Nuclear Engineering and Design 238 (2008) 3457–3467

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

Dynamic generation of accident progression event trees

Aram Hakobyan[a,*], Tunc Aldemir[a], Richard Denning[a], Sean Dunagan[b], David Kunsman[b], Benjamin Rutt[c], Umit Catalyurek[c]

[a] The Ohio State University, Nuclear Engineering Program, 201 West 19th Avenue, Columbus, OH 43210, United States
[b] Sandia National Laboratories, Albuquerque, NM 87185, United States[1]
[c] The Ohio State University, Department of Biomedical Informatics, 333 West 10th Avenue, Columbus, OH 43210, United States

ARTICLE INFO

Article history:
Received 16 May 2007
Received in revised form 15 July 2008
Accepted 6 August 2008

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 is presented 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.

1. Introduction

The objective of probabilistic risk assessment (PRA) is to make an informed assessment of the frequencies and magnitudes of accident scenarios as they impact the population and the environment. The accident progression event tree (APET) approach used in NUREG-1150 (USNRC, 1990) to quantify accident progression and containment response was a major improvement over the containment event trees of WASH-1400 (USNRC, 1975). APETs are used as a conventional tool for Level 2 and 3 risk assessments. An APET identifies the variety of ways in which containment failure or bypass can occur, as well as the various severe accident processes that affect the mode of failure, timing of failure, and magnitude of release of radioactive material to the environment. An APET describes the evolution of the accident in terms of possible branchings of the processes involved due to the variability of severe accident processes (aleatory[2] uncertainties) and the lack of knowledge of severe accident processes (epistemic[3] uncertainties). Although we treat epistemic and aleatory uncertainties as distinct concepts, state of knowledge can depend on the depth to which an analyst analyzes an event and hence it may be difficult to distinguish between epistemic and aleatory uncertainties.

The NUREG-1150 APET process is static in nature and quantified in an intuitive manner involving relatively simplified approximations to complex physical phenomena, such as the sequence of events leading to primary system depressurization due to creep rupture of hot leg or surge line. 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, or sometimes make assumptions. 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. For example, at different stages of the accident progression a question is asked whether hydrogen combustion occurs or not. Furthermore, in order to predict the loads leading to containment failure, it is necessary to approximate the combination of loads 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 or not 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 mechanistic consideration of heat losses during the pressure transient (USNRC, 1990). Prior to quantification of the APET, a number of calculations are

* Corresponding author.
E-mail address: hakobyan.1@osu.edu (A. Hakobyan).

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

[2] Aleatory uncertainties are those originating from the stochastic nature of events, such us the direction of the wind at the time that an accident occurs.

[3] Epistemic uncertainties are those originating from imprecise knowledge, such as the magnitude of the two-phase multiplier used to determine the frictional pressure drop in a pipe in a two-phase flow regime.

0029-5493/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.nucengdes.2008.08.005

performed with the accident simulation computer code (MELCOR in the case of this study), which include a range of accident variations that provide insights to the analyst on the magnitudes of separate effects. Normally, fault tree analysis is not used in estimating branching probabilities for APETs. Branching probabilities are typically determined for APETs by comparing physical conditions obtained in the severe accident scenario with branching criteria (USNRC, 1990). During the quantification process, the analyst combines these approximations using engineering judgment to estimate the loads that threaten the containment. Because of the possible use of such engineering judgment, the conventional APETs do not necessarily treat event timing in a physically consistent manner.

Conventional PRA techniques for APET generation may involve hundreds of manual simulations of severe accident codes (USNRC, 1990). This process requires significant amount of workforce involved in running different accident scenarios. Thus, the overall process of APET generation for a single initiating event may take a year or more in some cases. Also, simulation of accident scenarios by manually changing a portion of the input data for each new scenario may introduce inadvertent errors in the input deck with possible significant impact on the results. These types of errors are very hard to identify in the post-analysis of an enormous output database, thus making the overall time of the analysis even longer.

The objectives of this paper are to:

- present a software tool for automated generation of APETs (Section 3) using the dynamic event tree (DET) methodology (Section 2) which is independent of the severe accident systems analysis computer code being used, and,
- illustrate how the tool allows systematic quantification of the impact of the epistemic and aleatory uncertainties on the consequences of a given initiating event (Section 4).

2. The DET methodology

There are different interpretations to the word “dynamic” when used along with PRA. One use of the term dynamic PRA or “living PRA” is to describe periodic updates of the PRA to reflect any changes in the plant configuration (Sancaktar and Sharp, 1985). Another use is when the PRA model is updated to account for equipment aging (Vesely, 1991). The third use is to describe an approach that includes explicit modeling of deterministic dynamic processes that take place during plant system evolution along with stochastic modeling (Marchand et al., 1998; Smidts and Swaminathan, 1996; Acosta and Siu, 1993; Hsueh and Mosleh, 1996; Cacciabue et al., 1986; Cojazzi et al., 1994, 1996; Munoz et al., 1999a,b; Hofer et al., 2004). In this third use, plant parameters are represented as time-dependent variables in event tree-construction with branching times often determined from the severe accident systems analysis code being used to examine the plant. It is this last definition of dynamic PRA that is used within the context of this paper.

In dynamic PRA analysis, event tree scenarios are run simultaneously starting from a single initiating event. The branchings occur at user specified times and/or when an action is required by the system and/or the operator, thus creating a sequence of events based on the time of their occurrence. For example, every time a system parameter exceeds a threshold/setpoint, branching takes place based on the possible outcomes of the system/component response. These outcomes then decide how the dynamic system variables will evolve in time for each branch. Since two different outcomes at a branching may lead to completely different paths for system evolution, the next branching for these paths may occur not only in different times, but also based on different branching criteria. The main advantage of DET methodology over the conventional event tree method is that it simulates probabilistic system evolution in a manner consistent with the severe accident model. For example, in a severe accident scenario, hydrogen combustion can occur at a variety of times during the accident. At each hydrogen combustion event a threat exists for containment failure. The inventory of hydrogen and oxygen in the containment is also affected. In the NUREG-1150 methodology (USNRC, 1990), large event trees were developed to accommodate the multiplicity of event combinations. Complex approximate algorithms were developed to assess the magnitude of the challenge to containment for each event involving the addition of partial pressures of gases that varied with the relative timing of events. In contrast, each ADAPT scenario thread is consistent with the phenomenological model. It would be completely impractical to attempt to run such a variety of sequences manually.

Software development for DET generation began in mid-1980s. A variety of tools and techniques have been proposed. The research work has modeled the response of both the plant systems and plant operators to an initiating event that propagates into an accident. Several institutions have been involved in developing DET generation methodologies both in the United States (Acosta and Siu, 1993; Hsueh and Mosleh, 1996) and Europe (Cacciabue et al., 1986; Cojazzi et al., 1994, 1996; Munoz et al., 1999a,b; Hofer et al., 2004).

In the mid-1980s, researchers at the Joint European Center at Ispra, Italy, developed a methodology for dynamic reliability analysis called dynamic logical analytical methodology (DYLAM) (Cacciabue et al., 1986; Cojazzi et al., 1994, 1996). The basic idea of the DYLAM methodology is to provide a tool for coupling the probabilistic and physical behavior of a system for more detailed reliability analysis. All the knowledge about the physical system under study is contained in the system simulator. The active components of the system are allowed to have different states such as nominal, failed on, failed off and stuck. Once the simulator is linked to the DYLAM code, DYLAM drives the simulation by assigning initial states to each branch and triggering stochastic transitions in the component states, taking into account the time history of the logical states of components if necessary (e.g. for operator modeling). For each path (or branch), the (possibly time-dependent) probability of the system achieving that branch is evaluated from the user-provided branching probabilities. The probability of occurrence of a given consequence (or Top Event) is the sum of the probabilities of all the branches leading to that Top Event (Cojazzi et al., 1996). Each system component/operator is characterized by discrete states with different options to model transitions between these states, such as stochastic transitions with constant probabilities, functionally dependent transitions, stochastic and functionally dependent transitions, conditional probabilities, and stochastic transitions with variable transition rates. The time points at which the transitions (either on demand or stochastic) take place correspond to the branching points. The DYLAM approach has been used to perform dynamic reliability analysis not only in nuclear, but also in chemical, aeronautical, and other industries.

