Safety Science 51 (2013) 302–318

Contents lists available at SciVerse ScienceDirect

Safety Science

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

Supporting systems of systems hazard analysis using multi-agent simulation

Rob Alexander [*], Tim Kelly

Department of Computer Science, University of York, York, United Kingdom

ARTICLE INFO

Article history:
Received 12 July 2011
Received in revised form 15 June 2012
Accepted 29 July 2012
Available online 3 September 2012

Keywords:
Safety
Simulation
System of systems
Hazard analysis
Multi-agent

ABSTRACT

When engineers create a safety-critical system, they need to perform an adequate hazard analysis. For Systems of Systems (SoSs), however, hazard analysis is difficult because of the complexity of SoS and the environments they inhabit. Traditional hazard analysis techniques often rely upon static models of component interaction and have difficulties exploring the effects of multiple coincident failures. They cannot be relied on, therefore, to provide adequate hazard analysis of SoS. This paper presents a hazard analysis technique (SimHAZAN) that uses multi-agent modelling and simulation to explore the effects of deviant node behaviour within a SoS. It defines a systematic process for developing multi-agent models of SoS, starting from existing models in the MODAF architecture framework and proceeding to implemented simulation models. It then describes a process for running these simulations in an exploratory way, bounded by estimated probability. This process generates extensive logs of simulated events; in order to extract the causes of accidents from these logs, this paper presents a tool-supported analysis technique that uses machine learning and agent behaviour tracing. The approach is evaluated by comparison to some explicit requirements for SoS hazard analysis, and by applying it to a case study. Based on the case study, it appears that SimHAZAN has the potential to reveal hazards that are difficult to discover when using traditional techniques.

1. Introduction

A growing challenge for safety engineers is maintaining the safety of large-scale military and transport Systems of Systems (SoSs), such as Air Traffic Control (ATC) networks and military units with Network Enabled Capability (NEC). The term “SoS” can be defined in terms of key characteristics (Alexander et al., 2004): SoS consist of multiple components that are systems in their own right, each having their own goals and some degree of autonomy but needing to communicate and collaborate in order to achieve overall SoS goals. SoS are typically distributed over large areas (such as regions, countries or entire continents), and their components frequently interact with each other in an ad-hoc fashion. It follows that military and transport SoS have the potential to cause large-scale destruction and injury. This is particularly true for SoS incorporating new kinds of autonomous component systems, such as Unmanned Aerial Vehicles (UAVs).

This paper is concerned with one aspect of the safety process for SoS, specifically *hazard analysis*: determining the distinct causal chains by which the behaviour of the SoS can lead to an accident. Hazard analysis is a crucial part of any risk-based safety approach, but the defining characteristics of SoS make it very difficult.

Recent developments in SoS are likely to worsen the SoS safety problem. For example, there is a move towards dynamic reconfiguration, which greatly expands the number of system states that needs to be considered; any analysis may need to be carried out for all possible configurations. Similarly, SoS increasingly use ad hoc communications, meaning that information errors can propagate through the system by many, unpredictable, routes.

These factors overwhelm the ability of manual hazard analysis and therefore suggest a need for automated hazard analysis. There are a few automated approaches specifically designed for SoS safety, but what exists typically lacks any kind of systematic modelling process or has a very limited applicability in terms of the models it can analyse, and requires models that are built specifically for that analysis (for example, many approaches based on model-checking). Most of the extant SoS-specific methods are aimed at safety risk assessment (deriving quantitative values for the risk posed by the SoS); few of them are focussed specifically on hazard identification and hazard analysis (discovering the different hazards in the SoS and the distinct combinations of causes that can lead to them).

This paper presents SimHAZAN: a partly-automated hazard analysis method for SoS that avoids some of the problems associated with existing techniques. In particular, it has a systematic modelling process and a separate analysis approach that can be applied either to models developed through that process or to models developed

[*] Corresponding author. Address: Department of Computer Science, University of York, Deramore Lane, York YO10 5GH, United Kingdom. Tel.: +44 1904 325 474, +44 7813 134 388.

E-mail addresses: rob.alexander@york.ac.uk (R. Alexander), tim.kelly@cs.york.ac.uk (T. Kelly).

0925-7535/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ssci.2012.07.006

by other means. The process provides specific support for hazard analysis – it leads directly to a qualitative understanding of the chains of causes by which hazards occur. It can fit into an existing risk-based safety process by providing a source of hypotheses about hazards that can then be tested and mitigated by safety engineers. This presents a case study that demonstrates the method's potential to reveal hazards, and causes of hazards, that other methods do not. Because it provides hypotheses (rather than confirmatory evidence of safety), it can be used despite concerns about the validity and fidelity of simulation models – it can be evaluated on a purely return-on-investment basis, without necessarily making claims about achieving coverage of all possible hazards and causes.

This paper is summary of SimHAZAN, limited by the space available. For a fuller description (including a thorough literature review and many example artefacts from the case study) the reader is referred to (Alexander, 2007). That text does not use the term "SimHAZAN", but the approach presented here is a refinement of the approach described there.

