Incorporation of Markov Reliability Models for Digital Instrumentation and Control Systems into Existing PRAs

P. Bucci[1], L. A. Mangan[2], J. Kirschenbaum[1], D. Mandelli[2], T. Aldemir[2], S. A. Arndt[3]

[1]*The Ohio State University, Dept. of Computer Science & Engineering, 395 Dreese Labs, 2015 Neil Ave., Columbus, OH, 43210*
[2]*The Ohio State University, Nuclear Engineering Program, 650 Ackerman Road, Columbus, OH 43202*
[3]*U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, D.C. 20555*

Abstract – *Markov models have the ability to capture the statistical dependence between failure events that can arise in the presence of complex dynamic interactions between components of digital instrumentation and control systems. One obstacle to the use of such models in an existing probabilistic risk assessment (PRA) is that most of the currently available PRA software is based on the static event-tree/fault-tree methodology which often cannot represent such interactions. We present an approach to the integration of Markov reliability models into existing PRAs by describing the Markov model of a digital steam generator feedwater level control system, how dynamic event trees (DETs) can be generated from the model, and how the DETs can be incorporated into an existing PRA with the SAPHIRE software.*

I. INTRODUCTION

The direct interaction (through hardware/software/ firmware) and/or indirect interaction (through the controlled/monitored process) between components of digital instrumentation and control (I&C) systems may lead to statistical dependence between failure events [1]. Such a statistical dependence may be represented by Markov models [1]. However, for current and near term applications, one requirement for a methodology for digital I&C system reliability model construction is that it should be possible to incorporate the resulting model into an existing probabilistic risk assessment (PRA) for the overall plant that the digital I&C system is part of. This requirement often necessitates the Markov model to be converted to a form that is compatible with the available PRA software, such as SAPHIRE [2], most of which use the event-tree/fault-tree (ET/FT) approach. Such a conversion procedure has been reported earlier [3]. Using the steam generator feedwater level control system of a pressurized water reactor (PWR) as an example digital I&C system and the SAPHIRE model of a NUREG-1150 [4] (Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants) plant as an example PRA model, this paper illustrates how dynamic event trees (DETs) can be generated from the Markov model and integrated into an existing PRA.

The paper is organized as follows. In the next section we present a digital steam generator feedwater level control system (DFWCS) used throughout the paper. In the following sections we describe a Markov model of the DFWCS and how to generate dynamic event trees from the model. We then discuss how the generated event trees can be incorporated into an existing PRA. Finally we conclude the paper with a brief discussion of outstanding issues and areas where further work is required.

II. A DIGITAL STEAM GENERATOR FEEDWATER LEVEL CONTROL SYSTEM

A detailed description of a benchmark system and the steam generator feedwater level control system can be found in [5]. For the purposes of this paper, we summarize here the relevant characteristics of the system.

The DFWCS is intended to keep the water level inside the PWR's steam generator (SG) within a given range around the setpoint level by controlling a feedwater pump (FP), a main feedwater regulating valve (MFV) and a bypass feedwater regulating valve (BFV). The system includes two computers—a main computer (MC) and a backup computer (BC)—that execute the same control algorithms. The computers receive signals from sensors measuring feedwater level, neutron flux (power), feedwater flow, steam flow, and feedwater temperature, and send the computed output signals to MVF, FP, and BFV. Each of these physical devices receives the appropriate input (from MC or BC) as determined by a decision controller.

The feedwater control system can operate in several different modes depending on the power generated by the primary system. However, to illustrate how the reliability model constructed with the Markov methodology can be incorporated into an existing PRA, we will only consider the behavior of the controller as the result of one example initiating event. We assume the following:

1. Turbine trips
2. Reactor is shutdown
3. Power P is generated from the decay heat
4. Reactor power and steam flow rate reduce to 6.6% of 1500 MW_{th} 10 seconds after reactor shutdown
5. Feedwater flow is at nominal level
6. Off-site power is available

7. Main computer is failed

Following the plant trip, the feedwater control system is operating in low power mode, and that implies that the MFV is closed and the BFV is used exclusively to control the feedwater flow [5].

In the next section we show how the system is modeled using the Markov methodology and how to generate dynamic event trees for the chosen example initiating event.