In 1992, Acosta and Siu proposed a variant of DYLAM for Level 1 PRA called dynamic event tree analysis method (DETAM), to analyze the risk associated with nuclear power plant accident sequences (Acosta and Siu, 1993). DETAM provided a framework for treating stochastic variations in operating crew states, as well as in hardware states. The plant process variables used to determine the likelihood of stochastic branchings were calculated from a system simulator. The branchings were allowed to occur at user-specified fixed points in time. In case of hardware-related branchings, the system unavailabilities were modeled as demand failure frequencies. In the cases of diagnosis state and planning state transitions, mainly expert judgment was used to assign probabilities/frequencies.

In 1993, Hsueh and Mosleh developed the accident dynamic simulation methodology (ADS) (Hsueh and Mosleh, 1996; Zhu et al., 2008). The modeling strategy of ADS is based on breaking down the accident analysis model into different parts according to the nature of the processes involved, simplifying each part while retaining its essential features, and developing integration rules for full scale application. Whenever a hardware system state transition point or an operator interaction point is reached, the accident scheduler chooses one path to follow. After the simulation process reaches an end point, the scheduler directs the simulation back to the previous branch point, reinitializes every simulation module back to this time point, and follows the other branch point path. In the multiprocessor version of ADS (Zhu et al., 2008), the simulations are distributed among multiple client computers. The associated client scheduler module stores the system data and transfers state data and branching information to the server when branching points are activated during a simulation. The server then allocates a new simulation task for each new branch event. After the clients perform their assigned simulation tasks, the individual simulation task results are reassembled into a larger solution for the entire simulation. A central server is responsible for managing assignment of simulation tasks to individual clients and post-simulation reassembly of the simulation results.

Another tool for DET generation developed in 1999 is dynamic event network distributed risk-oriented scheduler (DENDROS) (Munoz et al., 1999a,b). The DENDROS was developed mainly to model response of safety features to a transient for Level 1 PRA and is a discrete event processor, managing messages coming from different calculation modules including the physical system simulator and decision processes. It is designed for a distributed computing environment using a network of processors exchanging information through an independent channel. During a simulation, the scheduler makes a decision about the need to create new processes if a setpoint is crossed (branching point), to change the already running processes to stand-by state for later reuse, or even to force some non-active ones to terminate based on the end conditions, such as probability falling below a user-specified cutoff value. The DENDROS was linked to the pressurized water reactor simulator transient response and test analyzer (TRETA).

In 2002, researchers from GRS,[4] Germany developed a DET method combined with Monte Carlo simulation called Monte Carlo dynamic event tree (MCDET) (Hofer et al., 2004). The MCDET considers all combinations of two characteristics of a transition: "when" and "where to". Discrete and random "when" and/or "where to" are taken into account by DET analysis, while continuous and random ones were handled by Monte Carlo simulation. The MCDET was implemented as a stochastic module that could be operated in tandem with any deterministic dynamics code. For each element of Monte Carlo sample, MCDET generates a discrete DET using the system code and computes the time histories of all system variables along each path together with the path probability. The mean conditional probability distribution (conditional on the initiating event and the values of randomly sampled aleatory uncertainties) over all trees in the sample is the final result. To keep the computational effort practicable, a probabilistic "cutoff" criterion was introduced that would allow to terminate any branches with a probability below that cutoff value. For practical application, the MCDET was linked with severe accident analysis code MELCOR (Summers et al., 1981). The focus was on the modeling of the response of the safety features of the plant and the reaction of the operating crew during severe accident progression.

[4] Gesellschaft für Anlagen und Reacktorsicherheit.

3. THE ADAPT approach

Like all the other DET generation techniques overviewed in Section 2, the philosophy of the analysis of dynamic accident progression trees (ADAPT) approach is to let a system code (simulator) determine the pathway of the scenario within a probabilistic context. When conditions are achieved that would lead to alternative accident pathways, a driver generates new scenario threads (branches) for parallel processing. The branch probabilities are tracked through the tree using Boolean algebra. To avoid unacceptable growth of the problem due to an enormous number of branch executions, it is necessary to terminate branches based on user defined truncation rules, such as truncating an execution when a branch probability falls below a given limit or when the user specified simulation time is exceeded. The truncation level must be set small enough that the associated impact on the probability of key events is negligible.

In principle, all the DET approaches proposed to date are similar in that they explore possible ways the dynamic system under consideration can evolve and quantify the likelihood of these scenarios, based on user specified:

- branching and stopping rules,
- system simulator, and,
- probability assignment rules to scenarios.

In that respect, their contribution is mainly reflected in how the above information is provided and used, including the specific algorithms utilized. Regarding its contribution to the state-of-the art, ADAPT combines the active component modeling approach (i.e. only considering failures upon demand) and parallel processing capability of DENDROS (Munoz et al., 1999a,b) with passive component handling capability of MCDET (Hofer et al., 2004) to facilitate Level 2 PRA. However, it differs from MCDET in the way uncertainties are handled. As indicated in Section 2, MCDET first divides the set of stochastic variables (which it regards as aleatory uncertainties) into two subsets of discrete (V_d) and continuous (V_s) variables. Then it selects an element $v_s \in V_s$ using Monte Carlo sampling from V_s and runs the simulator with v_s for all elements of V_d (considered as paths of an event tree). ADAPT also regards the variables associated with the stochasticity in the active (e.g. valves, pumps) and passive (e.g. pipes, steam generator tubes, containment) component behavior and other severe accident phenomena (e.g. hydrogen combustion) as aleatory uncertainties. Uncertainties associated with simulator inputs (e.g. heat transfer coefficients, friction coefficients, nodalization) are regarded as epistemic. The rationale behind this distinction is that the conditions under which the branching would occur for the uncertainties designated as aleatory are internally determined by the simulator without user control. This is consistent with the irreducible nature of aleatory uncertainties. On the other hand, the user has control over the uncertainties designated as epistemic and can reduce them with improved knowledge of the phenomena involved. However, as indicated in Section 1, state of knowledge can depend on the depth to which an analyst analyzes an event and hence it may be difficult to distinguish between epistemic and aleatory uncertainties. For active components, the ADAPT approach is similar to that used by DENDROS in that the timing of the branch initiation is determined by the simulator based on the computed magnitude of the process variables (e.g. pressure, temperature, level) and the control laws, as well as possible failure modes of the component. For example, the time at which a demand will be placed on a safety relief valve to open and close will be determined by the simulator based upon the computed pressure and valve setpoint. The valve may open and close in response to the setpoint pressure but may also fail to

Table 1
Severe accident phenomena modeled stochastically by ADAPT.

