Safety Science 109 (2018) 130–143

Contents lists available at ScienceDirect

Safety Science

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

System safety assessment based on STPA and model checking

Alheri Longji Dakwat[*], Emilia Villani

Instituto Tecnológico de Aeronáutica (ITA), Pça. Mal. Eduardo Gomes, 50, São José dos Campos CEP 12228-900, Brazil

ARTICLE INFO

Keywords:
Model checking
STPA
Safety assessment
UPPAAL

ABSTRACT

Due to the current pace of technological growth, the management of system safety has evolved with complex causes of accidents that are often beyond the identification of traditional safety assessment techniques. Recently, the hazard analysis tool Systems Theory Process Analysis (STPA) has emerged as an approach to improve safety of modern complex systems in concert with other hazard analysis tools. However, the effectiveness of STPA is a debatable issue in the industry and efforts towards incorporating some level of formalization in STPA steps are welcome. In this direction, this work presents a method for combining STPA and model checking, in order to provide a formal and unambiguous representation of the system under analysis and the threats identified by STPA. A practical case study of a robotic flight simulator is presented as an example of the proposed method. The results achieved with the proposed approach indicates that the merging of the two techniques improves the knowledge about the system under design and the consistence of the design changes proposed to tackle the safety constraints identified in STPA.

1. Introduction

The computer revolution has made significant impact in the society, especially through embedded systems. Embedded systems have made possible the incorporation of many new functionalities that are implemented in software. As we increase the complexity of the systems under design, traditional bottom-up or top-down safety assessment techniques, such as Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA), become insufficient to assure the product safeness (Stringfellow et al., 2010). One limitation is the difficult of exploring all the possible scenarios that may arise from the combination of the components' behavior when interacting with software products and human beings. In this context, the hazard analysis tool Systems Theory Process Analysis (STPA) has emerged as an approach for improving safety of modern complex systems in concert with other hazard analysis tools. The traditional approaches to safety, such as HAZOP, FMEA, and FTA, were developed for systems that were built more than 50 years ago. Their focuses is to address failures based on chain of events related to component malfunction. Modern embedded systems have developed new paths to accidents based on complex errors that emerge even when everything is working according to its specification (Leveson, 2011). A systematic review on STPA observed that there has been published works on STPA since 2000s and a sharp increase from 2011 in areas that include aviation, automotive, defense, software, space, medical and environment (Lahoz, 2016). This survey underscores the gaining relevance and acceptance of STPA as a safety tool.

Despite being a promising approach, STPA has not yet been extensively adopted by industry or recommended by certification authorities. Among the limitations of STPA, are the lack of formalism (Harkleroad et al., 2013) and the potential dependence of the results on the experience of the person that applies the method. In order to overcome these limitations, this work proposes the use of STPA combined with model checking, a formal verification technique. The challenges emerging from the exponential growth of embedded systems is resulting in an increasing interest in formal methods. Industry is currently considering the adoption of formal techniques, such as model checking and theorem proving, to complement the more common verification approaches of simulation and testing. These techniques are gaining grounds through industrial case studies, reinforcing the benefits of using formal methods (Clarke and Wing, 1996; Benaben et al., 2002).

In the current work, automata are adopted as a formal language to model the behavior of the system components, including software, while model checking is used to check the consistence of the safety constrains that are obtained from the STPA approach. The contribution of this work is safety assessment approach that combines STPA and model checking. From STPA, the proposed approach inherits the simplicity and ability of identifying scenarios that can lead to hazard situations. Complementing STPA, model checking contributes with the complete exploration of the system state space. It complements the identification of unsafe control actions and verifies the effect of the safety constraints in inhibiting hazard situations.

A practical case study of the embedded system of a flight simulator

[*] Corresponding author.
E-mail addresses: alheri.dakwat@airforce.mil.ng (A.L. Dakwat), evillani@ita.br (E. Villani).

https://doi.org/10.1016/j.ssci.2018.05.009
Received 7 June 2017; Accepted 15 May 2018

prototype is used to evaluate this proposal. The simulator, known as SIVOR, explores the use of an industrial robotic manipulator to provide the acceleration feelings to the pilot. Due to the innovative nature of the SIVOR project, it offers a rich field of analysis of new types of hazards.

This paper is organized as following. Section 2 discusses approaches to safety based on the concepts of STPA and Model Checking using UPPAAL. Section 3 presents the workflow of the proposed method and Section 4 details its application to the SIVOR case study. Finally, Section 5 draws some conclusions.

2. Literature review

A systematic literature review on STPA (Lahoz, 2016) indicated a positive trend in its acceptance. The review presented a summary of STPA applications since the early 2000s. These included 23 methods, 37 approaches, 8 tools and about 176 case study applications, out of which the highest numbers were in aviation, medicine, automotive and space industries and rising. From this extensive review, we selected 3 papers that went further to consider some form of scientific verification through formal methods. We then complement this set with 15 works on system safety based on STAMP/STPA published from 2010 in the Science Direct and IEEE databases after 2010, resulting in a final set of 18 analyzed works (Abdulkhaleq et al., 2015; Lopes and Hirata, 2015; Chiesi, 2016; Asplund et al., 2012; Cody et al., 2011; Takuto et al., 2011; Suo et al., 2011; Haruka et al., 2011; Sulaman et al., 2014; Gerrit et al., 2016; Asim et al., 2017; Kleve, 2016; Maria et al., 2016; Robert et al., 2017; Petit, 2004; Rejzek et al., 2015; Sun and Zhong, 2013; Shuai and Deming, 2014; Asim and Stefan, 2015). Among these works, we discuss in this section those that were more closely related to our proposal.