III. MARKOV MODEL OF DFWCS

To construct a Markov model of the system under consideration, we employ the cell-to-cell mapping technique (CCMT) [6]. The CCMT is a systematic procedure to describe the dynamics of both linear and non-linear systems in discrete time and discretized system state space (or the subspace of the controlled variables only). The CCMT first requires a knowledge of the Top Events for the partitioning of the state space into V_j (j=1,…,J) cells. The evolution of the system in discrete time is modeled and described through the probability $p_{n,j}(k)$ that the controlled variables are in a predefined region or cell V_j in the state space at time $t=k\Delta t$ (k=0, 1,…) with the system components (such as pumps, valves, or controllers) having a components states combination n=1,…,N. The state combination represents the system configuration at a given time and contains information regarding the operational (or the failure) status of each component. Transitions between cells depend on:

- the dynamic behavior of the system
- control logic of the control system
- hardware/firmware/software states.

The dynamic behavior of the system is usually described by a set of differential or algebraic equations as well as the set of control laws. The operating/failure states of each component are specified by the user.

III.A. Top Events

The purpose of the feedwater controller is to maintain the water level x inside the SG within ±2 inches of the setpoint level (defined at 0 inches). The controller is regarded as failed if the water level in SG rises above +2.5 feet or falls below –2 feet. So we can define two Top Events:

1. x < –2 feet (Low Level)
2. x > +2.5 feet (High Level).

III.B. Control Laws

Here are the equations describing the dynamic behavior of the system and the control laws as they are applied to the example initiating event. f_{wn} is the feedwater flow rate, f_{sn} is the steam flow rate, h_{wn} is the feedwater temperature; r_n is the level setpoint for the SG, p is the power level of the SG, μ_{Bn}, α_{Bn}, β_{Bn}, A, and τ_1-τ_5 are user-specified constants. η_{Bn} is a history-dependent value, i.e., the BFV position value determined at the previous time step.

Water Level (x_n):

$$\frac{dx_n}{dt} = A(f_{wn} - f_{sn}) \tag{1}$$

Water Level Error (E_{Ln}):

$$\tau_5 \frac{dE_{Ln}}{dt} = r_n - C_{Ln}(t) \tag{2}$$

Compensated Water Level (C_{Ln}):

$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln}(t) + x_n(t) + \tau_1 A(f_{wn} - f_{sn}) \tag{3}$$

Compensated Power (C_{pn}):

$$C_{pn}(t) = p(0)e^{-t/\tau_4} + \frac{(1+\tau_3)}{\tau_4}\int_0^t du\ p(t-u)e^{-u/\tau_4} \tag{4}$$

BFV Demand (σ_{Bn}):

$$\sigma_{Bn}(t) = \mu_{Bn}\alpha_{Bn} + \mu_{Bn}C_{pn}(t) + \beta_{Bn}(h_{wn})E_{Ln}(t) \tag{5}$$

BFV Position % (S_{Bn}):

$$S_{Bn} = \begin{cases} \sigma_{Bn} & \text{main or backup CPU up} \\ \eta_{Bn} & \text{both main and backup CPU down} \end{cases} \tag{6}$$

Power (p):

$$p(t) = p(0) \times \left(\frac{1}{(10+t)^{0.2}} - \frac{1}{(3.15\times 10^7 + t)^{0.2}} \right) \tag{7}$$

In Eq.(1), the water flow rate f_{wn} is 0 if BFV is closed. Otherwise, f_{wn} is obtained from the solution of

$$\frac{4.73\mathrm{L}(100/S_{Bi})^2 f_{wn}^{1.852}}{\mathrm{C}^{1.852}\mathrm{D}^{4.87}} = 136 + 6.3\times 10^{-6} f_{wn} - 4.6\times 10^{-11} f_{wn}^2 \tag{8}$$

where D is the diameter of the inlet pipe to the BFV (in feet) and f_{wn} is in ft^3/s. L is a fitting parameter, and C is a constant. Eq.(8) uses the pump and valve models given in NUREG/CR-6465 [7] and assumes that pump head is equal to the head loss in the valve.

The steam flow rate (f_{sn}) follows the primary system decay heat generation rate, i.e.,

$$f_{sn}(t) = f_{sn}(0) \times \left(\frac{1}{(10+t)^{0.2}} - \frac{1}{(3.15\times10^7 + t)^{0.2}} \right) \quad (9)$$

Eq.(9) assumes the reactor has operated for 1 year and the starting point of the analysis is 10 seconds after the turbine trip.

III.C. Partitioning of the State Space

The dynamics of the system is modeled as transitions between cells V_j (j=1,…,J) that partition the state space. For the example initiating event, Eqs.(1)–(4) show that the state space is 4-dimensional and is comprised of

- water level x_n
- water level error E_{Ln}
- compensated water level C_{Ln}
- BFV position S_{Bn}.