Creep rupture of major RCS components
Pressurizer surge line
Hot leg
Steam generator tubes
Hydrogen combustion in the containment
Containment overpressure failure
Failure of pressurizer relief valve to close
Power recovery (for station blackout accidents)

close on demand. At this point in time, ADAPT generates a branching point with two (or more) possible scenarios to be followed by the simulator. The main difference is that branching probabilities are external inputs in the current version of ADAPT whereas DENDROS infers them from fault trees in the form of binary decision diagrams. In the case of passive component behavior and other stochastic phenomena, ADAPT uses an approach similar to Latin hypercube sampling (LHS) from the cumulative distribution function (CDF) of the dynamic variables relevant to the components and phenomena under consideration (Section 3.1). The ADAPT approach to the stochastic modeling of passive components and severe accident phenomena allows reusable scenario information so that if the CDFs used to initiate the branches are changed, the simulations do not have to be repeated (Section 3.1). ADAPT also requires little effort on the part of the user for coupling of the driver to the simulator. Section 3.2 presents an overview of the computational infrastructure of ADAPT. The current version of ADAPT requires that a plant simulator have the following four features: (1) reads its input from command-line and/or text file, (2) has check-pointing feature, (3) allow user-defined control-functions (e.g. stopping if a certain condition is true), (4) output can be utilized to detect stopping condition. ADAPT driver has been designed to take advantage of these features. It is implemented with easy-to-customize modules such as processing output files, and modifying input files. When integrating a new simulator, the main responsibility of the user is to put in place the user-defined control-functions for their simulator.

3.1. Stochastic modeling of passive components and severe accident phenomena

Unlike the deterministic models of phenomena used in the accident simulation codes that initiate the physical processes when specified thermal–hydraulic conditions are reached, stochastic models assume that there is potential for phenomena to occur in a wide range of relevant thermodynamic parameters (in contrast to fixed values), but with different likelihoods. Thus, instead of using fixed criteria of occurrence, probabilistic models associate probability distribution functions (PDFs) with the occurrence criteria, normally assuming that the mean values of those distributions lie at the respective points used in the deterministic models. Table 1 provides some severe accident phenomena that can be modeled stochastically using ADAPT. A set of PDFs is developed prior to the analysis to enable the probabilistic treatment of uncertainties in the modeling of severe accident phenomena shown in Table 1. 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 the CDF for failure.

Creep rupture of surge line, hot leg, and steam generator tubes is chosen as an example to illustrate this process. In PWRs, containment bypass through steam generator tube rupture (SGTR) could represent a large, early release of radioactive material to the environment during station blackout scenarios with failure of the auxiliary feedwater system (AFWS). 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. This problem was recently addressed by Vierow et al. (2003) in a deterministic manner using the MELCOR code (Summers et al., 1981). In deterministic modeling of creep rupture events for severe accident analysis with MELCOR, the criterion for creep rupture to occur is given by the Larson–Miller correlation:

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

where t_f is the creep rupture failure time (s), m_p is intensity factor; for this study it is assumed that there are no substantial flaws in the hotleg, surge line and SG tubes and that $m_p = 1$, and σ is the mechanical stress in structure (kPa).

The denominator t_R in Eq. (1) is the time at which creep rupture will occur given that the structure is held at temperature T under stress condition $m_p\sigma$. A functional form for t_R is given by the Larsen–Miller correlation (Majumdar, 1999) and calculated by MELCOR. It is effectively used to describe the integrated creep damage experienced for the time a structure is exposed to temperature T. In this study, the correlation is used within the applicable temperature range. However, there is epistemic uncertainty regarding the accuracy of the correlation to transient conditions that is not addressed in this study.

For Inconel 600 SG tubes

$$t_R = 10^{((p/T)-15)} \tag{2}$$

$$p = -11333 \log \sigma + 43333 \tag{3}$$

and for SS316 hot leg/surge line

$$t_R = 10^{((p/T)-20)} \tag{4}$$

$$p = -13320 \log \sigma + 54870 \tag{5}$$

where p is the pressure inside the pipe (kPa) and T is the temperature of the structure (K).

For both the SG tubes and hot leg/surge line, m_p is assumed to be unity. Normally, MELCOR initiates creep rupture in a node when the creep rupture parameter

$$R = \int_0^{t_f} \frac{dt}{t_R(T, m_p\sigma)} \tag{6}$$

for the node is $R = 1$. Fig. 1 shows the MELCOR results following a station blackout. For the case analyzed, Fig. 1 shows that the surge

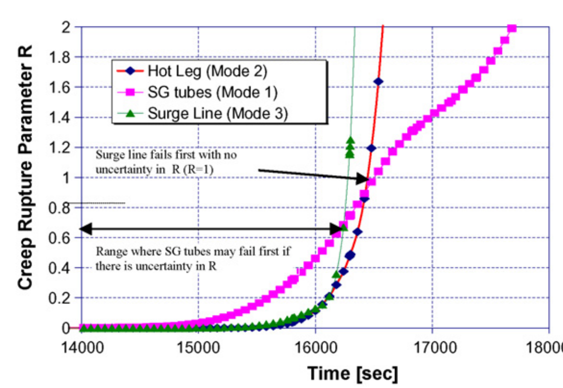

Fig. 1. Creep rupture parameter R for surge line, hot leg, and SG tubes.

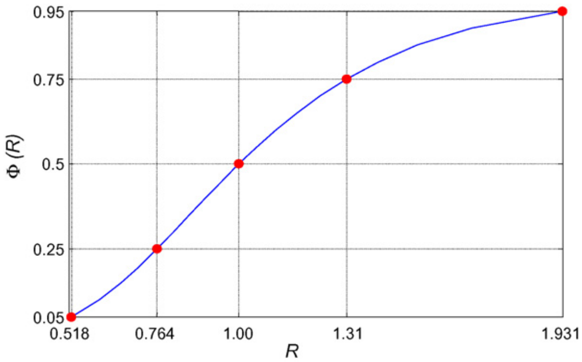

Fig. 2. The CDF for rupture for surge line, hot leg, and steam generator tubes.

line reaches the critical value $R=1$ prior to the hotleg or steam generator tubes and would fail first, as also reported by Vierow et al. (2003). However, since the uncertainty associated with the empirical Larson–Miller correlation propagates into R, there is also uncertainty in Eq. (6) results which may change the order of failure modes. Note that for R values up to about 0.6, steam generator tubes are closer to failure than the hot leg or surge line.

For quantification of the uncertainty at which failure would occur, the data were examined on which the correlation is based (Majumdar, 1999). The reasons for the spread of data around the fitted curve of failure time versus temperature are not identified but creep rupture is obviously very sensitive to the temperature environment and variations in material condition (such as the existence of flaws). A CDF $\Phi(R)$ was developed in the form of a lognormal distribution with a mean value of $\mu=1$ and standard deviation of 0.4 based on our assessment of the dispersion in the data:

$$\Phi(R)=\int_0^R \frac{\exp[-(1/2)((\ln(R'))/0.4)^2]}{0.4R'\sqrt{2\pi}}\,\mathrm{d}R' \qquad (7)$$

The $\Phi(R)$ is also called a *fragility curve* and yields the probability that the rupture parameter is below the value R. Fig. 2 illustrates the ADAPT branching procedure to determine creep rupture of major RCS components (surge line, hot leg, and SG tubes) through Eq. (6). A 5-point discretization of the fragility curve at 5%, 25%, 50%, 75% and 95%, provides corresponding R values 0.518, 0.764, 1.00, 1.31 and 1.931 as branching points. While the discretization scheme was used in the actual ADAPT runs, the choice of the percentages is for illustrative purposes only with no specific technical significance. When the creep rupture parameter R reaches each of these values, ADAPT initiates branches with 5%, 25%, 50%, 75% and 95% probability of rupture and with 95%, 75%, 50%, 25% and 5% probability of non-rupture of the component, respectively. For example, for the first branch initiation for the surge line, MELCOR execution is stopped through MELCOR control functions (part of MELCOR input) when $R=0.518$ corresponding to 5% probability of creep rupture for the surge line. At this point, ADAPT generates two branches (scenarios): (i) a scenario in which surge line has ruptured with a probability of 5%, and, (ii) a scenario in which surge line has not ruptured with a probability of 95%. For the branch with no rupture, the simulation proceeds until the R value for the hot leg or the SG tubes reaches 0.518 or the R value for the surge line reaches the second branching point of 0.764. The incremental probability of failure that occurs at the second branching point is conditional on the realization that failure did not occur at the first branching point. Thus, the probability of failure at this point is the increment in cumulative failure probability between the first and second branching points divided by the non-failure probability at the previous branch = (0.25 − 0.05)/0.95. Failure probabilities at subsequent branch points are similarly conditioned. The stopping and branching process continues until all the discrete points on the fragility curve are exhausted for at least one of the failure modes (i.e. surge line, hot leg or SG tube creep rupture).