The following section discusses the challenges of SoS hazard analysis and describes what is required of any method if it is to meet those challenges. Section 3 gives an overview of SimHAZAN, then Sections 4 and 5 give detailed accounts of the two major parts of SimHAZAN: modelling and analysis. Section 6 presents a case study, which also serves as an illustration of the method in practice. Section 7 briefly discusses the use of SimHAZAN in practical safety engineering, and Section 8 concludes the paper with a discussion of how well SimHAZAN meets the identified challenges; where there are shortfalls or opportunities, it outlines directions for future work.

2. The challenge of SoS hazard analysis

Aitken states that "*An SoS Hazard is the combined behaviour of two or more distinct nodes within the SoS that could lead to an accident. An accident that can be described by behaviour confined to a single node (i.e. a single system hazard) is not a SoS accident, even if that node is acting as part of a SoS*" (Aitken et al., 2011). A "node" here is a component of the SoS – something that is part of the SoS but has some degree of autonomy with respect to it. Examples could include an aircraft or a group of rescuers on the ground. SoS hazard analysis is thus the process of finding the conditions in which two or more distinct nodes can behave so as to give rise to an accident, and then finding the causal paths by which those conditions could be reached from a safe state. The specific objective of SimHAZAN is thus to associate the behaviours, states and interactions of SoS nodes with accidents.

The reader may ask, at this juncture, why there is such a concern with finding new hazards. After all, many hazard analysis techniques *start* with most hazards known, and concentrate on finding their causes. HAZOP is a typical example – although it works forwards from deviations in order to find their consequences, the set of dangerous consequences (system hazards) is mostly known at the start. This may not be typical for SoS. Although some hazards will be known at the start, many will only become apparent through exploratory analysis of the system. Hence, the current work is focussed on identifying possible behaviour variations ("deviations") of individual entities within the SoS ("nodes") and using simulation to project accidents that could occur because of those. The output is a set of causal chains, and it is then a task for engineers to turn those into a manageable set of hazards.

2.1. The problems of SoS hazard analysis

Perrow (1984) discusses what he calls 'normal accidents' in the context of complex systems. His 'Normal Accident Theory' holds that any complex, tightly-coupled system has the potential for catastrophic failure stemming from simultaneous minor failures. Similarly, Leveson (2002) notes that many accidents have multiple causes, which are all necessary and (only) collectively sufficient for the accident to occur. In such cases it follows that an investigation of any one cause *prior to the accident* (i.e. without the benefit of hindsight) might not have made the accident plausible to an analyst.

An SoS can certainly be a 'complex, tightly-coupled system', and as such is likely to experience such accidents. One strategy to improve SoS safety is to decouple the elements of the system, and Marais et al. note that this has worked well in the design of Air Traffic Control (ATC) SoS (Marais et al., 2009). This decoupling can have a cost in performance, however – for example, there are moves in ATC to move to free flight models where aircraft interact via decentralised data exchange which may increase airspace performance at the cost of increased coupling.

A 'normal accident' could also result from actions by each of two nodes that were safe in themselves (in their assumed context of use), but that are hazardous in combination with each other and the wider SoS context. Such *emergent hazards* are a major concern for SoS. These problems are also present in conventional systems – see, for example, Wilkinson and Kelly (1998) – but the characteristics of SoS exacerbate them.

Raheja and Moriarty (2006), when discussing SoS safety, comment that SoS can be tightly coupled at long distances and hence a change in one part of the system may have difficult-to-predict consequences in other parts. They also stress the contribution of system architecture to safety, noting however that in SoS the architecture may be dynamic. In decentralised systems with dynamic structure, predicting the long-range effects of local events is notoriously difficult.

The difficulty of detecting hazardous combinations of events is greater because many SoS will incorporate component systems drawn from multiple manufacturers, developed at different times, and operated by multiple organisations. The evolutionary and dynamic nature of SoS structures means that a component system designer may never understand the entire SoS context.

A further complication is that SoS elements, by definition, have some degree of operational autonomy – they have some goals of their own (such as self preservation) in addition to goals at a higher level (such as destroying priority targets). There are likely, indeed, to have goals at several levels – individually, local to the team or unit, and globally to the whole SoS. The safety-critical behaviour of an SoS can thus only be understood by using models that can capture these goals, and analyses that can derive their (combined) consequences.

Discussion of military SoS inevitably involves reference to cutting-edge technologies, such as advanced unmanned vehicles. This creates an additional pressure in that of course, being novel, these technologies may not be well understood. Their developers often do not know how to make them safe, or how to assure others that they are safe. Unmanned vehicles are a particular concern in that they are likely to be very dumb responders to information shared over the SoS – they are particularly vulnerable to errors in network data or commands. This creates a need for modelling and analysis approaches that can capture some of their behaviour and help safety engineers determine the consequences in the SoS context.

Existing work on SoS dependability concentrates mostly on software and networks – there is little attention given to *embodied* SoS. An example of this is the DSoS project at the University of Newcastle (Gaudel et al., 2003), which almost exclusively studied enterprise networks. Safety requires more than this – engineers need to consider the physical nodes (e.g. aircraft and weapons systems) that are part of the SoS, along with the organisational structure of its human components (Rasmussen, 1997).