The work of Asplund et al. (2012) presented an analysis of the implementation of STPA on an industrial tool chain. Some major difficulties observed included how to identify basic risks for new contexts, defining control structures and limiting the domains. The authors identified the need of tools to support STPA as well as perform simulations through which STPA users can explore the consequences of inadequate control in a control structure and equally facilitate the appropriate strategies for updating the system.

Another similar effort, by Lopes et al. (Baier and Katoen, 2007), proposed a rule-based approach to perform Step 1 of STPA. The proposal was validated based on its application to the single controller of an automated door system of a train. Although the authors conclude that the result was encouraging, the application of the rule-based approach on larger systems with multiple controllers and Step 2 was uncertain.

In another work, Abdulkhaleq et al. (Leveson, 2013) proposes the use of a software tool (ASTAMP) to perform STAMP (CAST and STPA) activities. An extended version of the software, XSTAMP has continued to evolve with the extension to perform verification. The possibility of combining UML and STPA (Kleve, 2016) also presents a practical approach to safety based on the integration of system and safety engineering.

Two other published works (Maria et al., 2016) and (Petit, 2004) combine formal methods with STPA. Sun & Zhong (Maria et al., 2016) considered using STPA thinking combined with a technique to convert natural language to automaton using a traffic light as example. Asim & Stefan (Petit, 2004) proposes to integrate STPA with state machine analysis. In this approach, they used STPA to determine the control actions of a model and then constructed a finite state machine of the controller with state variables to examine each potential hazardous control action and the effect of combined system states. One of the challenges identified in this work as the need of developing an appropriate model of the real system.

Based on the literature review, we believe that the challenges of STPA are mainly due to the lack of a scientific method to address the hypothetical nature involved in its application. This is further confounded by the problem of how and what is to be verified in STPA when using UPPAAL. To resolve this, we considered that both STPA and UPPAAL have limitations when applied independently. The hypothetical quality of STPA allows it to accommodate unforeseen hazards while model checking is restricted to the system model, not the actual system (Rejzek et al., 2015). That is, STPA looks at hypothetical actions **(WHAT IF?)** as a means to eliminating hindsight bias (Leveson, 2011) while model checking looks at available information **(WHAT IS?)**. The outcome of this union is considered as the answer, **(WHAT WILL BE)** as illustrated in Fig. 1. These assumptions could also be viewed from the perspectives of human and machine limitations. That is, the human can make predictions that may be right or wrong but the computer cannot. Likewise, the computer cannot predict the future as it can only make calculated analysis based on existing data. In view of these challenges, the problem considered is how to achieve a scientific method that can facilitate the combination STPA and model checking or validating the results of STPA using scientific methods.

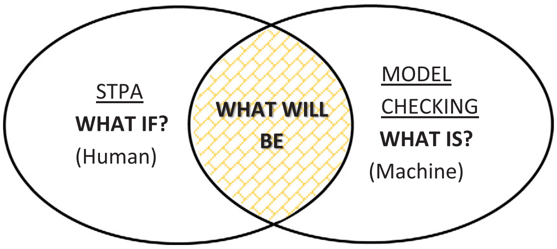

Fig. 1. Assumptions of proposed method.

3. The proposed method

STPA is one of the hazard analysis tools of STAMP conceived by Nancy Leveson in the early 2000s. STPA is performed in 2 steps (Leveson, 2011) and its goal is to identify scenarios leading to hazards and then define safety constrains that the system under design must fulfill. STPA identifies a larger set of causal factors than other safety techniques, many of them not involving failures or unreliability of components (Sun and Zhong, 2013). It also provides appropriate information to guide the design process, instead of requiring a design to exist before the analysis can start (Leveson, 2011).

On the other hand, model checking is a verification technique that explores all possible system states in an exhaustive manner (Rejzek et al., 2015). To confirm that a given system model truly satisfies a certain property, model checking must examine all possible system scenarios in a systematic manner (Shuai and Deming, 2014). The 3 basic steps in model checking involves building the model, formalizing the properties to be verified and using the model checker tool to check if the specified properties are true or false in the dynamic behavior of the model. In this work, we use the model checker UPPAAL, which requires the system modelling as a network of timed automata (Asim and Stefan, 2015).

An appropriate workflow to combine both methods is based on the premise that both complement each other based on their characteristics. For example, STPA requires a preliminary assessment that defines elements and control actions of the system. These elements could be assumed to correspond to templates and channels in UPPAAL. The workflow of the proposed method is organized in 6 steps that are illustrated in Fig. 2 and discussed in subsequent paragraph.

The inputs and outputs of each step are illustrated in Fig. 3. The workflow starts with Step 1, from STPA. The output of this step yields the system goals, elements (human, software and hardware), list of hazards and accidents, list of control actions, the control structure and a description of system operation (UML charts may come in handy during this process depending on the complexity of the system). Based on these outputs, Steps 2 and 3 can be performed concurrently. Step 2, which is