An advantage of the ADAPT approach is that epistemic and aleatory uncertainties for passive components are treated in a consistent and computationally efficient manner. In the ADAPT approach, epistemic uncertainties are associated with the mathematical models used in the simulation codes. For passive components, aleatory uncertainties are regarded as those that relate to the stochastic nature of the phenomena such as conditions under which rupture and combustion would occur. For example, for given T, σ, m_p, and t_f in Eq. (6), the R value at which creep rupture would occur for the surge line, hot leg or SG tube is regarded as an aleatory uncertainty, but the uncertainty in T, σ, m_p, and t_f are regarded as epistemic uncertainties. The rationale behind this distinction was partly explained at the beginning of Section 3. For this study, only the uncertainty in the R value was considered. From an analysis of the deviation of experimental values around the derived correlation, a standard deviation was calculated. This result is treated as if, had the experimentalist attempted to reproduce failure for piping that was randomly selected from stock that was believed to be essentially identical, the observed failures would have had a distribution of R value at failure with this standard deviation. This viewpoint also provides justification for the classification of the uncertainty in R as an aleatory uncertainty.

Fig. 3 graphically illustrates the approach used for the quantification of the impacts of the epistemic uncertainties on the metric chosen as a system figure of merit (e.g. risk) for the initiating events under consideration. Given the PDFs for the parameters contributing to the epistemic uncertainties (e.g. parameters for heat flux correlations used), parameter values x_n ($n=1, \ldots, N$) and parameter intervals $x_n-\Delta x\le x_n\le x_n+\Delta x$ are selected by the user to

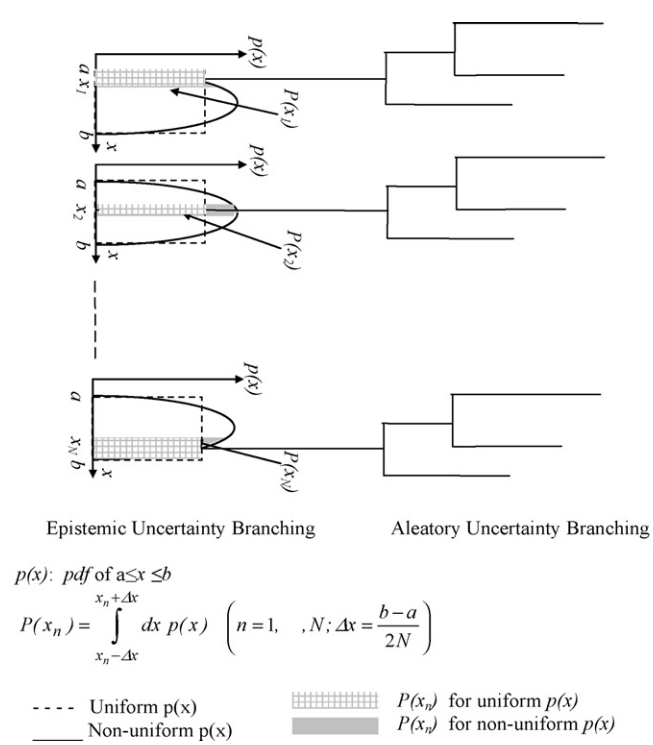

Fig. 3. Quantification of the epistemic uncertainties associated with simulator input parameter $a\le x\le b$ for two example PDFs by fixed points x_n ($n=1, \ldots, N$).

represent different regions in the PDFs. Fig. 3 assumes a single parameter x and uniform size for Δx for clarity of illustration, but the approach can be extended in a straightforward manner to joint PDFs and non-uniform Δx. The integrals of the PDFs over the intervals $x_n - \Delta x \leq x_n \leq x_n + \Delta x$ yield the probabilities associated with x_n. Then ADAPT runs are performed for each x_n ($n = 1, \ldots, N$) with subsequent aleatory branchings for active and passive components as described earlier. The main advantage of this approach (over Monte Carlo and LHS, for example) is that if a sensitivity analysis needs to be performed to quantify the impact of the choice of the PDFs on the system figure of merit, the simulator runs do not have to be repeated. All that needs to be done is to requantify the integrals of the PDFs over the intervals $x_n - \Delta x \leq x_n \leq x_n + \Delta x$. Fig. 3 illustrates the requantification process graphically for a uniform and a non-uniform PDF. ADAPT then can propagate the new epistemic probabilities along the branches in a few seconds, whereas a single simulator run for each APET branch can be order of tens of hours. Note that a new set of simulator runs would be needed for Monte Carlo or LHS approaches every time the PDFs are changed. For a large number of input variables with uncertainty, Taguchi orthogonal arrays (Fowlkes and Creveling, 1995) defined over x_n can be used to identify the most significant parameters and hence to reduce the number of simulator runs without sacrificing coverage (Sharma et al., 2007).

In its current version, ADAPT does not consider similarity between scenarios before-the-fact. For highly non-linear systems for nuclear power plants, this is a difficult problem since a small change in initial conditions and/or model parameters can lead to vastly different consequences. However, standard PRA tools such as SAPHIRE (Smith et al., 2005) can be used to classify the DETs (such as produced by ADAPT), based on several user specified features including consequences, events and initial conditions (Bucci et al., 2006).

3.2. Computational infrastructure

In this subsection we will briefly present the highlights of the computational infrastructure. For detailed description of the computational infrastructure please refer to Rutt et al. (2006) and Catalyurek et al. (2008). The APET generation is managed by a driver 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 probability 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 DET generation schemes, a plant simulator (e.g. MELCOR code) is used to follow the transient along each branch. Branches are pruned based on user-specified criteria to prevent numerical catastrophe. Some significant features of the computational scheme are the following:

1. It is designed for a distributed computing environment. That is, similar to ADSNet (Zhu et al., 2008), ADAPT can execute parts of DET on different computers, such as compute nodes of a cluster, concurrently. Hence the scheduler can track multiple branches simultaneously. This yields a significant reduction in DET generation time, in comparison to executing all branches sequentially on a single computer.
2. The scheduler is modularized so that the branching strategy can be modified, both in terms of the branching rules and execution priority. For example, the CDF of Fig. 2 can be discretized into a larger number of points to check the sensitivity of APET consequences to the number of points chosen to represent the CDF. Similarly, the scheduler can be operated to assign high-execution priority to branches with high probabilities or severe consequences.
3. A distributed database system manages data from the simulation tasks running on different compute nodes, as well as storing the APET structure. Using a distributed database for storing and loading inputs and outputs of the branches avoids creating single point bottlenecks for storing large output files. It both enables to use of aggregate storage capacity of multiple machines, as opposed to using a single machine's storage to store all the results, and enables disk read/write parallelism when a user query intersects multiple branches. For example, if a user query involves a simulator output variable to be plotted for a particular scenario consisting of multiple branches, the distributed database system can read outputs of the branches stored, possibly in multiple machines, and answer the query during the run. APET construction can occur during or after the run.

