A Methodology for Generating Dynamic Accident Progression Event Trees for Level-2 PRA

Aram Hakobyan, Richard Denning, Tunc Aldemir,
The Ohio State University, Nuclear Engineering Program, 650 Ackerman Road, Columbus, Ohio 43202, Email:hakobyan.1@osu.edu

Sean Dunagan, David Kunsman
Sandia National Laboratory, Albuquerque, New Mexico, 87185

Abstract

Currently, the development and analysis of Accident Progression Event Trees (APETs) are performed in a manner that is computationally time consuming, difficult to reproduce and also can be phenomenologically inconsistent. A software tool (ADAPT) is described for automated APET generation using the concept of dynamic event trees. The tool determines the branching times from a severe accident analysis code based on user specified criteria for branching. It assigns user specified probabilities to every branch, tracks the total branch probability, and truncates branches based on the given pruning/truncation rules to avoid an unmanageable number of scenarios. While the software tool could be applied to any systems analysis code, the MELCOR code is used for this illustration. A case study is presented involving station blackout with the loss of auxiliary feedwater system for a pressurized water reactor.

KEYWORDS: dynamic event tree, uncertainty, probability, APET

1. Introduction

The Accident Progression Event Trees (APET) used in NUREG-1150 [1] were a major improvement over the containment event trees of WASH-1400 [2]. However, the NUREG-1150 APET process has some substantial drawbacks. Traditionally, APETs are static in nature and are quantified in an intuitive manner involving simplified approximations to complex physical phenomena (for example, primary system depressurization due to creep rupture of hot leg or surge line) Determination of the time sequencing of events is a critical element in generating APETs. A priori, the analyst does not know whether Event A precedes or follows Event B but must determine the ordering of events based on sensitivity calculations. Often, because of uncertainties in accident progression (either aleatory or epistemic in nature), it is possible that Event A might precede Event B under some circumstances and follow Event B under other circumstances. Thus, it is necessary to consider the occurrence of events at multiple stages of the scenario (such as hydrogen combustion). Furthermore, in order to predict the loads leading to containment failure, it is necessary to approximate the combination of forcing functions in an artificial manner. For example, if a hydrogen burn at the time of vessel melt-through could potentially result in containment failure, it is necessary to know whether a burning event occurred earlier in the accident sequence, depleting the amount of oxygen and hydrogen in the containment. Based to a large extent on judgment, increments of partial pressure in the containment are added to determine peak pressure, without a mechanistic consideration of heat losses during the pressure transient. Prior to quantification of the event tree, a number of calculations are performed with the systems code, which include a range of accident variations that provide insights to the analyst on the magnitudes of separate effects. During the quantification process, the analyst simplistically combines these approximations to estimate the forcing functions that threaten containment.

The concept of dynamic event trees in which the timing of events is determined directly by the system simulation code has been developed to address the deficiencies of static event trees. Different techniques have been developed for dynamic event tree generation. DYLAM (Dynamical Logical Methodology) [3] is a simulation driver that is able to generate branchings at user specified time intervals and to coordinate the simulation of every branch. The DYLAM approach [4] has been used to identify a number of scenarios that lead to auxiliary feedwater system failure in a pressurized water reactor (PWR) that could not be identified with the conventional event-tree/fault-tree approach. DETAM (Dynamic Event Tree Analysis Method) [5] is similar to DYLAM but explicitly addresses specific operator states and the evolution of these states over the course of a scenario, such as potential errors in decision making by the operating staff as influenced by the scenario dynamics, the crew's previous decisions, the crew's internal state (stress and confidence) and external factors (e.g., economics). The ADS (Accident Dynamic Simulator) [6] also explicitly considers operator states, but initiates branchings at times when the system or the operator takes an action (rather than at pre-specified times), accounts for possibility of repair, and maintains the plant history along branches to determine performance shaping factors for operator actions. The DDET (Discrete Dynamic Event Tree) approach presented in [7] shows how the error in the prediction of process evolution due to branching only at user specified time intervals can be quantified. DDET/MC hybrid simulation technique as described in [8] generates the possible branchings with a DDET engine and selects k number of branches to be followed out of the possible n branches by Monte Carlo sampling. The MCDET [9] focuses on Level 2 PRA using DDET/MC methodology and MELCOR [10] as the dynamics code. The DENDROS code [11] only branches when a setpoint for system intervention is crossed. The underlying assumption in DENDROS is that the likelihood of failure on demand dominates erroneous activation or deactivation of continuously operating systems, resulting in fewer branches to be followed.

