemonte, Orientale A. Avogadro, Via T. Michel 11, 15121 Alessandria, Italy

ARTICLE INFO

Article history:
Received 14 March 2008
Received in revised form 8 May 2009
Accepted 25 May 2009
Available online 2 June 2009

Keywords:
Dynamic Bayesian Networks
Dynamic fault trees
Repair policies
Reliability analysis
DBN inference

ABSTRACT

In this paper, we present an approach to reliability modeling and analysis based on the automatic conversion of a particular reliability engineering model, the Dynamic Fault Tree (DFT), into Dynamic Bayesian Networks (DBN). The approach is implemented in a software tool called RADYBAN (Reliability Analysis with DYnamic BAyesian Networks). The aim is to provide a familiar interface to reliability engineers, by allowing them to model the system to be analyzed with a standard formalism; however, a modular algorithm is implemented to automatically compile a DFT into the corresponding DBN. In fact, when the computation of specific reliability measures is requested, classical algorithms for the inference on Dynamic Bayesian Networks are exploited, in order to compute the requested parameters. This is performed in a totally transparent way to the user, who could in principle be completely unaware of the underlying Bayesian Network. The use of DBNs allows the user to be able to compute measures that are not directly computable from DFTs, but that are naturally obtainable from DBN inference. Moreover, the modeling capabilities of a DBN, allow us to extend the basic DFT formalism, by introducing probabilistic dependencies among system components, as well as the definition of specific repair policies that can be taken into account during the reliability analysis phase. We finally show how the approach operates on some specific examples, by describing the advantages of having available a full inference engine based on DBNs for the requested analysis tasks.

1. Introduction

It is well-known that the modeling possibilities offered by *Fault Trees* (*FT*), one of the most popular techniques for dependability analysis of large, safety critical systems, can be extended by relying on *Bayesian Networks* (*BN*) [2,4,14,22,24,21]. In [16], we have shown how BNs can provide a unified framework in which also *Dynamic Fault Trees* (DFT) [9], a rather recent extension to FTs able to treat several types of dependencies, can be represented. However, while reliability engineers are quite familiar with FT-based formalisms, they are not usually comfortable with the use of formalisms like BN and their extensions. This is also due to the fact that, for reliability purposes, simple and modular techniques are sometimes sufficient for the definition of the required analysis framework. Of course, a clear trade-off exists between the simplicity of the formalism and its modeling, as well as analysis capabilities. FTs are maybe the most simple combinatorial formalism in reliability analysis, but they fail in capturing important aspects like several kinds of dependencies among the system components [20,2,14]. DFTs overcome some of the limitations of standard FTs, by allowing some kinds of dynamic dependencies among components. They offer a quite simple and structured framework very useful for modeling purposes, but still quite limited

☆ This work has been partially supported by the EU-Project Crutial IST-2004-27513.
* Corresponding author.
E-mail addresses: portinal@di.unipmn.it (L. Portinale), raiteri@di.unipmn.it (D.C. Raiteri), stefania@di.unipmn.it (S. Montani).

0888-613X/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2009.05.009

Nomenclature

2TBN	2 time-slice Temporal Bayesian Network
BK	Boyen–Koller algorithm
BN	Bayesian Network
CAS	Cardiac Assist System
Cov	Coverage Set
Cov_{BE}	basic Coverage Set
CPT	Conditional Probability Table
CR	Component Repair policy
CTMC	Continuous Time Markov Chain
DBN	Dynamic Bayesian Network
DFT	Dynamic Fault Tree
FDEP	Functional DEPendency gate
FT	Fault Tree
JT	Junction Tree inference
PAND	Priority AND gate
PDEP	Probabilistic DEPendency gate
RB	Repair Box
SGR	Subsystem Global Repair policy
SLR	Subsystem Local Repair policy
TE	Top Event
WSP	Warm SPare gate

from the analysis point of view, especially if more complex modeling features (like for instance repair policies or special probabilistic dependencies) are required by the application.

The starting point of our work is to make available to reliability engineers a tool where they can take advantage of the simplicity and modularity of either plain FTs or DFTs, by making them available at the same time a more powerful analysis engine based on *Dynamic Bayesian Networks* (DBN). In fact, the quantitative analysis of DFTs typically requires to expand the model in its whole state space, and to solve the corresponding Continuous Time Markov Chain (CTMC) [9]. The above approach, even if can be improved through modularization techniques [11], usually suffers from a state explosion problem. With respect to CTMC, the use of a DBN allows one to take advantage of the factorization in the temporal probability model, via the conditional independence assumptions represented in the DBN. Moreover, by taking into account the basic features of DBNs, we aim at offering to the user a set of useful extensions to the basic DFT formalism, like the introduction of sophisticated probabilistic dependencies among the failure of system components, the introduction of specific repair policies and the possibility of analyzing the system behavior under the gathering of specific observations on the system components. Our approach is based on a translation of the extended DFT model into an equivalent DBN and has been implemented in a tool called RADYBAN (Reliability Analysis with DYnamic BAyesian Networks) [17]. It allows the design of the reliability model through a graphical interface where the analyst can access and exploit all the familiar modeling constructs of DFTs, as well as the supported extensions cited above; the resulting model is then compiled into an equivalent DBN and the analysis is performed in a transparent way to the user, who has just to specify the desired type of analysis algorithm. This has the advantage of avoiding the reliability engineer to learn the details of a completely new formalism; in case she/he is not willing to forsake her/his traditional formalism, she/he can still use it (possibly with few simple extensions) while having available a more powerful analysis engine.

The rest of the paper is organized as follows: In Section 2 we briefly review the basic frameworks of DFTs, while in Section 3 we introduce the extensions we have provided to augment the modeling power of the basic formalism. In Section 4 we recall DBN basics and in Section 5 we sketch the main functionalities of our approach, in particular by taking into consideration, through a running example, the compilation process concerning the provided modeling extensions. Finally, in Section 6, we show a more general application of the approach on a case study taken from [5] and based on a real-world system. Conclusions are then reported in Section 7.

2. Dynamic fault trees

Fault Trees allow one to represent the combination of elementary causes that lead to the occurrence of an undesired catastrophic event named the *Top Event* (*TE*) [2,14]. By specifying failure probabilities on the basic components of the modeled system (the elementary causes of the *TE*, also called *basic events*), the whole system unreliability (probability of the *TE*) at a given mission time can be computed. The model assumes every event to be Boolean, with value `true` corresponding to a failure event.

In recent years, an effort has been documented in the literature, aimed at increasing the modeling power of *FT* by including new primitive gates, able to accommodate complex kinds dependencies. This augmented *FT* language is referred to by the