Reliability Engineering and System Safety 115 (2013) 146–160

Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

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

Scenario clustering and dynamic probabilistic risk assessment

Diego Mandelli [a,*], Alper Yilmaz [b], Tunc Aldemir [a], Kyle Metzroth [a], Richard Denning [a]

[a] Nuclear Engineering Program, The Ohio State University, Columbus, OH, United States
[b] Photogrammetric Computer Vision Lab., The Ohio State University, Columbus, OH, United States

ARTICLE INFO

Article history:
Received 2 February 2012
Received in revised form
3 January 2013
Accepted 11 February 2013
Available online 28 February 2013

Keywords:
Transient analysis
Scenario clustering
Dynamic PRA

ABSTRACT

A challenging aspect of dynamic methodologies for probabilistic risk assessment (PRA), such as the Dynamic Event Tree (DET) methodology, is the large number of scenarios generated for a single initiating event. Such large amounts of information can be difficult to organize for extracting useful information. Furthermore, it is not often sufficient to merely calculate a quantitative value for the risk and its associated uncertainties. The development of risk insights that can increase system safety and improve system performance requires the interpretation of scenario evolutions and the principal characteristics of the events that contribute to the risk. For a given scenario dataset, it can be useful to identify the scenarios that have similar behaviors (i.e., identify the most evident classes), and decide for each event sequence, to which class it belongs (i.e., classification). It is shown how it is possible to accomplish these two objectives using the Mean-Shift Methodology (MSM). The MSM is a kernel-based, non-parametric density estimation technique that is used to find the modes of an unknown data distribution. The algorithm developed finds the modes of the data distribution in the state space corresponding to regions with highest data density as well as grouping the scenarios generated into clusters based on scenario temporal similarities. The MSM is illustrated using the data generated by a DET algorithm for the analysis of a simple level/temperature controller and reactor vessel auxiliary cooling system.

Published by Elsevier Ltd.

1. Introduction

Dynamic methodologies for probabilistic risk assessment (PRA) are those that account for possible coupling between triggered or stochastic events through explicit consideration of the time element in the system evolution. They are usually needed when the system has more than one failure mode, control loops, and/or hardware/process/software/human interaction [1]. Dynamic methodologies are also capable of modeling the impacts of both the epistemic[1] and aleatory[2] uncertainties on the system figure-of-merit within a phenomenologically consistent framework.

Dynamic PRA methods include Dynamic Logical Analytical Methodology (DYLAM) [2], Dynamic Event Tree Analysis Method (DETAM) [3], ADS [4], ADAPT [5], Sequence Diagrams (ESDs) [6], Petri Nets [7], Dynamic Flowgraph Methodology (DFM) [8], Discrete Dynamic Event Trees (DDETs) [9], Markov/Cell-to-Cell Mapping Technique [10] and Monte Carlo Dynamic Event Tree (MCDET) [11]. The list is not exhaustive and only provides some samples of dynamic PRA methods. A more comprehensive discussion of dynamic methods is given in [1].

The DYLAM, DETAM, ADS and ADAPT are among methodologies that use dynamic event trees (DETs) to account for aleatory uncertainties. ADAPT can also account for epistemic uncertainties within the DET framework. A DET is an expansion on traditional static event trees (ETs), and seeks to incorporate timing and process relationships into the stochastic system model. Static ETs have a fixed and predetermined event sequence defined by the analyst, determined after a series of review processes and thermal hydraulic calculations. Fig. 1 shows a simplified ET for a large break loss of cooling accident (LOCA). In order to reach a safe state of the plant, the reactor protection system trips the reactor and performs the cooling of the reactor through the emergency cooling system (ECCS). A failure in any of these two systems will cause core damage (CD).

The DETs are generated by the direct coupling between the dynamic model of the plant and the stochastic behavior of system components (including software/firmware) and human actions. The branching conditions in a DET are generated by user specified rules, such as activation/non-activation upon demand of components, correct/faulty crew action depending on specific plant conditions or when state variables reach predefined setpoints during the simulation. Process and modeling uncertainties (which can affect the ordering of the events [5]) are also taken into account in terms of branching conditions. The dynamic model of the plant is built using system analysis codes (e.g., RELAP [12] or MELCOR [13])

* Corresponding author. Tel.: +1 614 579 5653.
E-mail addresses: diego.mandelli@inl.gov, mandelli.1@osu.edu (D. Mandelli), yilmaz.15@osu.edu (A. Yilmaz), aldemir.1@osu.edu (T. Aldemir), metzroth.1@osu.edu (K. Metzroth), denning.8@osu.edu (R. Denning).

[1] Uncertainties associated with lack of knowledge such as modeling uncertainties.

[2] Uncertainties due to stochastic variability of input and/or model parameters.

0951-8320/$ - see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.ress.2013.02.013

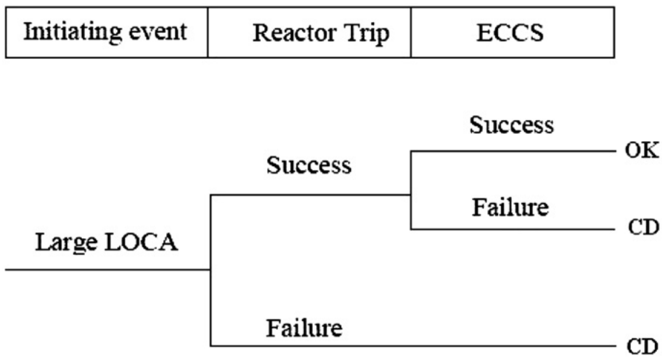

Fig. 1. Example of an event tree.

which evaluate its temporal behavior and determine the timing and nature of each branch. Subsequently, the use of DETs allows a more systematic and mechanistic search of the uncertainty space and also allows consideration of both the epistemic and aleatory uncertainties in a consistent phenomenological framework.