An important point that needs to be taken into account in generating dynamic event trees for Level 2 PRA is that the branching rules may contain both epistemic and aleatory uncertainties (see Section 2) and not only aleatory uncertainties as it is mostly the case with the failure of active components. For example, creep rupture may not be just sensitive to the choice of the probability distribution function (pdf) to describe the stochasticity in the rupture occurrence for a given magnitude of the rupture parameter, but also to the uncertainty in the determination of the process variables that define the rupture parameter (see Section 4). This paper shows how dynamic event trees can be used to account for aleatory uncertainties in Level 2 PRA. The methodology differs from the previously proposed dynamic event tree generation schemes [3-11] in that modeling of the stochasticity in the occurrence of the events, as well as the quantification of the uncertainty in the probability assigned to this stochasticity is performed in a one-step process rather than a two-step process [7,8]. A case study is presented demonstrating the use of the approach with the MELCOR accident simulation code [4].

2. The ADAPT Methodology

In the ADAPT (Analysis of Dynamic Accident Progression Trees) methodology, the accident analysis code (for example, MELCOR [4]) simulates the accident progression path in a probabilistic context. In addition to modeling the effect of active safety features on accident progression, the proposed technique includes the probabilistic modeling of passive components, such as creep rupture of major reactor coolant system (RCS) components and overpressure failure of the containment boundary. Whenever branching criteria are satisfied that can lead to two or more possible scenarios, new branches are established and followed with parallel processing. Furthermore, when an event occurs that could threaten containment (such as a hydrogen combustion event), a mechanistic analysis of the event is performed by the systems code for risk quantification rather than the non-mechanistic combining of approximations that occurs in the quantification of static event trees.

The branch probabilities are tracked through the tree using Boolean algebra. To avoid numerical catastrophe due to enormous number of branch executions, it is necessary to truncate branches based on user defined truncation rules, such as truncating an execution when a branch probability falls below a given limit.

A set of pdfs is developed prior to the analysis to enable the probabilistic treatment of uncertainties in the modeling of severe accident phenomena such as creep rupture of RCS components, containment overpressure failure, hydrogen burning in the containment, reactor coolant pump (RCP) seal leakage, power recovery, and pressurizer/safety relief valve opening/closing upon demand. The corresponding cumulative distribution functions (Cdfs) are discretized to define the branching points. Branching occurs at the values of the physical parameter associated with these selected values of Cdf for failure. The significant advantage of this approach is that aleatory uncertainty quantification is performed simultaneously with the modeling of the stochasticity in the process. Aleatory uncertainties are those that have a stochastic nature and are often associated with the failure properties of components. Normally, aleatory uncertainty quantification is a two-step process, first associating a probability with the stochasticity of an event, and then using Monte Carlo sampling over specified Cdfs for the choice of the probability [7,8] In Level 2 PRA it is also often difficult to distinguish between epistemic and aleatory uncertainties. Epistemic uncertainties are associated with the mathematical models used in the simulation codes and lack of knowledge about certain physical phenomena taking place during severe accidents. The process physics itself may have a stochastic nature, such as the effect of random flaws on the rupture of a pipe as a function of pressure and temperature. The proposed approach allows separation of epistemic and aleatory uncertainty quantification as well. For example, the impact of uncertainties in heat transfer properties can be evaluated by sampling over the relevant model inputs and generating a dynamic event tree for each input set.

Figure 1 illustrates a branching procedure for the passive failure of a PWR component by creep rupture. When the creep rupture parameter R reaches the values 0.518, 0.764, 1.00, 1.31

and 1.931 there is, respectively, 5%, 25%, 50%, 75% and 95% probability of rupture of the component. The creep rupture parameter is defined as