The partitioning needs to be performed in such a way that, other than V_j being disjoint and covering the whole space (definition of partitioning), values of the controlled variables defining the Top Events (in our case x_n) and the setpoints must fall on the boundary of V_j and not within V_j. If this requirement is not satisfied for some $V_{j'}$, then the system state becomes ambiguous when the state variables are within $V_{j'}$ since the methodology assumes that $p_{n,j}(k)$ is uniformly distributed over $V_{j'}$.

Tables I.–IV below show the actual partitioning scheme we used for each process variable.

TABLE I. Partitioning for water level

Interval for x_n	Range
–2	$x_n < -2.0$
–1	$-2.0 \le x_n < -0.17$
0	$-0.17 \le x_n < 0.17$
+1	$0.17 \le x_n \le 2.5$
+2	$x_n > 2.5$

TABLE II. Partitioning for water level error

Interval for E_{Ln}	Range
–1	$-1000 \le E_{Ln} < -1.587$
0	$-1.587 \le E_{Ln} < 4.203$
+1	$4.203 \le E_{Ln} \le 1000$

TABLE III. Partitioning for compensated water level

Interval for C_{Ln}	Range
–1	$-500 \le C_{Ln} < -100$
0	$-100 \le C_{Ln} < 100$
+1	$100 \le C_{Ln} \le 500$

TABLE IV. Partitioning for BFV position

Interval for S_{Bn}	Range
0	$0 \le S_{Bn} < 30$
+1	$30 \le S_{Bn} < 70$
+2	$70 \le S_{Bn} \le 100$

The number and size of the intervals to partition each process variable and the choice of the time increment Δt are dependent on each other. Essentially, a finer partition (with a larger number of smaller intervals) can yield a better approximation of the system at a cost of extra computational resources. Furthermore, the time increment is dependent on the size of the cells: too small a time increment may result in the CCMT not producing useful results if most of the sample points and trajectories fail to leave the starting cell; too large an increment may cause some CCMT trajectories to cross multiple setpoint boundaries. Therefore, it is necessary to determine the partitioning scheme and the time interval by analyzing the actual system.

The partitioning chosen for the level variable is based on the following observations:

- the Low Level and High Level points have been identified in Section III.A;
- Section III.A also points out that it is desirable to keep the level between ± 2 inches of the setpoint, i.e., ± 0.17 feet;
- the other intervals for the level variable were added to provide a finer description of the behavior of the variable of primary interest.

The partitioning chosen for the BFV position is based on NUREG/CR-6465 [7]. The range of this variable is naturally 0%-100%. The range for level error and compensated level were determined experimentally through simulation of the system. The middle interval of the level error captures the range of values which correspond to the entire range of values of the BFV position variable (which is computed as a function of level error). Finally the partitioning for the compensated level was chosen to minimize the number of intervals while still modeling nominal, low, and high levels for this variable.

Given the partitioning of the process variables, Δt = 1 second was chosen experimentally as a reasonable time increment relative to the size of the process variables intervals

III.D. Modeling the System Components

Under the assumptions made in Section II about the example initiating event, we have to consider only three system components: the backup computer (BC), the bypass feedwater valve (BFV), and its controller.

Through a failure modes and effects analysis (FMEA) of the DFWCS, we have determined the following significant states of for the components of interest.

The BC can be in one of 4 states:

1. Operating
2. Loss of input signals
3. Loss of output signals
4. Down (failed)

The following diagram shows the states for BC and the possible transitions (we assume no recovery from a failed state is possible).

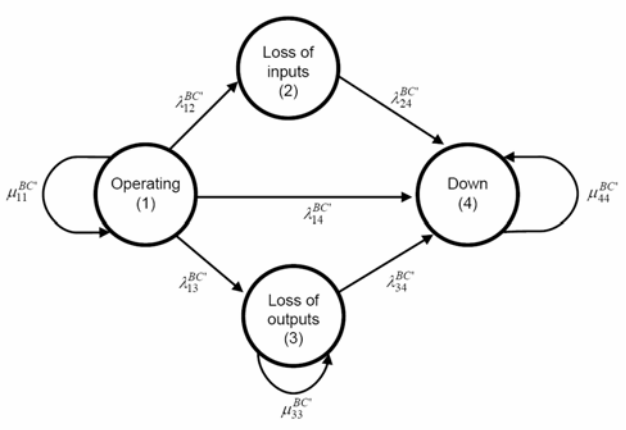


Fig. 1. States of the backup computer

In modeling the BFV and its controller, we have chosen to combine them into a single macro-component because of their tight coupling. For the remainder of the paper, we will refer to the combination BFV and BFV controller as BFV for simplicity. The BFV can be in one of 3 states:

1. Operating and able to detect failures from the computers
2. Operating but not able to detect failures from the computers
3. Stuck in the position of the previous time step

The following diagram shows the states for the BFV and the possible transitions (no recovery from a failed state is possible).