A schematic overview of the infrastructure is shown in Fig. 4 (Rutt et al., 2006). Following 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 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 meeting branching criteria (e.g. crossing setpoints or reaching the creep rapture parameter values as specified by Fig. 2). The database can consist of minimum cut sets for the Top Events relevant to the branch in the form of binary decision diagrams for fast pre-processing 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 executes 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 execute 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.

ADSNet (Zhu et al., 2008) also uses distributed computing for executing branches of APET concurrently. The major differences are:

1. ADAPT is simulator code-agnostic, it is designed to work with different simulators conforming to a small set of requirements.
2. Parallelization is utmost priority in ADAPT for both improved performance and also to be able to handle large problems. Therefore, both simulation results are stored in a distributed database, and also any required post-processing is also done in parallel by distributed computers.
3. Scheduler is modularized and customizable by user. For example, different execution priorities can be implemented.

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 at the following instances:

1. during execution for task branching and migration, and,

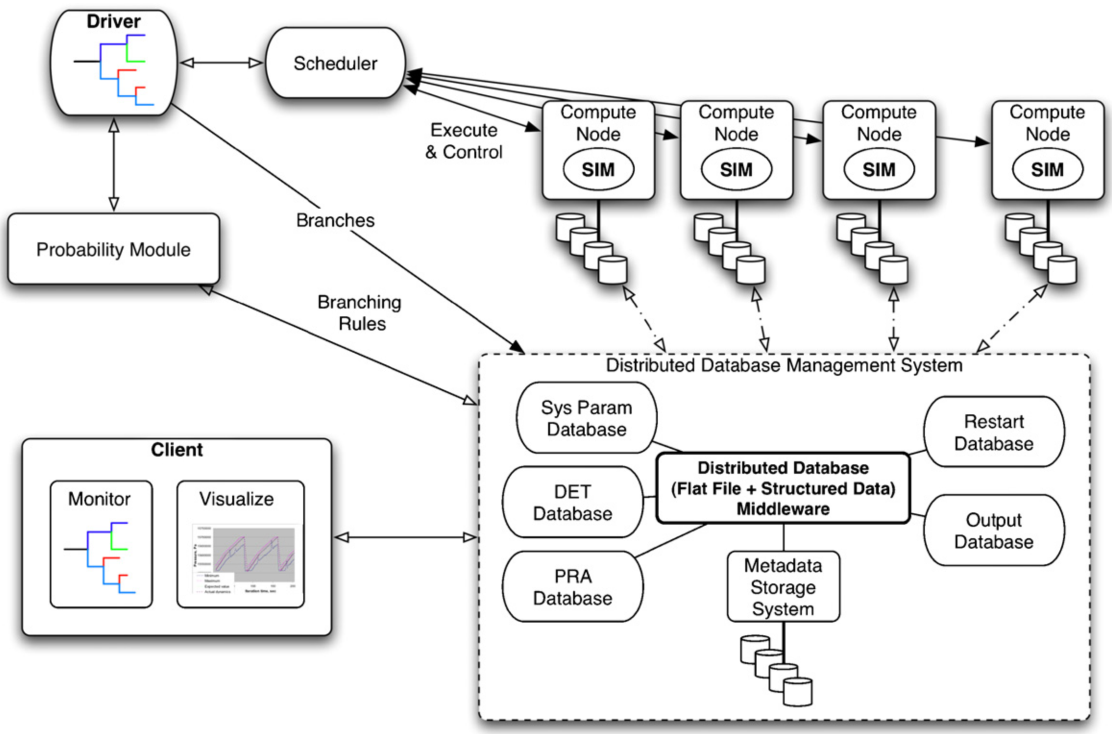

Fig. 4. A schematic overview of computational infrastructure.

2. before and after execution, to load and store its state and results.

The *Driver* communicates with the *Distributed Database Management 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 stores the output files generated by the plant simulator on the *Distributed Database Management System*. The machines on which the distributed databases store the data can be the same as the machines on which the simulation code is executed. Therefore, the *Distributed Database Management System* can take advantage of local disk and avoid expensive network transfers.

4. Implementation and results

The MELCOR code (Summers et al., 1981) was used with ADAPT for dynamic APET generation. Section 4.1 gives a brief overview of MELCOR. Section 4.2 describes the initiating event under consideration and the computational environment. Section 4.3 presents the results.

4.1. The MELCOR code

MELCOR is a fully integrated, relatively fast-running code used to simulate the progression of accidents in light water reactor nuclear power plants. A wide range of accident phenomena can be modeled with MELCOR including thermal–hydraulic response of the reactor coolant system, reactor cavity, containment and confinement buildings; core heat-up, degradation, and relocation; ex-vessel debris behavior; core-concrete attack; hydrogen production, transport, and combustion; fission product release and transport; impact of engineered safety features on thermal–hydraulic and radionuclide behavior. MELCOR has been validated against experimental and plant data (Burns et al., 2005; Birchley, 2004). It uses the "control volume" approach to describe the plant systems. No specific nodalization of a system is forced on the user, which allows a choice of the degree of detail appropriate to the task at hand. Reactor-specific geometry is imposed only in modeling of the reactor core.

A MELCOR calculation is executed in two parts. First, an executable called MELGEN is used to specify, process, and check input data, as well as to generate the initial restart information, written to a restart file. Then, the second executable called MELCOR uses that restart file and specific MELCOR input data (general information including the problem duration, time steps, edit information, etc. written to a separate file called MELCOR Input File) to advance the problem through time.

MELCOR consists of a number of modules called packages. The packages that are of particular interest from the viewpoint of this research work include the Control Functions (CF) package, Flow Paths (FP) package, Burn (BUR) package, and Executive (EXEC) package. The CF package is used by the user to define functions of variables in the MELCOR database. The values of these functions are made available to other packages in MELCOR. ADAPT utilizes the CF package to implement the branching rules for simulations. For example, pressures in appropriate control volumes may be used to control the opening of a valve or initiate the failure of containment, the temperature in a volume may define the enthalpy associated with a mass source/sink, or the particle loading on a filter may modify the flow resistance in the corresponding flow path. The user can also simulate the complicated control logic, involving the values of a number of variables in the system. The FP package, together with Control Volume Hydrodynamics (CVH) package, is used to model thermal-hydraulic behavior of liquid water, water vapor, and gases in MELCOR. The main application of the FP package is to connect the control volumes from the CVH package. The BUR package allows the user to model gas combustion in control volumes. The EXEC package is used to control the overall execution of MEL-

GEN and MELCOR calculations. It coordinates different processing tasks for other MELCOR packages, including file handling, input and output processing, modification of sensitivity coefficients, selection of system time-step, time advancement, and calculation termination.

4.2. Implementation

A PWR with a large dry containment was used as a reference system, with station blackout as the initiating event compounded by assumed failure of the AFWS and a stuck open safety relief valve on the secondary side of the plant. In this case, it is not possible to maintain an adequate water level in the steam generators to remove fission product decay heat from the system.