$$R = \int_0^{t_f} \frac{dt}{t_R(T, m_p \sigma)} \qquad (1)$$

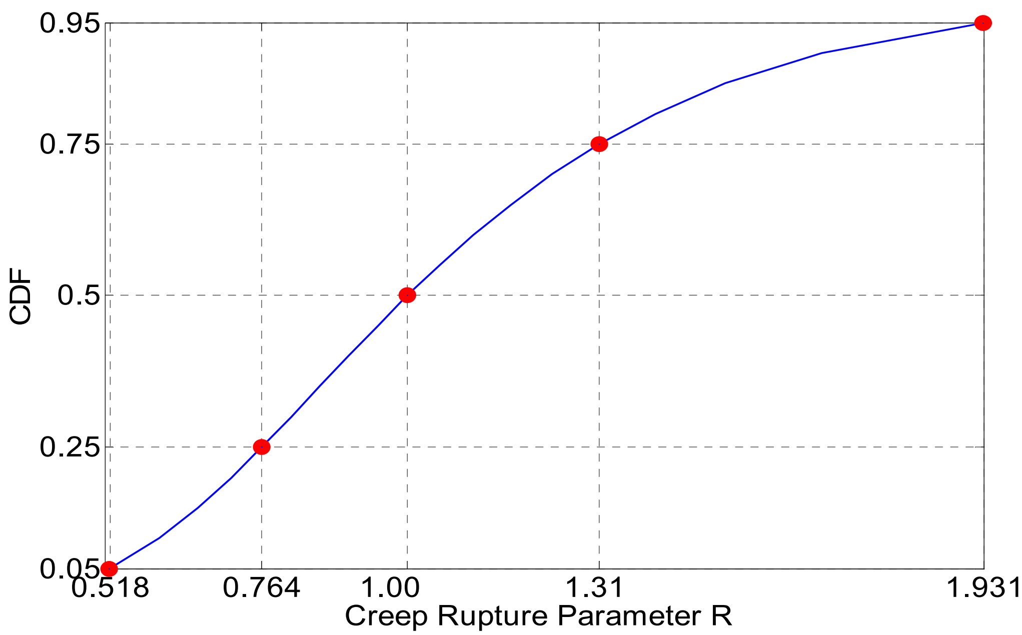
where

t_f = creep rupture failure time (sec)
t_R = time to rupture as a function of T, σ (sec)
m_p = intensity factor (unitless)

Failure is predicted to occur when the creep rupture parameter is unity. The creep rupture correlations are developed by fitting the results of experimental data. The associated differences between the data and the correlation are in part the result of experimental error, inadequacy of the form of the correlation, and property variability.

The denominator t_R in (1) is given by the Larsen-Miller correlation [13] and calculated by the accident simulation code. In Fig.1, the Cdf is determined under the assumption that probability of rupture is normally distributed with mean 1 and standard deviation 0.4 (see Section 4).

