Reliability Engineering and System Safety 97 (2012) 14–26

Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

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

Timing analysis of safety properties using fault trees with time dependencies and timed state-charts

Jan Magott *, Pawel Skrobanek

Institute of Computer Science, Control and Robotics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland

ARTICLE INFO

Article history:
Received 10 July 2009
Received in revised form
24 August 2011
Accepted 18 September 2011
Available online 24 September 2011

Keywords:
Safety analysis
Fault tree
Fault tree with time dependencies
Timed state-chart
Railroad crossing

ABSTRACT

Behavior in time domain is often crucial for safety critical systems. Standard fault trees cannot express time-dependent behavior. In the paper, timing analysis of safety properties using fault trees with time dependencies (FTTDs) and timed state-charts is presented. A new version of timed state-charts (TSCs) is also proposed. These state-charts can model the dynamics of technical systems, e.g. controllers, controlled objects, and people. In TSCs, activity and communication times are represented by time intervals. In the proposed approach the structure of FTTD is fixed by a human. Time properties of events and gates of FTTD are expressed by time intervals, and are calculated using TSCs. The minimal and maximal values of these time intervals of FTTD can be calculated by finding paths with minimal and maximal time lengths in TSCs, which is an NP-hard problem. In order to reduce the practical complexity of computing the FTTD time parameters, some reductions of TSCs are defined in the paper, such as sequential, alternative, loop (iteration), and parallel. Some of the reductions are intuitive, in case of others—theorems are required. Computational complexity of each reduction is not greater than linear in the size of reduced TSC. Therefore, the obtained results enable decreasing of the costs of FTTD time parameters calculation when system dynamics is expressed by TSCs. Case study of a railroad crossing with a controller that controls semaphores, gate, light-audio signal close to the gate will be analyzed.

1. Introduction

Standard fault-tree analysis (FTA) [1] is widely used in safety and reliability analysis. FTA can be used to determine the following: minimal cut sets that cause a hazard [2], probabilities of the hazard, and faults. When designing safety-critical real-time systems, the following time parameters often have to be considered: delay time between the causes and effect, hazard tolerance time if the hazard can be tolerated for some duration, fault reaction (exposure) time [3], fault detection time, and time of non-persistent consequences [4]. The values of these parameters are required in development of safe-guards and recovery subsystem, or in evaluation of the fault reaction time. Standard fault trees can express neither time-dependent, nor sequence-dependent dynamic behavior. Hence, in standard FTA, timing analysis of safety properties cannot be performed.

There are different directions in the development of standard FTA.

Dynamic fault trees [5,6] enable extension of standard probabilistic safety assessment (PSA) to perform analysis of phased-mission systems. In these systems, different operational modes can occur over time. This approach does not address the real-time requirements for concurrent processes of one phase.

The notion of state-event-fault-trees (SEFT) has been presented in [7] and applied in safety analysis of component-based systems [8]. In SEFT there are states that last over a period of time and events that are immediate. The analysis using SEFT is oriented on PSA.

In order to analyze time aspects there are two approaches: temporal fault tree analysis and timing properties analysis.

Temporal fault trees (TFTs) are based on different temporal logics. TFTs presented in the paper [9] are based on duration calculus [10]. TFTs given in [11] are based on the interval temporal logic [12]. In the paper [13], the definition of priority-AND gates is given. Papers [13,14] deal with sequence-dependent dynamic behavior. The analysis using TFTs is qualitative when such problems as reachability, satisfiability of a formula are considered. TFTs in [15] contain many new gates. Some of them express quantitative time relations between causes and effects.

The above extensions of FTA can model sequence-dependent dynamic behavior. However, they are not sufficient when timing analysis of safety properties is required.

Our goal is to extend FTA by introducing quantitative timing analysis.

The timing properties analysis can be performed using fault trees with time dependencies (FTTDs) [16–18] that are a current

* Corresponding author.
E-mail addresses: jan.magott@pwr.wroc.pl (J. Magott), pawel.skrobanek@pwr.wroc.pl (P. Skrobanek).

0951-8320/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2011.09.004

research issue [19]. In FTTD, events are expressed by their duration times. Additionally, there are: generalization gates and causal gates. In the case of generalization gates: the effect is either a combination of causes or just one of the causes. The causal gates are characterized by the delay times between the causes and the effect.

Nowadays, UML diagrams are widely used in software engineering. When designing real-time systems, UML state-charts can be used. In timed state-charts given in paper [20], a pair of time parameters is assigned to transition. In UML state-charts, activity is associated with the state. Therefore, in timed state-charts that are defined in our paper, execution time of activity performed in state is given by the pair of minimal and maximal values assigned to this state.