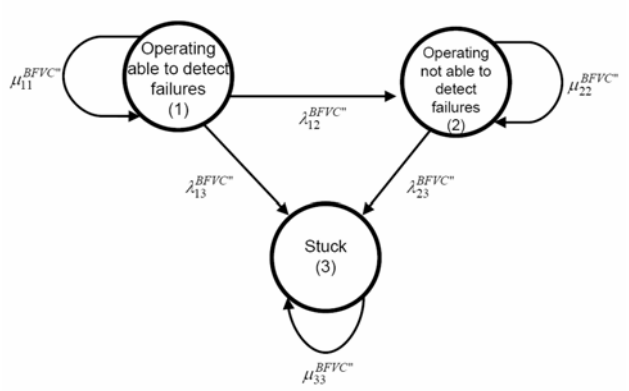


Fig. 2. States of the bypass valve and its controller

When the two components (BC and BFV) are considered together, it turns out that there are dependencies between them. Not all transition implied by the diagrams in Figs. 1 and 2 can actually occur. For instance, if the BC is in the Loss-of-outputs state and the BFV is in the Operating-and-able state, the BFV will detect the problem with the BC and will make a transition into the Stuck state. Table V below shows the possible transitions. A 'Y' indicates that the transition from the row state to the column state can occur. An 'N' or an 'X' means that the transition cannot occur. The 'X's show those transition that are ruled out by the coupling of the two units (BC and BFV).

TABLE V. Possible transitions of system components

	BFV 1 BC 1	BFV 1 BC 2	BVF 1 BC 3	BFV 1 BC 4	BFV 2 BC 1	BFV 2 BC 2	BFV 2 BC 3	BFV 2 BC 4	BFV 3 BC 1	BFV 3 BC 2	BFV 3 BC 3	BFV 3 BC 4
BFV 1 BC 1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BFV 1 BC 2	N	N	N	Y	N	N	N	Y	N	N	N	Y
BFV 1 BC 3	N	N	**X**	**X**	N	N	**X**	**X**	N	N	Y	Y
BFV 1 BC 4	N	N	N	**X**	N	N	N	**X**	N	N	N	Y
BFV 2 BC 1	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
BFV 2 BC 2	N	N	N	N	N	N	N	Y	N	N	N	Y
BFV 2 BC 3	N	N	N	N	N	N	Y	Y	N	N	Y	Y
BFV 2 BC 4	N	N	N	N	N	N	N	Y	N	N	N	Y
BFV 3 BC 1	N	N	N	N	N	N	N	N	Y	Y	Y	Y
BFV 3 BC 2	N	N	N	N	N	N	N	N	N	N	N	Y
BFV 3 BC 3	N	N	N	N	N	N	N	N	N	N	Y	Y
BFV 3 BC 4	N	N	N	N	N	N	N	N	N	N	N	Y

The BFV position is a function also of the state of the two system components as reflected by Eq.(6). This reflects the history dependence intrinsic to the system. In this respect, the BFV position as a function of all the possible combinations of component states is presented in Table VI.

TABLE VI. BFV position as function of system components

n	BFV	BC	BFV Position
1	OK able	OK	CV
2	OK able	Input loss	OV
3	OK able	Output loss	OV
4	OK able	Down	OV
5	OK not able	OK	OV
6	OK not able	Input loss	OV
7	OK not able	Output loss	Closed
8	OK not able	Down	Any
9	Stuck	OK	OV
10	Stuck	Input loss	OV
11	Stuck	Output loss	OV
12	Stuck	Down	OV

In Table VI., "CV" stands for current value, i.e., the valve position is determined from the equations; "OV" stands for old value, i.e., the valve position determined at the previous time step; "Closed" means that the valve is completely closed; and "Any" means that the valve position can be any value over the 0…100 range.

IV. GENERATION OF DYNAMIC EVENT TREES FROM MARKOV MODEL

The generation of dynamic event trees from the Markov model uses the Markov model transition probability matrix as the description of a finite-state machine representing a discrete process model of the stochastic dynamic behavior of the system [3]. Starting from a set of initial states of interest, the algorithm searches the state space for all possible paths to failure and generates the corresponding dynamic event tree. The generated tree maintains not only the order of occurrence in time of the events, but also detailed timing information about when each event has occurred. Dynamic event trees can be generated to a predefined depth to explore the behavior of the system for a specified interval of time. They can also be pruned by ignoring branches for which the probability of occurrence is below a chosen threshold or by only considering specific events of interest (e.g., failure of a particular component or even a particular kind of failure).