The base MELCOR input deck for this scenario was provided by Sandia National Laboratories. Fig. 5 shows the configuration control volumes used (nodalization). The plant has four RCS loops each with a U-tube SG and a reactor coolant pump. The four RCS loops are represented by two loops, a single loop with the pressurizer (nodes 500–590) and a triple loop containing the remaining three loops (nodes 600–690). The nodalization is identical for the single and triple loops, except for the presence of the pressurizer in the single loop. The hot leg is divided into two directions, each with two nodes (500–503 for single loop, and 600–603 for triple loop), to account for the steam counter flow from the steam generators during certain accident scenarios.

The primary side of steam generator has a finer nodalization scheme (nodes 514–518 and 614–618). The SG outlet plenum is represented with a single node (nodes 585 and 685). Finally, the cold leg has four nodes: two before (nodes 520–521 and 620–621) and two after the reactor coolant pump (nodes 522–523 and 622–623).

There is a pressurizer on one of the loops connected to the hot leg. The overall pressurizer volume including the surge line is divided into seven nodes: six in the pressurizer itself (402–407) and one for the surge line (490). The pressurizer relief tank is represented by a single node (450). No control volumes are allocated to pressurizer pressure operated relief valves (PORVs) and safety relief valves (SRVs). Instead, fluid removal from the pressurizer through these valves is simulated using flow paths.

The core nodalization is represented as a 5-ring, 12-level model with three core control volumes per thermal–hydraulic level and 10 heated levels (not shown in Fig. 5). The rest of the reactor vessel area is represented by four nodes: downcomer (node 310), lower plenum (node 320), core bypass (not shown in Fig. 5), and upper head (node 399).

The results of a recent study (Eide et al., 2005) performed by the U.S. Nuclear Regulatory Commission (NRC), provided data for the non-recovery probability for offsite electric power as a function of time (Table 2). The containment failure fragility curve was based on the least robust of the large, dry containment designs (Pilch et al., 1996). Thus, the resulting analyses are not truly representative of any existing nuclear power plant.

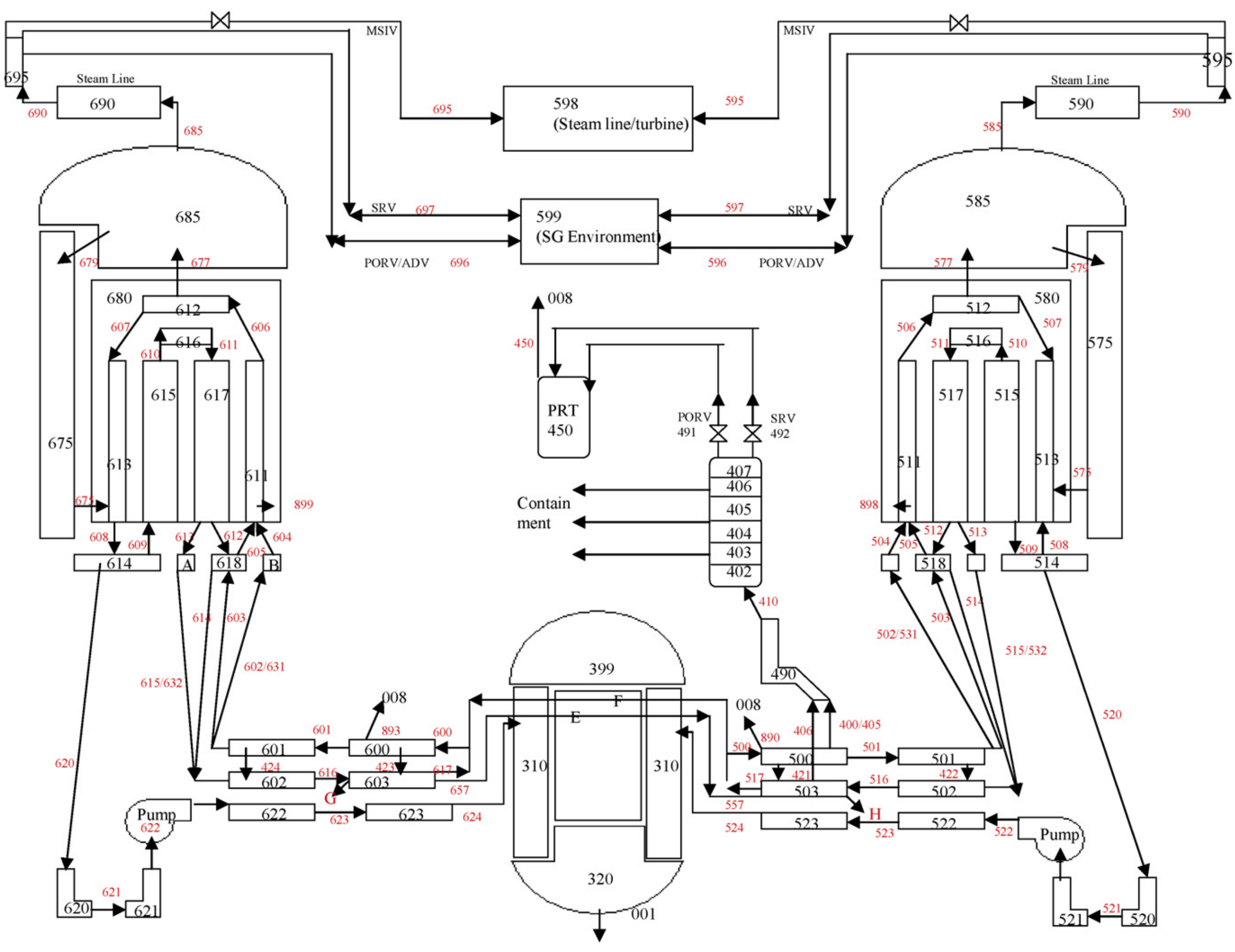

Fig. 5. Reference power plant nodalization (Vierow et al., 2003).

Table 2
Non-recovery probability for off-site power versus duration (Eide et al., 2005).

Duration (h)	Non-recovery probability
1	0.53
2	0.32
4	0.16
6	0.0096
8	0.0067
10	0.0051
12	0.0040
14	0.0033
16	0.0029
18	0.0025

The coupled ADAPT/MELCOR package was run under Linux platform on a cluster of 32 processors. The starting time for the application was chosen 200 s before the initiating event to allow the system parameters to be stabilized. The end time of the accident simulation was chosen 120,000 s (about 33 h) into the accident by which time containment failure will have occurred if power has not been restored. The branch probability cutoff value was set to 1 to allow running all possible accident scenarios. The approximate running time to the overall completion was about 15 days and included the simulation of about 600 scenarios. It should be noted that not all branches were able to run to that specified end time due to MELCOR abnormal terminations. As a result, it was necessary for the analyst to examine each of the terminated cases to determine how the failure to complete the run would affect the resulting containment failure probability. Although this was feasible in the simple case analyzed, in general, the analyst would have to determine the reason for abnormal termination, correct it, and rerun the failed cases. Thus, robustness of the transient analysis code is essential to the successful application of an automated technique like ADAPT. In cooperation with a large, international group of MELCOR users, continued upgrading of the MELCOR code is decreasing the number of abnormal terminations experienced.

4.3. Results

The results of this study illustrate a number of advantages of the DET approach in comparison to the standard use of static event trees. The critical time periods and modes of containment failure in this example are: containment bypass resulting from steam generator tube rupture following core uncovery, containment failure from hydrogen deflagration following substantial core degradation, containment failure by hydrogen deflagration late in the accident following power recovery, and long-term containment failure by over pressurization.