Figure 1: The Cdf for rupture for surge line, hot leg, and steam generator tubes



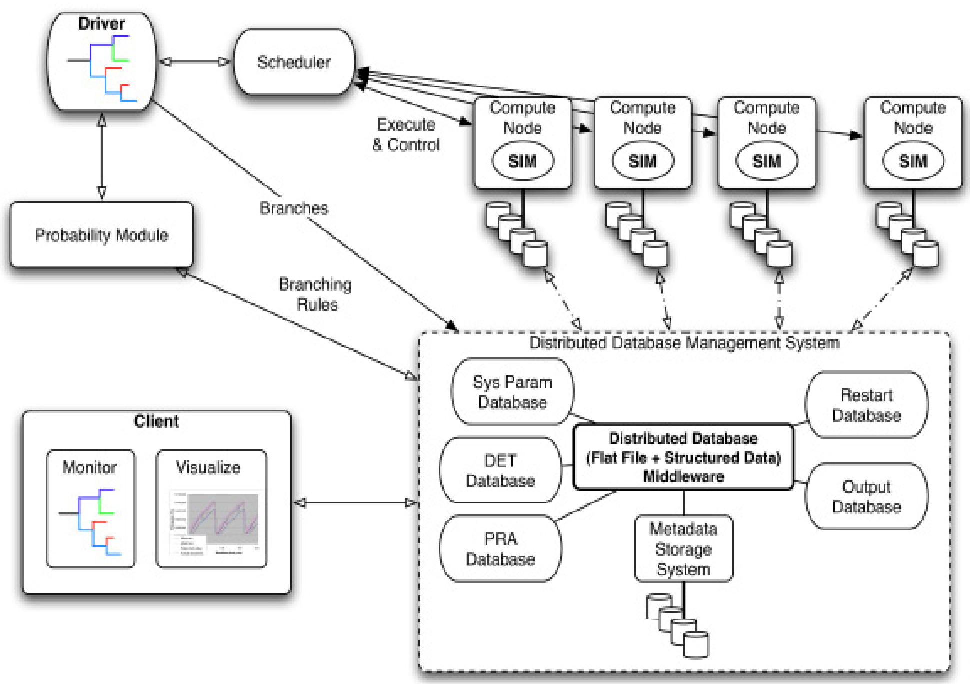
3. Computational Infrastructure

The APET generation is managed by a master controller that determines when branching is to occur, initiates multiple restarts of system code analyses, determines the probabilities of scenarios, determines when a scenario can be terminated (e.g. the containment has failed or the importance of a specific scenario has fallen below a user specified cutoff), and combines similar scenarios to reduce the scope of the analysis.

As with all dynamic event tree generation schemes, a plant simulator (e.g. MELCOR accident simulation code [9]) is used to follow the transient along each branch. A branch is pruned when its probability of occurrence falls below a user-specified value. Some significant features of the computational scheme are the following: 1) It is designed for a distributed computing environment; the scheduler can track multiple branches simultaneously; 2) The scheduler is modularized so that the branching strategy can be modified; 3) An independent database system stores data from the simulation tasks and the DET (Dynamic Event Tree) structure so that the event tree can be constructed later.

A schematic overview of the proposed infrastructure is shown in Figure 2 [12]. Following

Figure 2: A schematic overview of computational infrastructure [12]



an initiating event (or at any user-specified starting time point during an accident progression), the *Distributed Database Management System* provides initiating conditions as well as the duration of the simulation (time parameters) to the *Plant Simulator* SIM. The *Driver* runs the Plant Simulator until a stopping condition is reached. The *Scheduler* decides whether to branch or not depending on the information received from: a) the Plant Simulator on setpoint crossing or equipment demand in general, and, b) the *Probability Module* on the branch probability. The *PRA Database* contains data to quantify the likelihood of branches generated upon crossing setpoints. The database can consist of the minimum cut sets for the Top Events relevant to the branch in the form of binary decision diagrams for fast pre-processing [11] or simply contain probabilities based on operational failure data. The branching probabilities (possibly obtained through preprocessing) are passed on to the Probability Module. If branching is initiated, the Scheduler then spawns a process to follow the branch. If the Scheduler receives other demands on equipment from the Plant Simulator while this process is running and decides on branching using the criteria above, then it can spawn as many processes as needed to follow the subsequent branches. The resulting tree structure, branch probabilities, and simulation results are also sent to the Distributed Database Management System for possible post-processing and/or load distribution in a distributed computing environment.

The interface to the plant simulator (e.g. MELCOR) is abstracted to allow use of different plant simulators with possibly different computational models. The plant simulator needs to interface with the runtime system in two places: 1) during execution for task branching and migration, and, 2) before and after execution, to load and store its state and results. The developed driver communicates with the distributed database system to retrieve and store the necessary input and output files needed by the plant simulator. In other words, the driver stages the necessary input files prior to execution of the plant simulator, and after completion of the execution it stores the output files generated by the plant simulator on the distributed database system. The storage nodes of the distributed database system do not need to be different than the compute nodes.

4. Case Study - SGTR Bypass in Station Blackout Accident

In pressurized water reactors, containment bypass through steam generator tube rupture (SGTR) could represent a major risk of a large, early release of radioactive material to the environment during station blackout scenarios with failure of the Auxiliary Feedwater System. The consequences of this accident scenario are very sensitive to whether steam generator tubes fail (Mode 1) prior to the failure of the hot leg (Mode 2) or surge line (Mode 3). Mode 2 and 3 failures would result in depressurization of the reactor coolant system (RCS) and preclude the potential large early release of radionuclides to the environment associated with steam generator tube failure. To model the creep rupture events, Larson-Miller correlation [13] is used for the calculation of time to rupture t_R used to evaluate the creep rupture parameter as defined in (1).
For SG tubes (Inconel 600), it is given by

$$t_R = 10^{(\frac{p}{T}-15)} \qquad (2)$$

$$p = -11333 \log\sigma + 43333$$
(3)