Figure 3 (at the end of the paper) shows part of a dynamic event tree generated for the example initiating event. The tool used to generate and display dynamic event trees starts from a normal state in which all the system components are operational and the process variables are within their nominal range. It then generates all possible configurations at the next time step (in this case 1 second) keeping track of all the possible states the process variables may be in at that point in time and in that configuration of the system components.

Figure 3 shows the tool window: the left pane shows a primitive representation of the event tree and the right pane shows the possible process states for the configuration and time step currently selected in the left pane with associated probabilities. Instead of showing the events between branching points (as it is usually done when displaying event trees), the representation of the event tree in the left pane shows the configuration of the control units at each branching point. The event(s) corresponding to a specific branch in the tree can be deduced by comparing the configurations to the left and to the right of the branch. For instance, if in the configuration at the left of a branch the BFV is in the OK/ABLE state and in the configuration to the right the BFV is STUCK, the event that has occurred along that branch must be that the BFV got stuck at its current position. At each branching point (or node) in the event tree, the label shows the state of the BFV and of the BC.

The event tree in the left pane is generated on demand. The top of the tree (displayed in the top-left corner of the left pane in Fig. 3) represents the normal configuration where all the system components are operational (OK/ABLE-OK, i.e., the BFV is in the OK/ABLE state and the BC is in the OK state). Whenever the user clicks one of the displayed nodes (branching points), the program generates all the possible configurations in which the system may evolve in the given time step. For example, there are 12 such possible distinct configurations after the first time step since the BFV can be in one of 3 states and the BC can be in one of 4 states. By repeatedly clicking and expanding the tree nodes, the user can explore any possible scenario in the tree. For instance, the (partial) event tree showed in Fig. 3 corresponds to one possible path (or failure scenario) leading to the level going below the LOW setpoint (dryout).

Boxes in the left pane of Fig. 3 highlight a possible failure scenario presented in detail in Table VII.

TABLE VII. Example failure scenario

Time	System Config.	Process State
t = 0	BFV: OK/ABLE BC: OK	$-0.17 \le x_n < 0.17$ $-1.587 \le E_{Ln} < 4.203$ $-100 \le C_{Ln} < 100$ $0 \le S_{Bn} < 30$
t = 1	BFV: OK/ABLE BC: OK	$-2.0 \le x_n < -0.17$ $4.203 \le E_{Ln} < 1000$ $-100 \le C_{Ln} < 100$ $70 \le S_{Bn} \le 100$
t = 2	BFV: OK/UNABLE BC: OK	$0.17 \le x_n < 2.5$ $-1.587 \le E_{Ln} < 4.203$ $-100 \le C_{Ln} < 100$ $0 \le S_{Bn} < 30$
t = 3	BFV: OK/UNABLE BC: LOSS/OUT	$-0.17 \le x_n < 0.17$ $-1.587 \le E_{Ln} < 4.203$ $-100 \le C_{Ln} < 100$ $0 \le S_{Bn} < 30$

t = 4	BFV: STUCK BC: LOSS/OUT	$-2.0 \le x_n < -0.17$ $4.203 \le E_{Ln} < 1000$ $-100 \le C_{Ln} < 100$ $0 \le S_{Bn} < 30$
t = 5	BFV: STUCK BC: DOWN	$x_n < -2.0$ (**LOW**) $4.203 \le E_{Ln} < 1000$ $-100 \le C_{Ln} < 100$ $0 \le S_{Bn} < 30$

At time t=0, both BFV and BC are in their operational state and all process variables are in their nominal range. At t=1, the level has decreased and the valve is opened more. At t=2, BFV (controller) becomes unable to recognize problems with BC; the level has increased and the BFV is closed. At t=3, BC experiences a loss of output. In this system configuration, according to Table VI., the BFV is closed completely. Because no repairs are allowed in the model, the system is now destined to fail low. At t=4, BFV becomes stuck (closed) and the level keeps decreasing. Finally, at t=5, the system fails when the level goes below the Low Level mark.

