Markov/CCMT Modeling of the Benchmark System and Incorporation of the Results into an Existing PRA

D. Mandelli[a1], T. Aldemir[a], P. Bucci [b], L. A. Mangan[a], J. Kirschenbaum[b],M. Stovsky[b], E. Ekici[c], S. A. Arndt[d]

[a]The Ohio State University, Nuclear Engineering Program, Columbus, OH, USA
[b]The Ohio State University, Dept. of Computer Science & Engineering, Columbus, OH, USA
[c]The Ohio State University, Dept. of Electrical & Computer Engineering, Columbus, OH, USA
[d]U.S. NRC, Office of Nuclear Reactor Regulation, Washington, D.C. USA

Abstract: In the reliability modeling of digital control systems, conventional methodologies based on the event-tree/fault-tree (ET/FT) methodology may not represent adequately the statistical dependence between failure events in certain digital systems. Dynamic methodologies can be considered as an important alternative to overcome this limitation. The Markov/CCMT (Cell-to-Cell Mapping Technique) has been proposed as a dynamic methodology for the probabilistic risk assessment (PRA) of digital control systems. The application of Markov/CCMT is illustrated using the digital control system for a feedwater system of a pressurized water reactor (PWR). Discrete hardware/software/firmware states are defined and transitions between these states are deduced from the control logic of the system, as well as from the failure modes and effects analysis (FMEA) performed on each component. Markov/CCMT is used to represent the dynamics of the system as the probability of transitions between process variable magnitude intervals (cells) that partition the state space. The resulting Markov model is converted into dynamic event trees which then are incorporated into an existing ET/FT based PRA of a PWR using the SAPHIRE code.

Keywords: Digital systems, probabilistic risk assessment, dynamic PRA, Markov.

1. 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 modeling is that it should be possible to incorporate the resulting model into an existing probabilistic risk assessment (PRA) for the overall plant. 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 (SG) digital feedwater control system (DFWCS) 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.

2. SYSTEM UNDER CONSIDERATION

A detailed description of a benchmark DFWCS 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 SGs within a given range around the level setpoint by controlling a feedwater pump (FP), a main feedwater regulating valve (MFV) and

1 Mandelli Diego, mandelli.diego@gmail.com, The Ohio State University, Nuclear Engineering Program, 201 West 19th Avenue, Columbus, OH 43210

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 benchmark DFWCS 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 DFWCS 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 MWth 10 seconds after reactor shutdown
5. Feedwater flow is at nominal level
6. Off-site power is available

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 a chosen example initiating event.

3. MARKOV/CCMT MODELING OF THE 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 $V_j\,(j = 1,\ldots,J)$ 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 $t = k\Delta t\ (k = 0, 1,\ldots)$ with the system components (such as pumps, valves, and controllers) having a component state combination $n = 1,\ldots,N$. 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,
- the control laws, and,
- the 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.

3.1. 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 define two Top Events:

- $x < -2$ feet (Low Level)
- $x > +2.5$ feet (High Level).

3.2. CONTROL LAWS MAND SYSTEM DYNAMICS

Equations (1) – (9) describe the dynamic behavior of the system and the control laws as they are applied to the example initiating event:

$$\text{Water Level } (x_n)\text{:} \quad \frac{dx_n}{dt} = A(f_{wn} - f_{sn}) \tag{1}$$

$$\text{Water Level Error } (E_{Ln})\text{:} \quad \tau_5 \frac{dE_{Ln}}{dt} = r_n - C_{Ln}(t) \tag{2}$$

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

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

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

$$\text{BFV Position \% } (S_{Bn})\text{:} \quad S_{Bn} = \begin{cases} \sigma_{Bn} & \text{main or backup CPU up} \\ \eta_{Bn} & \text{both main and backup CPU down} \end{cases} \tag{6}$$

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

Here, 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. In Eq.(1), the water flow rate f_{wn} is 0 if the 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 the 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.,

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

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

3.3. PARTITIONING OF THE STATE SPACE

The dynamics of the system are 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 using the following rules:

- all cells V_j must be disjoint and the must cover 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 1- 4 below show the actual partitioning scheme used for each process variable:

Table 1: 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 2: 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 3: 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 4: 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.