For hot leg/surge line (material: SS316):

$$t_R = 10^{(\frac{p}{T}-20)} \qquad (5)$$

$$p = -13320 \log\sigma + 54870$$

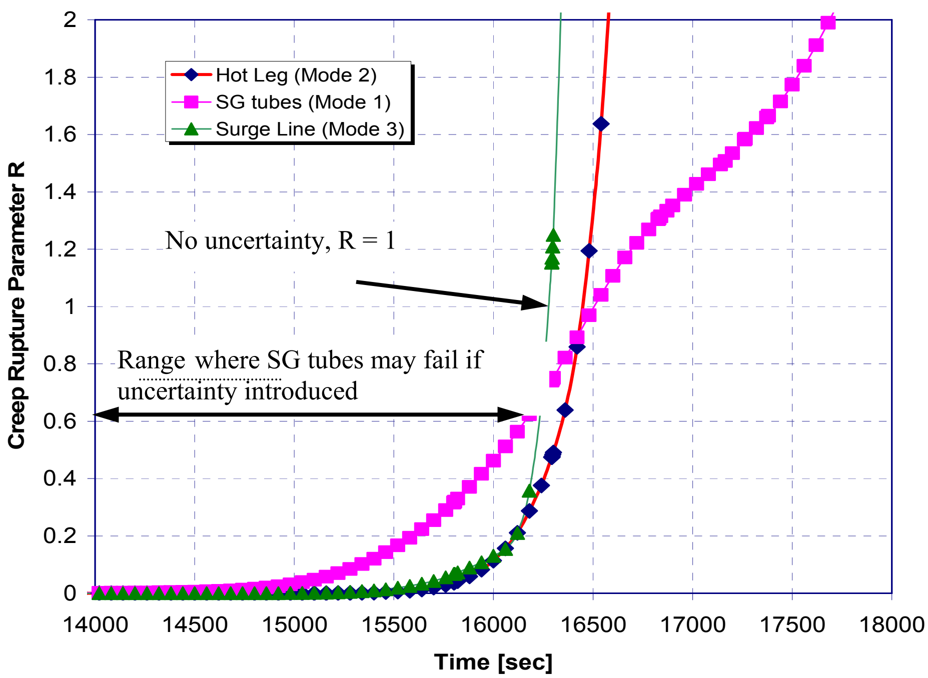
(4)

p = pressure inside the pipe (kPa),
T = temperature of the structure (K),
σ = mechanical stress in structure (log σ given in kPa).

The severe accident analysis was performed with MELCOR [9]. Figure 3 shows that for the case analyzed, the surge line reaches the critical value R = 1 prior to the hot leg or steam generator tubes and would fail first, as also reported in [14]. However, as with any correlation, the Larson-Miller correlation has an associated uncertainty that propagates into R. Note that for R values up to about 0.6, the steam generator tubes are closer to failure than the hot leg or surge line.

To account for the uncertainty at which failure would occur, a Cdf ($\Phi(R)$) was developed in the form of a lognormal distribution with a mean value of $\mu = 1$ and standard deviation of $\sigma = 0.4$ based on assessment of the dispersion in the supporting experimental data [13].

Figure 3: Creep rupture parameter *R* for surge line, hot leg, and SG tubes



$$\Phi(R) = \int_0^R \frac{exp\left[-\frac{1}{2}\left(\frac{ln(R')}{0.4}\right)^2\right]}{0.4R'\sqrt{2\pi}} dR' \tag{6}$$

The $\Phi(R)$ in (6) is also called a *fragility curve*. A 5-point discretization of the fragility curve at 5%, 25%, 50%, 75%, and 95%, such as shown in Fig. 1, provided corresponding R values as branching points. MELCOR execution stops when an R value for one of these three modes of failure reaches 0.518 corresponding to 5% probability of failure (see Fig.1). Two branches are generated: a scenario in which failure has occurred with a probability of 5% and a scenario in which failure has not occurred with a probability of 95%. For the unfailed branch, the simulation proceeds until the R value for one of the other failure modes reaches 0.518 or the R value for the mode in which failure is first observed reaches the second branching point of

0.764. The stopping and branching process continues until all five discrete points of the fragility curve for at least one of the three modes of failure are exhausted, or the simulation reaches the user-specified end point.