The separation of PRA into system related events in Level 1 and accident progression related events in Level 2 does not work effectively for systems that are affected by accident events or for the recovery of failed systems. When dealing with the feedback of Level 2 events on Level 1 events, or with the recovery of failed systems, the analyst must consider multiple combinations of events, which can only be addressed approximately. Typically, the uncertainty bounds are so great that an approximate treatment can be tolerated without impacting the validity of conclusions. On the other hand, the DET analysis can address these issues in a consistent manner and avoid the need to assess the acceptability of an approximate treatment. For example, the transition from Level 1 to Level 2 PRA can be performed in a seamless fashion with core damage states determined in Level 1 providing initial conditions for Level 2. At this point it should be mentioned that while any DET tool has this capability inherently, ADAPT is the only one to the authors' knowledge that is specifically designed to handle phenomena encountered in Level 2 PRA (e.g. rupture, hydrogen burn), as well as active system failures mostly encountered in Level 1 PRA. Again, ADAPT is also the only DET tool to the authors' knowledge that performs uncertainty quantification simultaneously with the PRA, rather than an additional task following the PRA.

In this case study, power recovery is modeled explicitly. Early power recovery (prior to 3 h and 20 min after the transient) would arrest the accident prior to the initiation of core damage. The non-recovery probability at this point is 0.20. Thus, 80% of the scenarios that are identified in Level I analysis as station blackout scenarios potentially leading to core damage are shown in the ADAPT/MELCOR analysis to be terminated before core damage would occur Because of inadequate heat removal, the PRVs on the reactor coolant system would cycle open and closed to limit the pressure. Failure of a PRV to close is also modeled explicitly in the case study. Such a failure would result in the loss of water inventory from the reactor coolant system and decrease the time available for power recovery to prevent core damage from 3 h and 20 min to 2 h and 30 min. Power recovery later in the accident will result in deinerting of the containment atmosphere and the potential for a hydrogen deflagration. If power recovery occurs by approximately 10 h, late overpressure failure of the containment can be precluded.

The focus of this case study is on the competition between failure of the reactor coolant system by surge line failure, hot leg failure, or steam generator tube rupture. The offsite consequences of steam generator tube rupture are particularly severe because it leads to a release of radionuclides to the environment that bypasses the containment building. If either surge line failure or hot leg failure precedes steam generator tube rupture, tube rupture would be precluded. A deterministic analysis of this event leads to the conclusion that surge line failure precedes steam generator tube failure. However, the probabilistic analysis using ADAPT indicates a probability of steam generator tube rupture and containment bypass of 0.3%. Although a stuck open relief valve decreases the time available for power recovery before fuel damage, it has the benefit of reducing the potential for steam generator tube rupture. It should be indicated at this point that the authors make no claim that the 0.3% containment bypass is an accurate characterization of a real plant. It is merely a realistic demonstration of a process.

The other failure mode with potentially high consequences occurs at the time of failure of the reactor coolant system by creep rupture of the surge line or hot leg. At this point there is both a rapid depressurization of the reactor coolant system into the containment and also the potential for a hydrogen deflagration event potentially threatening the integrity of the containment. In the Three Mile Island Unit 2 (TMI-2) accident, a hydrogen deflagration event occurred at a similar stage of the accident following an opening of the PORV. In that accident, the hydrogen deflagration event resulted in an increase in containment pressure of approximately 2 bar. Prior to the deflagration event, the containment pressure had been approximately atmospheric. In the case study, depressurization of the reactor coolant system without hydrogen combustion results in an increase in containment pressure from the pre-existing value of 2–3.5 bar due to the release of gases (water vapor and hydrogen) to the containment atmosphere. In the NUREG-1150 approach to event tree quantification, pressure increments of the gases in the containment were added at different stages of the accident to assess containment failure probability. If hydrogen combustion were to add two bars to the atmospheric pressure, as in TMI-2 accident, the resulting pressure of 5.5 bar would result in an approximate probability of containment failure of 5%. However, the more mechanistic calculation performed by MELCOR only indicates a peak pressure of 4 bar, which results in a negligible probability of this mode of early containment failure.

Table 3
Containment failure modes and probabilities.

Containment failure mode	Timing (s)	Probability (%)	
		Overall	Conditioned on core damage
Steam generator tube rupture–containment bypass	15,000–16,200	0.3	1.5
Hydrogen combustion at vessel failure	16,330–16,450	Negligible	Negligible
Hydrogen combustion at late power recovery	16,500–35,000	Negligible	Negligible
Late overpressure failure	>35,000	2.8	13.9

If power is not recovered, long-term containment failure is assessed to occur with 100% probability. The transfer of heat through the containment wall is insufficient to prevent the continued increase of pressure. In the time period of 35,000 s (9.7 h) to 70,000 s (19.4 h), as containment pressure increases, the fragility curve of the probability of containment failure as a function of pressure indicates an increase in failure probability from 0% to 75%. Over this time period, the power non-recovery probability decreases from approximately 5% to 2.5%. The combination of power recovery versus time and containment failure probability versus pressure results in a late containment failure probability of 2.8%. Table 3 summarizes containment failure mode probabilities obtained in the case study analyses. As indicated earlier, these results are not representative of a specific nuclear power plant. For the analysis performed, there is a significant truncation error associated with the choice of only five points to represent the fragility curves. An even greater source of uncertainty is associated with the judgment involved in the development of a fragility curve. Thus, in the process of developing fragility curves it is incumbent on the analyst to make a judgment as to the associated uncertainty in order to be able to characterize the uncertainties in the branching probabilities.

There is also a significant advantage in using parallel processing for DET generation in terms of the computational effort spent on the same scope of analysis. Fig. 6 illustrates the time savings on a log scale when using multiple processors for dynamic analysis with parallel distributed computing versus a single processor. The comparison is performed between two cases with, respectively, 1 processor and 32 processors, for the total of 310 completed jobs (event tree branches). As shown, there is an order of magnitude of wall-clock time difference between the two cases (1 week versus 2.5 months). While the advantages of parallel computing are generic to DET not specific to ADAPT, ADAPT is the only tool available that the authors are aware of that uses this advantage in APET generation and analysis.

In the NUREG-1150 process for quantifying event trees, a large number of sensitivity studies are performed as input to a subjective judgment of event tree branching probabilities. There are no clear guidelines available for the number or scope of those sensitivity studies. In the dynamic event tree approach, judgment is also required but the manner in which uncertainties are being addressed is transparent. In the sensitivity analysis approach, each scenario is typically a complete analysis. In the multiple parallel processor approach used in the ADAPT analysis, the early stages of the analysis are not repeated, which results in significant time saving. There is also much less effort involved in problem setup.

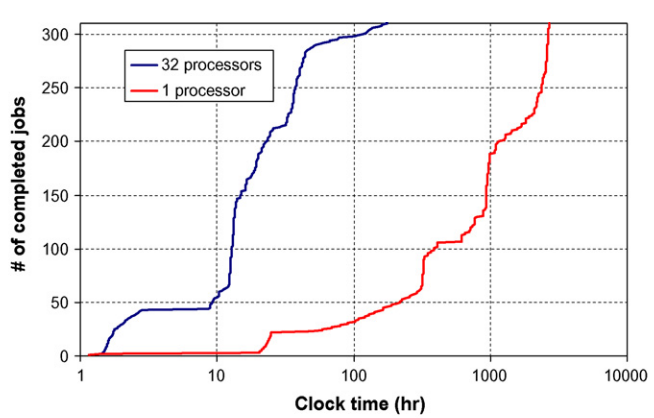

Fig. 6. Comparison of computational effort between regular and parallel processing.