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.

3.4. DEFINITION OF THE HARDWARE/FIRMWARE/SOFTWARE STATES

The components that need to be considered for the example initiating event [1, 3] are the BC and BFV controller. Then the relevant states for the example initiating event are:

1. BC operating and BFV controller operating
2. BC loss of inputs and BFV controller operating
3. BC down and BFV controller OK
4. Freeze
5. Arbitrary output
6. 0 vdc
7. Stuck

Figure 1 shows transition diagram for the example initiating event.

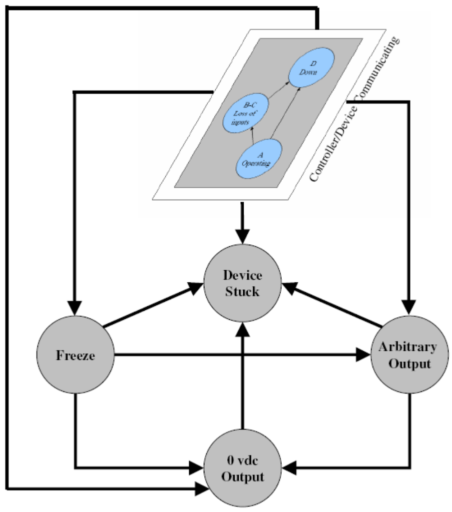

Fig. 1: Transition diagram for the example initiating event

3.5. DETERMINATION OF CELL-TO-CELL TRANSITION PROBABILITIES

In principle, the evolution of any dynamic system depends on three factors:

- The dynamic equations of the system
- The control laws of the control system
- The state of each component

Consequently, as also stated in Section 3, the probability of the system to transit from a cell $V_{j'}$ to cell V_j also depends on these three factors. In the Markov/CCMT, the first two factors are accounted for in the transition probability $g(j|j',n',k)$ while the third one is captured by the transition probability $h(n|n',j' \to j)$.

The cell-to-cell transition probabilities $g(j|j',n',k)$ are conditional probabilities that the controlled variables are in the cell V_j at time $t = (k+1)\Delta t$ given that:

- the controlled variables are in the cell $V_{j'}$ at time $t = k\Delta t$, and,
- the system components are in component state combination $n(k) = n'$ at time t.

As shown in [3] the $g(j|j',n',k)$ can be determined from

$$g(j \mid j',n',k) = \frac{1}{v_j} \int_{V_j} dv' e_j \{\tilde{x}_{k+1}(x',n',k)\} \tag{10}$$

$$e_j\{\tilde{x}_k\} = \begin{cases} 1 \to \tilde{x}_k \in V_j \\ 0 \to otherwise \end{cases} \tag{11}$$

where:

- $v_{j'}$ is the volume of the cell $V_{j'}$
- $\tilde{x}_k$ is the arrival point in the controlled variable state-space (CVSS) at time $t = (k+1)\Delta t$
- x' is the starting point in the cell V_j' at time $t = k\Delta t$
- n' is the component state combination at time $t = k\Delta t$.

The algorithm to determine $g(j|j',n',k)$ is the following:

1. Partition a cell j' into $N_{k,j'}$ subcells
2. Choose the midpoint of each subcell
3. Integrate the equations which describe the dynamic behaviour of the system (e.g., for the DFWCS the equations are presented in Section 3.2) over the time interval $k\Delta t \leq t \leq (k+1)\Delta t$ under the assumption that the component state combination remains n' at all times during $k\Delta t \leq t \leq (k+1)\Delta t$.
4. Observe the number of arrivals $N_{k+1,j}$ in cell j at time $t = (k+1)\Delta t$ (i.e., $\tilde{x}_{k+1}(x',n',k)$)
5. Obtain $g(j|j',n',k) = N_{k+1,j} / (N_{k,j'})$

3.6. DETERMINATION OF HARDWARE/FIRMWARE/SOFTWARE STATE TRANSITION PROBABILITIES

The stochastic behavior of hardware/software/firmware is represented through $h(n|n',j' \to j)$, which is the probability that the component state combination at time $t = (k+1)\Delta t$ is n, given that:

- $n(k) = n'$ at $t = k\Delta t$, and
- the controlled variables transit from cell $V_{j'}$ to cell V_j during $k\Delta t \leq t < (k+1)\Delta t$.