4. Conclusion

Many potential accident scenarios are ignored in current conventional Level 2 PRA analyses because of the static nature of the conventional event-tree/fault-tree analysis. A wider variety of accident scenarios is generated and quantified with ADAPT. Although in the deterministic treatment of creep rupture ($R = 1$), steam generator tube failure would be precluded by the earlier failure of the surge line, this paper shows that by considering model uncertainties, non-zero probabilities can be assigned to each of the modes of RCS failure. The ADAPT methodology also proposes a solution for a combined treatment of aleatory and epistemic uncertainties as an integral part of the Level 2 analysis using dynamic accident progression event trees.

In summary, an improved method has been developed for the construction and analysis of APETs. The ADAPT methodology provides a more reproducible and phenomenologically consistent approach to APET construction and analysis. It also substantially reduces the man-power requirements associated with the performance of multiple sensitivity studies. The ADAPT approach also allows: a) separation of the epistemic and aleatory uncertainties in passive safety system performance to be accounted for in assessing failure probability, as required in assessing the risk from passively-safe water reactor and GEN-IV designs, and, b) accounting for aleatory uncertainties simultaneously with dynamic event tree generation.

Acknowledgements

The research presented in this paper was partially supported by a contract from the Sandia National Laboratory (SNL). The information and conclusions presented here in are those of the authors and do not necessary represent the views or positions of the SNL. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assume any legal liability or responsibility for any third party's use of this information.

References

1. Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants, NUREG -1150, 1990
2. Reactor Safety Study – An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, WASH-1400 (NUREG-75/014), 1975
3. G. COJAZZI, "The DYLAM approach to the dynamic reliability analysis of systems", R*eliability Engineering & System Safety*, **52,** 279-296 (1996).
4. P. C. CACCIABUE, A. AMENDOLA, G. COJAZZI, "Dynamic logical analytical methodology versus fault tree: The case of auxiliary feedwater system of a nuclear power plant", *Nuclear Technology*., **74,** 195-208 (1986)

5. C. ACOSTA, N. SIU, “Dynamic Event Trees in Accident Sequence Analysis: Application to Steam Generator Tube Rupture”, *Reliability Engineering and System Safety*, **41,**135-154 (1993)
6. H. KAE-SHENG, A. MOSLEH, “The Development and Application of the Accident Dynamic Simulator for Dynamic Probabilistic Risk Assessment of Nuclear Power Plants”, *Reliability Engineering & System Safety*, **52,** 297-314 (1996)
7. C. SMIDTS, S. SWAMINATHAN “Improvements to Discrete Dynamic Methodologies”, PSA-96, 159-166, American Nuclear Society (1996)
8. S. MARCHAND, B. TOMBUYSES, P. LABEAU “DDET and Monte Carlo Simulation to Solve Some Dynamic Reliability Problems”, *PSAM 4*, 2055-2060, Springer Verlag, New York, (1998)
9. R. M. SUMMERS, R. K. COLE, E. A. BOUCHERON, M. K. KARMEL, S. E. DINGMAN, J. E. KELLY, “MELCOR 1.8.0: A Computer Code for Nuclear Reactor Severe Accident Source Term and Risk Assessment Analyses”, Nuclear Regulatory Commission, Washington, DC (1991)
10. E. HOFER, M. KLOOS, B. KRZYKACZ-HAUSMANN, J. PESCHKE, M. SONNENKALB, “Dynamic Event Trees for Probabilistic Safety Analysis”, GRS, Garsching, Germany (2004)
11. R. MUNOZ, E. MINGUEZ, “A Second Generation Scheduler for Dynamic Event Trees”, *Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, pp. 1358-1367, Madrid, Spain (1999)
12. B. RUTT, U. CATALYUREK, A. HAKOBYAN, K. METZROTH, T. ALDEMIR, R. DENNING, S. DUNAGAN, D. KUNSMAN, “*Distributed Dynamic Event Tree Generation for Reliability and Risk Assessment*”, CLADE 2006 Workshop, Paris, France, June 2006 (accepted for publication)
13. S. MAJUMDAR, “Prediction of Structural Integrity of Steam Generator Tubes Under Severe Accident Conditions”, *Nuclear Engineering and Design, pp. 31-55* (1999)
14. Y. LIAO, K. VIEROW, “MELCOR Analysis of Steam Generator Tube Creep Rupture in Station Blackout Severe Accident”, *Nuclear Technology*, *pp. 302-313* (2005)