Although the ADAPT type of analysis appears to be complex and difficult to verity, it is not difficult to assess the reasonableness of results at the end. The uncertainty distributions used in the analysis are explicitly developed at the outset of the analysis and are available for critique by the reviewer. Because mean values propagate through the analysis, it is possible to subdivide the process into major events, combine results to obtain mean probabilities for each event, combine event probabilities, to obtain scenario probabilities and combine these with scenario consequences. The mean risks obtained in this manner can then be compared with the mean values of the risk obtained from the study.

5. Conclusion

This paper describes the ADAPT methodology with implementation to severe accident phenomenological uncertainty treatment. As examples of probabilistic modeling of severe accident phenomena in Level 2 PRA, ADAPT was applied to quantify the likelihood of creep rupture of pressurizer surge line, hot leg, and SG tubes in a PWR with a large dry containment using MELCOR. A station blackout initiating event with a failure of the AFWS was considered as a test case.

The results of the study indicate that the approach presented in this paper can significantly reduce the manual and computational effort in Level 2 PRA analysis. ADAPT does not require any human intervention throughout the analysis. By implementing the model mechanistically, it also eliminates the potential of introducing errors while making changes in the input decks manually for running new accident scenarios. From the phenomenological viewpoint, it can also treat the epistemic and aleatory uncertainties associated with complex physical phenomena taking place during severe accident progression. Because the DET approach allows the order of events to vary, many potential accident scenarios that are ignored or treated in an approximate aggregate manner in current conventional PRA Level 2 analyses (see hydrogen combustion example in Section 1) are accounted for in a phenomenologically consistent manner in the proposed methodology, resulting in the consideration of a much wider variety of accident scenarios. The ADAPT methodology can also be potentially used for Level 1 PRA, as well as Level 2 analysis, of future plants with passively safe accident mitigation features.

Acknowledgements

The research presented in this paper was partially supported by a contract from the Sandia National Laboratories (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 there of, 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

Acosta, C., Siu, N., 1993. Dynamic event trees in accident sequence analysis: application to steam generator tube rupture. Reliability Engineering and System Safety 41, 135–154.

Birchley, J., 2004. Assessment of the MELCOR code against Phebus experiment FPT-1 performed in the frame of ISP-46. In: Proceedings of the International Conference on Nuclear Engineering (ICONE-12), vol. 3, pp. 551–560.

Bucci, P., Mangan, L.A., Kirschenbaum, J., Mandelli, D., Aldemir, T., Arndt, S.A., 2006. Incorporation of Markov reliability models for digital instrumentation and control systems into existing PRAs. In: Proceedings of NPIC&HMIT 2006. American Nuclear Society, La Grange, Illinois.

Burns, C.J., Liao, Y., Vierow, K., 2005. MELCOR code assessment by simulation of TMI-2 phases 1 and 2. In: Proceedings of the 11th International Topical Meeting on Nuclear Reactor Thermo-hydraulics (NURETH-11), October 2–6, Avignon, France.

Cacciabue, P.C., et al., 1986. Dynamic logical analytical methodology versus fault tree: the case of AFWS of a nuclear power plant. Nuclear Technology 74, 195–208.

Catalyurek, U., Rutt, B. et al., 2008. Development of a Code-Agnostic Computational Infrastructure for the Dynamic Generation of Accident Progression Even Trees. Technical Report OSUBMI_TR_2008_n03. The Ohio State University, Department of Biomedical Informatics.

Cojazzi, G., Sardella, R., Trombetti, T., Vestrucci, P., 1994. Assessing DYLAM methodology in the frame of Monte Carlo simulation. In: Proc. Probabilistic Safety Assessment and Management. International Association for Probabilistic Safety and Management, pp. 011/13–18.

Cojazzi, G., et al., 1996. The DYLAM approach to the dynamic reliability analysis of systems. Reliability Engineering & System Safety 52, 279–296.

Eide, S.A., et al., 2005. Reevaluation of the Station Blackout Risks at Nuclear Power Plants, Analysis of Loss of Offsite Power Events: 1986–2004. NUREG/CR-6890, vol. 1, U.S. Nuclear Regulatory Commission.

Fowlkes, W.Y., Creveling, C.M., 1995. Engineering Methods for Robust Product Design: Using Taguchi Methods in Technology and Product Development (Engineering Process Improvement Series). Prentice Hall.

Hofer, E., Kloos, M., et al., 2004. Dynamic Event Trees for Probabilistic Safety Analysis. GRS, Garsching, Germany.

Hsueh, K.-S., Mosleh, A., 1996. 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.

Majumdar, S., 1999. Prediction of Structural Integrity of Steam Generator Tubes Under Severe Accident Conditions. Nuclear Engineering and Design, pp. 31–55.

Marchand, S., et al., 1998. DDET and Monte Carlo Simulation to Solve Some Dynamic Reliability Problems, PSAM 4. Springer Verlag, New York, pp. 2055–2060.

Munoz, R., et al., 1999. A Second Generation Scheduler for Dynamic Event Trees, Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications. Madrid, Spain, pp. 1358–1367.

Munoz, R., Minguez, E., Melendez, E., Izquierdo, J.M., Sanchez-Perea, M., 1999b. DENDROS: a second generation scheduler for dynamic event trees, mathematics and computation, reactor physics and environmental analysis in nuclear applications. In: Conference Proceedings, Madrid, Spain.

Pilch, M.M., et al., 1996. Resolution of the Direct Containment Heating Issue for All Westinghouse Plants with Large Dry Containments or Subatmospheric Containments, NUREG/CR-6338, SAND95-2381. Nuclear Regulatory Commission.

Rutt, B., Catalyurek, U., et al., 2006. Distributed dynamic event tree generation for reliability and risk assessment. In: CLADE 2006 Workshop, Paris, France.

Sancaktar, S., Sharp, D.R., 1985. Living PRA concept for plant risk, reliability, and availability tracking. In: Proceedings of International Conference on Nuclear Power Plant Aging, Availability Factor and Reliability Analysis, San Diego, CA.

Sharma, L., Aldemir, T., Parker, R., 2007. Uncertainty analysis using Taguchi methods with virtual experiments. Transactions of the American Nuclear Society 95, 427–429.

Smidts, C., Swaminathan, S., 1996. Improvements to Discrete Dynamic Methodologies, PSA-96. American Nuclear Society, pp. 159–166.

Smith, C.L., Knudsen, J.M., Calley, Beck, S., Kvarfordt, K., Wood, S.T., 2005. SAPHIRE Basics: An Introduction to Probability Risk Assessment Via the Systems Analysis Program for Hands-on Integrated Reliability Evaluations (SAPHIRE) Software. Idaho National Laboratory, Idaho Falls, ID.

Summers, R.M., Cole, R.K., et al., 1981. MELCOR 1.8.0: A Computer Code for Nuclear Reactor Severe Accident Source Term and Risk Assessment Analyses.

US Nuclear Regulatory Commission (USNRC), 1975. Reactor Safety Study—An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, WASH-1400 (NUREG-75/014).

US Nuclear Regulatory Commission (USNRC), 1990. Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants, NUREG-1150.

Vesely, W.E., 1991. Incorporating aging effects into probabilistic risk analysis using a Taylor expansion approach. Reliability Engineering and System Safety 32 (3), 315–337.

Vierow et al., "Task Order No. 12, MELCOR Assessment against SCDAP/RELAP5", Annual Report, January 1, 2002 through December 31, 2002, prepared for the U.S. Nuclear Regulatory Commission, Jan. 2003.

Zhu, D., Chang, Y.H., Mosleh, A., 2008. The Use of Distributed Computing for Dynamic PRA: The ADS Approach. In: Proceedings of PSAM 9. CD-ROM Version, IAPSAM.