For components with statistically independent failures, the probabilities $h(n|n',j' \to j)$ are the products of the individual component failure or non-failure probabilities (for all the M components) during the mapping time step from $k\Delta t$ to $(k+1)\Delta t$, i.e.,

$$h(n \mid n', j' \to j) = \prod_{m=1}^{M} c_m (n_m \mid n'_m, j' \to j) \tag{12}$$

where $c_m(n_m|n'_m,j' \to j)$ is the transition probability for component m from the combination n'_m to n_m within $[k\Delta t, (k+1)\Delta t]$ during the transition from the cell $V_{j'}$ to V_j.

As an example of determining the $h(n|n',j' \to j)$, suppose that the transition from the configuration n'_m to n_m involves the transition from the “Freeze” state (i.e., State 4) to the “Arbitrary Output” state (i.e., State 5) with a failure rate equal to λ_{45}. Since there are only two components (i.e., BC and BFV plus controller), M = 2. Also, since the controller is in the Freeze state, BC is down and the system maintains the BFV demand at the most recent correct value, which implies that the controller remains in the same state with probability $h(n|n',j' \to j) = \lambda_{45}\Delta t$.

3.7. CONSTRUCTION OF THE MARKOV MODEL

The probability $p_{n,j}(k+1)$ (j=1,…,J) that at $t = (k+1)\Delta t$ the controlled variables are in cell V_j and the component state combination is n is the sum of $N \times J$ terms where N is the total number of component state combinations. Each of these probabilities is the product of two factors:

- The probability for the system to transit from the cell $V_{j'}$ and component state combination n' to cell V_j and component state combination n (i.e., $q(n,j|n',j',k)$)
- The probability that the system is in the initial cell $V_{j'}$ and state combination n' (i.e., $p_{n,j}(k)$)

Thus:

$$p_{n,j}(k+1) = \sum_{n'=1}^{N} \sum_{j'=1}^{J} q(n, j \mid n', j', k) p_{n',j'}(k) \tag{13}$$

The elements of the transition matrix $q(n,j|n',j',k)$ are functions of both:

- The cell to cell transition probability $g(j|j',n',k)$
- The component state transition probabilities $h(n|n',j',j)$, i.e.,

$$q(n, j \mid n', j', k) = g(j \mid n', j', k) h(n \mid n', j' \to j) \tag{14}$$

Since cells V_j cover the whole CVSS and N includes all the possible state combinations:

$$\begin{gathered} \sum_{n'=1}^{N} \sum_{j'=1}^{J} q(n, j \mid n', j', k) = 1 \\ \sum_{n'=1}^{N} \sum_{j'=1}^{J} p_{n',j'}(k) = 1 \end{gathered} \tag{15}$$

The grouping of several components into macro-components can be useful to decrease the number of possible state combinations, which can be very large for systems that involve a large number of components. Note that for autonomous processes, the transition matrix $q(n,j|n',j',k)$ has to be constructed only once and not at each step throughout the duration of the mission of the system.

3.8. GENERATION OF EVENT TREES FROM THE MARKOV MODEL

From Sections 3.4 – 3.7 it is possible to identify all the possible trajectories (i.e., a list of all the possible event sequences) given the set of initial conditions. Starting from these initial conditions, the algorithm branches through discrete time steps such that each level of branching in the tree represents all the possible states in which the system may be after a given time interval (see Fig.2). Branching stops any time a branch reaches a "sink" state (i.e., a state from which the system cannot move out) or the probability associated with the branch is below a chosen threshold. It is also possible to stop after a certain amount of system time has elapsed or, equivalently, once the branching has reached a chosen depth in the tree.