V. INCORPORATION OF DYNAMIC EVENT TREES INTO AN EXISTING PRA

Once a dynamic event tree for an initiating event has been generated, the tree can be incorporated into an existing PRA through the MAR-D feature of SAPHIRE using text files or graphically (Section V.B). While dynamic event trees have been generated above, it is recommended to import all trees into SAPHIRE as fault trees, do to the simplicity involved in the fault tree logical format. This may be done since the event trees may be represented as a series of AND events, which may be modeled as fault trees. The format for importing fault tree logical information is quite simple, and fault trees may also be easily connected to the existing PRA through appropriate placement of the model controller top event.

In this step, we must ensure that (1) the events in the dynamic event tree are appropriately named so that SAPHIRE is able to recognize the identical events in the dynamic tree as the same events in the rest of the tree, and, (2) the timing of the events is not lost when the dynamic event tree is incorporated into the existing model, so that timing information can be included in the resulting analysis. In the integration of the dynamic event trees into SAPHIRE, these objectives are achieved, respectively, by following a specific, consistent naming scheme when naming events and by time tagging the events to maintain exact timing information. Currently SAPHIRE does not have the ability to deal directly with timing information. Therefore, post-processing of the prime implicants resulting from SAPHIRE's analysis of the (partially dynamic) event tree may be necessary to eliminate outputs that violate the timing constraints. Again, through the MAR-D feature, minimal cut sets/ prime implicants may be exported into text files for post-processing. These files may then be re-imported into SAPHIRE for quantification if sufficient failure data are available.

V.A. Example Plant PRA

The model PRA to be used represents a simplified model of the example two-unit nuclear power plant from [4]. Both units are PWRs, each with a three loop design. Both units are rated at 2441 MW(th), or 788 MW(e). The Unit 1 reactor first started commercial operation in 1972. The PRA to be used was modeled using the SAPHIRE PRA code, which uses the ET/FT methodology.

The example plant PRA models include several initiating events, including loss of offsite power (LOSP), loss of coolant accidents (LOCA), and fire and seismic events. Each initiating event leads to an ET modeling how various plant systems attempt to respond to the initiating event. For example, Figure 4 shows part of the ET that models the plant's response to a turbine trip. A turbine trip could occur for several reasons, such as a loss of vacuum in the main condenser or if the turbine experiences overspeed. As can be seen by the ET, the plant's first response would be to scram the reactor through the reactor protection system (RPS). Failure of the RPS to scram the reactor will lead to an anticipated transient without scram (ATWS), and is modeled in a separate ET. After a successful reactor scram, the primary and secondary system safety relief valves (SRVs) must close (failure to do so leads to another ET modeling further plant actions in this scenario). With both the reactor scrammed and the SRVs closed, the Auxiliary Feedwater System (AFW) must then provide water to the SGs, maintaining a heat sink for the reactor. If the AFW system is unable to provided adequate water to the SGs, then the Main Feedwater (MFW) system is brought back online to provide cooling water to the SGs. Failure of both of these systems (not shown in Fig. 4) will require High Pressure Injection (HPI), and opening of the relief valves for feed and bleed, and could possibly lead to core damage. Successful operation of the auxiliary or MFW system will result in a safe condition for the plant. The turbine trip ET represents an ideal model to incorporate a model for a digital feedwater controller, as the SG water level is critical to the safety of the plant and water is added from either the AFW or the MFW system.

V.B. Integration of Dynamic Event Trees into PRA

With the model Digital Feedwater Control System imported into the SAPHIRE database through the MAR-D feature, it can then be connected with the existing plant PRA. This can be performed by appropriately placing the top event (or a transfer gate to this top event) of the

imported fault tree into the existing fault tree logic of the pertinent systems. In this case, as the dynamic model is for a digital feedwater controller, it should be tied to the MFW and/or AFW systems, which supply water to the steam condensers (in a typical plant PRA, the AFW system will be modeled in greater detail as it is typically used in emergency conditions while the MFW system is left offline). Proper placement of the dynamic model into the existing system will require some knowledge and understanding of the actual systems being modeled. For graphical input, an option for the event trees in Figs.3 and 4 would be to place the DFWC system between MFW and SEAL COOLANT FLOW and insert the tree in Fig.3 on the success branch of MFW.

Again, a proper naming scheme shared between the existing plant PRA and the dynamic model will ensure that SAPHIRE recognizes that components shared by both the existing PRA and the dynamic model are in fact the same. Recovery rules may be written to remove inconsistent event sequences, or to flag questionable sequences for post-processing.

The plant PRA can be solved and cut sets generated (although in this case, with the presence of the dynamic model, the term 'prime implicants' may be more appropriate). These cut sets can be viewed, and if necessary, exported from SAPHIRE for post-processing with additional tools. If desired, these cut sets can then be re-imported back into SAPHIRE. With acceptable failure data, the cut sets can be quantified to generate an overall failure frequency or probability. As with any SAPHIRE project, importance, uncertainty, and sensitivity analysis can then be performed.

