Chapter 52
Improving Dependability of Robotics Systems, Experience From Application of Fault Tree Synthesis to Analysis of Transport Systems

Nidhal Mahmud
University of Hull, UK

ABSTRACT

The use of robotics systems is increasingly widespread and spans a variety of application areas. From manufacturing, to surgeries, to chemical, these systems can be required to perform difficult, dangerous and critical tasks. The nature of such tasks places high demands on the dependability of robotics systems. Fault tree analysis is among the most often used dependability assessment techniques in various domains of robotics. However, there is still a lack of adjustment methods that can efficiently cope with the sequential dependencies among the components of such systems. In this paper, the authors first introduce some relevant techniques to analyze the dependability of robotics systems. Thereafter, an experience from research projects such as MAENAD (European automotive project investigating development of dependable Fully Electric Vehicles) is presented; emphasis is put on a novel approach to synthesizing fault trees from the components and that is suitable for modern high-technology robotics. Finally, the benefits of the approach are highlighted by using a fault-tolerant case study.

1. INTRODUCTION

The use of robotics systems is widespread and spans a variety of application areas. From healthcare, to manufacturing, to nuclear power plants, to space missions, these systems are typically conceived to perform difficult, rote, dangerous or critical tasks. The nature of such tasks —e.g., surgery operations, radioactive waste clean-up or space mining— places high demands on the dependability of robotics

DOI: 10.4018/978-1-5225-8060-7.ch052

systems. Dependability is an umbrella concept which associates reliability, safety, availability, security and maintainability. However, in this paper emphasis is mainly put on the reliability and safety of the robotics systems. The latter can be seen as an extension of the former. That is, a system is in a safe state when it is in any state of correct service (the system is still reliable), or when it is in a state of incorrect service but without catastrophic consequences on the environment (Avižienis, Laprie, Randell, & Landwehr, 2004). For example, a robot arm-based hazardous waste retrieval manipulator can still function safely after failure of one of two redundant sensors in a critical joint, assuming that no compromising event occurred somewhere else.

The preoccupations in the dependability of robotics systems are not new. Fault Tree Analysis (FTA, Vesely 1981) and Failure Modes and Effects Analysis (FMEA, IEEE Std.352 1987) are among the most often used techniques in various domains of robotics. For instance, Visinsky, Walker, and Cavallaro (1993) describe the use of FTA for robots operating in remote and hazardous environments. This technique is also emphasized by Walker, and Cavallaro (1996) in the context of a radioactive waste clean-up robot manipulator. Moreover, Guiochet, Tondu, and Baron (2001) describe the importance of both FTA and FMEA in the assessments of medical robots. Other fields of application include industrial robots, like in (Karbasian, Mehr, & Agharajabi, 2012); modular and swarm robots, like in (Winfield & Nembrini, 2006), (Murray, Liu, Winfield, Timmis, & Tyrrell, 2012), and in (Bjerknes & Winfield, 2013); as well as exploration and target searching robots, like in (Yakymets, Dhouib, Jaber, & Lanusse, 2013).

The widespread use of FTA in the dependability assessment of complex systems is mainly due to the flexibility and ease of use of the fault trees. These enable the use of efficient Boolean calculus in the elimination of component failures that are irrelevant to the total failure of the system, and thereby simplifying the process to produce overall probabilities of system hazards. In this direction, a number of authors proposed to generate fault trees from system models. For example, Rauzy (2002) described an approach to compiling fault trees from mode automata which are used to capture complex behavioral aspects of systems. Another example in the same vein consists of the work in (Joshi, Vestal, & Binns, 2007). The authors produced a static fault tree generator prototype based on AADL models. AADL is an Architecture Analysis and Design Language which is intended to be an aerospace standard.

However, since it is impossible to represent dependencies among components in a static (i.e., ‘pure’ Boolean) fault tree, some approaches to extending its modelling power with dynamic gates were developed. The most prominent example is the Dynamic Fault Tree (DFT) approach (Dugan, Bavuso, & Boyd, 1992). The added dynamic features in the DFTs are useful to preserve the significance of the sequencing of events often exhibited by dynamic systems. Therefore, Dehlinger and Dugan (2008) extended the work described in (Joshi et al., 2007) such that dynamic rather static fault trees are generated from the AADL models. Although the DFTs are very favorable to quantitative studies using diverse methods e.g. Markov techniques (Boudali, Crouzen, & Stoelinga, 2007, 2010), the full power of the Boolean calculus offered through the use of the conventional static fault trees was sacrificed here. In other words, there has been less focus on qualitative analysis of the added gates at the level of the DFT, i.e. downsizing the fault tree and determining its reduced (or minimal) cut-sequences. Consequently, Mahmud and Mian (2013) proposed a transformation of AADL models to qualitative Temporal Fault Trees (TFTs). The TFTs which are targeted in this work preserve the advantage of using the Boolean methods which is rather extended with temporal logic calculus, see the temporal laws in (Walker & Papadopoulos, 2009) as well as the algebraic framework and theorems in (Merle, Roussel, & Lesage, 2011). This approach is supported by a body of work, especially a suitable transformation of an automata hierarchy into a combination of static and dynamic fault trees as appropriate and depending on the nature of the system