The major challenges in using DETs (as well as other dynamic methodologies) are the heavier computational and memory requirements compared to the classical ET analysis. This is due to the fact that each branch generated can contain time evolutions of a large number of variables (about 50,000 variables (or data channels) are present in MELCOR) and thus a much larger number of scenarios can be generated (on the order of several thousands) compared to the traditional ET/fault-tree (FT) approach. Such large amounts of information are very difficult to organize and interpret in regard to the main trends in scenario evolutions and the main risk contributors for each initiating event [14]. A solution to this problem is to partition the set of scenarios into groups, called clusters, and analyze each group individually rather than all the scenarios simultaneously. The partition is performed by identifying similarities among scenarios and grouping them according to a predefined similarity criteria. Once the partition is obtained, the user can analyze each group and identify differences among groups.

When dealing with nuclear transients, it is possible to analyze the set of scenarios in two modes:

- *End State Analysis* classifies the scenarios into clusters based on the end state of the scenarios.
- *Transient Analysis* classifies the scenarios into clusters based on time evolution of the scenarios.

While the first mode has been widely used in the classic fault tree/event tree analysis [15], the second one is starting to be considered in the recent years [16]. From a safety point of view, for example, it might be useful to group scenarios based on their temporal behavior and identify how sequence and timing of events affect the overall system dynamics other than focusing only on the end result of the simulation.

This paper presents a scenario clustering algorithm which can simplify the organization and analysis of the large dataset generated by a DET. By scenario clustering we mean two actions:

1. Identify the scenarios that have a similar behavior (i.e., identify the most evident clusters).
2. Associate each scenario with a unique cluster.

When clusters are determined, the user can then identify similarities among the scenarios in each cluster (e.g., timing and sequence of events) and compare them among different clusters. For example, clustering applied to dynamic PRA helps the user to understand how small changes in sequence/timing of events impact the overall system dynamics (see Section 4.3). A metric of success is, thus, the ability to determine a set of clusters that can help the user to identify such effects.

In Sections 2 and 3, the notions of clustering and classification are introduced along with the need and the approach to pre-processing of the raw data. In Section 4, we introduce the MSM using the dataset generated by DET for the level controller presented in [17] as an example (Section 4.2). In Section 4.3, we apply the Mean Shift Methodology (MSM) to the reactor vessel auxiliary cooling system (RVACS) of a conceptual design for a sodium-cooled fast reactor. Section 5 presents the conclusions of the study.

2. Clustering: an overview

Clustering is the process of organizing objects into groups whose members are in some way similar. A cluster is therefore a collection of objects which are similar to each other and are dissimilar to the objects belonging to the other clusters [18].

Fig. 2 shows an elementary example of partitional clustering [19] applied to a two-dimensional data. Here, we easily identify the 3 clusters into which the data can be divided. The similarity criterion is the distance measure: two or more objects belong to the same cluster if they are "close" according to a specified distance measure. The approach of using distance metrics to clustering is called distance-based clustering and will be used in this work by employing the Euclidean distance as a measure of the similarity between two D-dimensional data points $\vec{x_i}$ and $\vec{x_j}$

$$d(\vec{x_i},\vec{x_j}) = \left(\sum_{k=1}^{D} |x_{ik}-x_{jk}|^2\right)^{1/2}. \quad (1)$$

where data point $\vec{x_i}$ represents a scenario in the D-dimensional space of the data channels or variables of interest.

More formally, the concept of clustering [18] that we aim is to find a partition $\mathbf{C}=\{C_1,\dots,C_l,\dots,C_L\}$ of the set of I scenarios $\mathbf{X}=\{\vec{x_1},\dots,\vec{x_i},\dots,\vec{x_I}\}$ Each C_l $(l=1,\dots,L)$ is called a cluster. The partition $\mathbf{C}$ of $\mathbf{X}$ is given as follows:

$$\begin{cases} C_l \neq \emptyset, l=1,\dots,L \\ \bigcup_{l=1}^{L} C_l = \mathbf{X} \end{cases} \quad (2)$$

Clustering algorithms can be divided into two classes [18]:

- Hierarchical algorithms
- Partitional algorithms

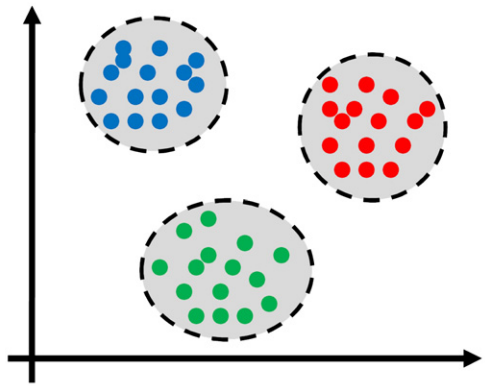

Fig. 2. Example of clustering process.