VI. CONCLUSIONS

We have shown how the Markov reliability model of a digital feedwater level control system can be incorporated into an existing PRA by generating dynamic event trees from the Markov model, loading those trees into SAPHIRE, and linking them to the existing event tree.

The main outstanding issue is that dynamic event trees are generated with exact timing information attached to all paths and nodes, but existing PRA tools such SAPHIRE do not support dynamic methodologies and are not equipped to deal with timing information beyond simple ordering of events. One possible approach to deal with this problem is to time-tag the events in the dynamic event tree. The dynamic event tree can then be incorporated into the existing PRA, and the necessary analysis can be run. Because the PRA tool is unaware of timing and or time-tagging, the prime implicants resulting from the analysis may need to be post-processed to eliminate outputs that violate the timing constraints.

ACKNOWLEDGMENTS

The research presented in this paper was sponsored by the U.S. Nuclear Regulatory Commission (US NRC). The information and conclusions presented here in are those of the authors and do not necessary represent the views or positions of the US NRC. 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] T. ALDEMIR, D. W. MILLER, M. STOVSKY, J. KIRSCHENBAUM, P. BUCCI, A. W. FENTIMAN, and L. M. MANGAN, Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments, NUREG/CR-6901, U. S. Nuclear Regulatory Commission, Washington, D.C. (2006)

[2] K. RUSSELL, Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE), Version 6.0: System Overview Manual, NUREG/CR-6532, U.S. Nuclear Regulatory Commission, Washington, D.C. (1999)

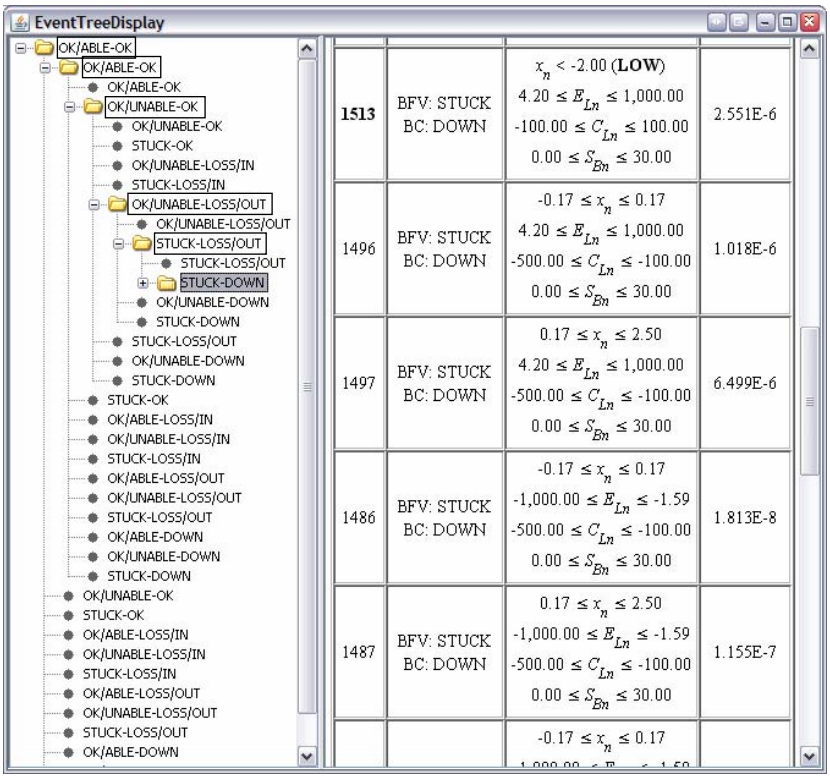
[3] P. BUCCI, J. KIRSCHENBAUM, T. ALDEMIR, C. L. SMITH and T. S. WOOD, "Constructing Dynamic Event Trees From Markov Models", M. STAMATALETOS and H. S. BLACKMAN (Eds.), *PSAM8: Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management, CD-ROM Version*, Paper # 318, Elsevier Science Ltd. (2006).

[4] *Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants*, U.S. Nuclear Regulatory Commission, Washington, D.C. (1990).

[5] KIRSCHENBAUM.JASON, M. STOVSKY, D. MANDELLI, P. BUCCI, T. ALDEMIR, D. MILLER, E. EKICI and S. ARNDT, "A Benchmark System for the Assessment of Reliability Modeling Methodologies for Digital Instrumentation and Control Systems in Nuclear Plants", *Proceedings NPIC&HMIT 2006*, Albuquerque, New Mexico (2006).