In the approach presented in this paper, a new version of timed state-charts (TSCs) is used to model the dynamics of technical systems and people that are important in safety analysis. In TSCs, activity times and interaction times are represented by time intervals. These dynamics models are the starting point for constructing the FTTDs. Structure of the FTTDs is created manually. Time parameters of events and gates of FTTDs are expressed by time intervals. A method of calculation of these intervals using TSC dynamics models is presented. Minimal and maximal values of events and gates time intervals are given by lengths of minimal and maximal paths in TSCs. Hence, FTTDs can be determined in part automatically. This paper is an extension of the paper [21], which formulated the first approach in partially automatic generation of FTTDs on the basis of UML state-charts.

Contribution of the paper is as follows. In the paper we outline how to calculate the minimal and maximal values of event duration times and delay times between input and output events of causal gates of a FTTD. The minimal and maximal values of these time parameters of FTTD are calculated by finding paths with minimal and maximal time lengths in TSCs, i.e., by solving minimal and maximal execution time problem (MMETP) [22]. This problem is NP-hard [22]. In order to reduce the practical complexity of FTTD time parameter computations, reductions of TSCs such as sequential, alternative, loop (iteration), and parallel are defined in the paper. Some of the reductions are intuitive, but for the others theorems are required. The reductions are computationally effective. Case study of a railroad crossing with controller that controls: semaphores, gate, and light-audio signal close to the gate will be analyzed. Dynamics of the above objects will be expressed by TSCs, and time parameters of FTTD will be given.

The paper is organized as follows. TSCs and some examples of modeling the object dynamics using TSCs are presented in Section 2. FTTDs are then described. A calculation method, determining the time parameters of events and gates of FTTDs on the basis of TSCs, is outlined in Section 4. In the next section, the case study of a railroad crossing is given. In Section 6, a comparison with related work is given. Finally, there are the conclusions.

2. Timed state-charts

TSCs are based on UML state-charts [23]. It is assumed that the reader is familiar with basic UML state-chart models.

Details of syntax and semantics of TSCs are presented in paper [22].

Examples of TSCs are given in Sections 2, 4, and 5.

System is combined from the set of n TSCs: $S=\{SC_1,\ldots,SC_n\}$.

St is the finite set of states, which can be partitioned into disjoint subsets: *INITIAL*, *SIMPLE*, *FINAL*, *AND*, *XOR*, where: *INITIAL* is a set of initial pseudostates, *SIMPLE* is a set of simple states, *FINAL* is a set of final states, *AND* is a set of AND type states, and *XOR* is a set of XOR type states.

The main difference between UML state-charts and TSCs is the pair $\langle t_{\min},t_{\max}\rangle$ of times assigned to simple states of TSCs. The pair can represent execution time of an activity performed in the state or can be undefined. Another difference is the representation of message transmission times.

Description of the state is placed inside the graphical representation of it, and has the following form:

S: meaning_of_the_state $\langle t_{\min},t_{\max}\rangle$

where

- $S\in St$, where St is the finite set of states,
- $t_{\min}$ and $t_{\max}$ are the minimal and maximal values of simple state time function,
- meaning_of_the_state is informal meaning of the state.

$StT:SIMPLE\rightarrow R_+\times(R_+\cup\{\infty\})\cup\{\langle\_,\_\rangle\}$ is a simple state time function. R_+ is the set of non-negative real numbers. Values $t_{\min}\in R_+$, $t_{\max}\in R_+\cup\{\infty\}$, respectively, where $t_{\min}\le t_{\max}$, are the minimal, the maximal duration times of the activity associated with a simple state. The meaning of the symbol ∞ is the following: $\infty+r=\infty-r=\infty$, where $r\in R_+$. The symbol ∞ represents infinite time, while "_" denotes the undefined value of time.

Minimal and maximal values of simple state time function can be obtained by the analysis of source code of the activity associated with the state. These values can be in a wide range. Our approach accepts such cases as when the minimal time is underestimated and the maximal time is overestimated. Such cases are the cause of inaccuracy.

The transitions are labeled (see Fig. 1).

On the basis of the labeling, there are three types of transitions between the states:

- trigger transition,
- timed transition,
- completion transition.

The label of the **trigger transition** has the following form:

operation_name[guard]/sequence_of_actions

Only the *operation_name* is obligatory.

The informal semantics of this transition is as follows. Let us consider the transition from source state S1 to target state S2. This transition can occur in time instant τ' when the *operation_name* call event is occurring, provided that the guard is satisfied. When the above conditions are satisfied, then the *sequence_of_actions* is executed and the TSC can transit from the state S1 to the state S2. It is assumed that:

1. operation call event lasts zero time,
2. the execution time of the *sequence_of_actions* is equal to zero.

If the second assumption cannot be accepted then a new state has to be added according to Fig. 2. At this figure, the label is of the form "operation_name [guard]".

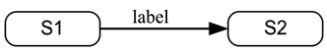

Fig. 1. The labeled transition from state S1 to state S2.

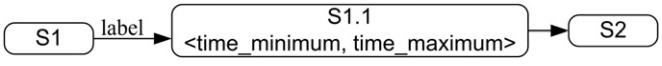

Fig. 2. State S1.1 represents the execution time of the *sequence_of_actions* with non-zero execution time.