[6] C. S. HSU, *Cell-to-cell Mapping: A Method of Global Analysis for Nonlinear Systems*, Springer-Verlag, New York, NY (1987).

[7] S. GUARRO, M. YAU, and M. MOTAMED, Development of Tools for Safety Analysis of Control Software in Advanced Reactors, NUREG/CR-6465, U.S. Nuclear Regulatory Commission, Washington, D.C. (1996)



1513	BFV: STUCK BC: DOWN	x_n < -2.00 (**LOW**) 4.20 ≤ E_{Ln} ≤ 1,000.00 -100.00 ≤ C_{Ln} ≤ 100.00 0.00 ≤ S_{Bn} ≤ 30.00	2.551E-6
1496	BFV: STUCK BC: DOWN	-0.17 ≤ x_n ≤ 0.17 4.20 ≤ E_{Ln} ≤ 1,000.00 -500.00 ≤ C_{Ln} ≤ -100.00 0.00 ≤ S_{Bn} ≤ 30.00	1.018E-6
1497	BFV: STUCK BC: DOWN	0.17 ≤ x_n ≤ 2.50 4.20 ≤ E_{Ln} ≤ 1,000.00 -500.00 ≤ C_{Ln} ≤ -100.00 0.00 ≤ S_{Bn} ≤ 30.00	6.499E-6
1486	BFV: STUCK BC: DOWN	-0.17 ≤ x_n ≤ 0.17 -1,000.00 ≤ E_{Ln} ≤ -1.59 -500.00 ≤ C_{Ln} ≤ -100.00 0.00 ≤ S_{Bn} ≤ 30.00	1.813E-8
1487	BFV: STUCK BC: DOWN	0.17 ≤ x_n ≤ 2.50 -1,000.00 ≤ E_{Ln} ≤ -1.59 -500.00 ≤ C_{Ln} ≤ -100.00 0.00 ≤ S_{Bn} ≤ 30.00	1.155E-7
		-0.17 ≤ x_n ≤ 0.17	

Fig. 3. Display of part of the dynamic event tree

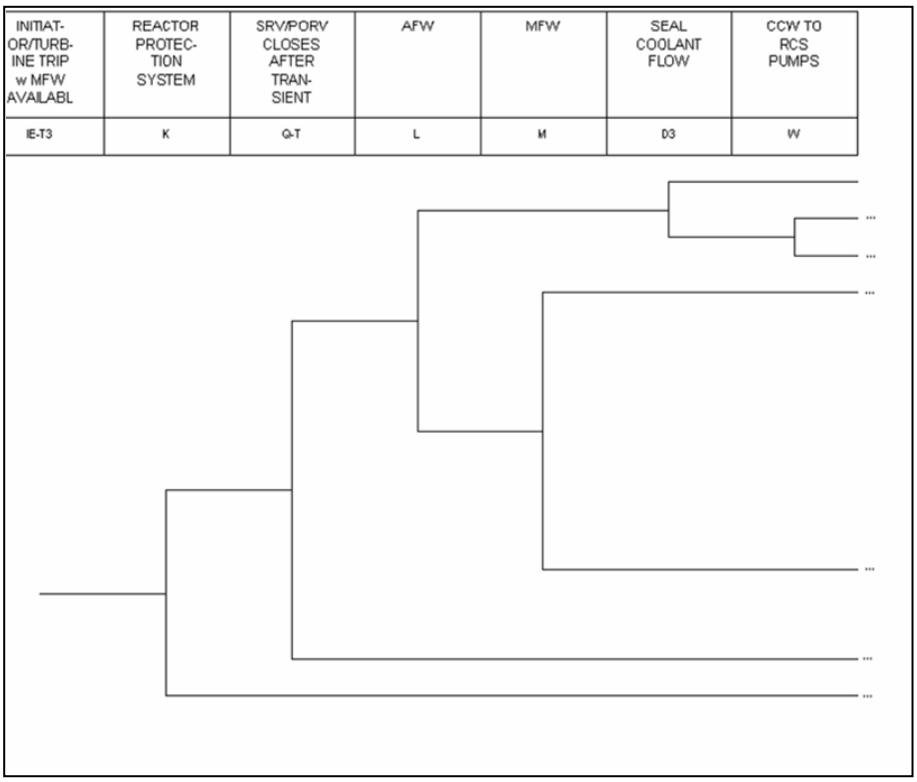


Fig. 4. Partial event tree for turbine trip