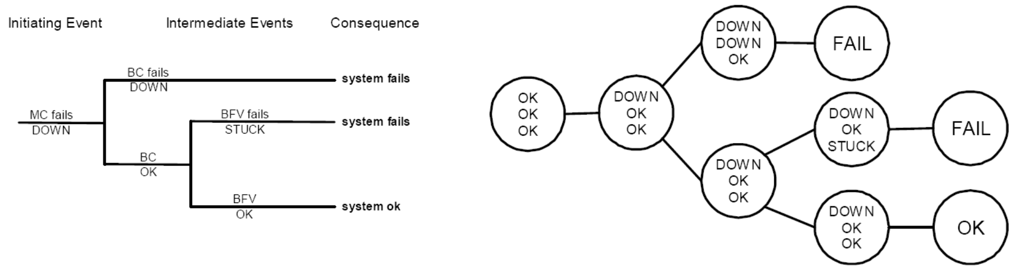

Fig. 2: Dynamic Event Tree and tree data structure

The DET is represented by a tree data structure (see Fig.2). A tree data structure is composed of "nodes" (where information is stored) and "links" that connect the nodes. The nodes in the tree data structure correspond to the branching points in the DET and the links represent the branches. Figure 3 shows part of a DET generated for the system.

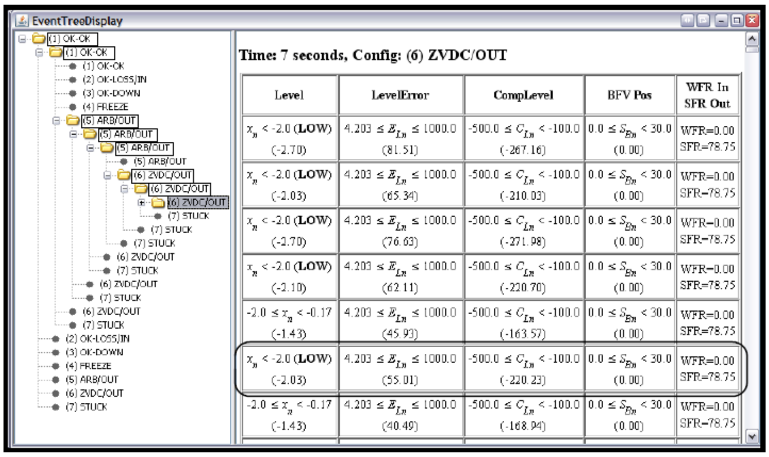

Fig. 3: Display of Part of the Dynamic Event Tree

Table 5 summarizes the number of failure scenarios exhibited by the system as a function of the depth of the tree, i.e., the length of time for which the system is analyzed. The percentage of the total number of scenarios for a given depth that lead the system to fail LOW, fail HIGH, and not fail is included in parentheses. In this analysis, 27 sample points on a regular grid within the process state space cell where each variable is in its nominal range were used as initial conditions. The number of possible scenarios grows with the number of sample points employed.

Table 5: Number of Failure and Non-Failure scenario

Time (in seconds) (Depth of DET)	Number of LOW failure scenarios	Number of HIGH failure scenarios	Number of scenarios without failure
1	0 (0.0%)	0 (0.0%)	243 (100.0%)
2	0 (0.0%)	0 (0.0%)	1,242 (100.0%)
3	530 (10.8%)	0 (0.0%)	4,384 (89.2%)
4	1,480 (9.3%)	0 (0.0%)	14,439 (90.7%)
5	4,999 (10.2%)	186 (0.4%)	43,727 (89.4%)
6	14,811 (10.2%)	2,518 (1.7%)	127,292 (88.0%)
7	47,881 (11.5%)	6,531 (1.6%)	362,153 (86.9%)
8	140,644 (11.9%)	18,559 (1.6%)	1,022,695 (86.5%)
9	411,240 (12.3%)	50,259 (1.5%)	2,871,468 (86.2%)
10	1,126,498 (12.0%)	143,922 (1.5%)	8,091,530 (86.4%)

As can be seen from Table 5, there are a large number of possible scenarios. The majority of scenarios for each DET depth fail to lead the system to failure within the chosen time limit. However, there are still a substantial number of scenarios leading to failure due in part to the presence of the ARB/OUT state. In this state the BFV receives an arbitrary signal from the controller. Sensitivity of the results to initial conditions were explored using 3 different values of the BFV position, one for each of the 3 intervals in which the BFV position has been partitioned (see Table 4).

Another observation is that the number of LOW failure scenarios is always much larger than the number of HIGH failure scenarios. This is due to the existence in the model of a state (ZVDC/OUT) in which the BFV is closed. Whenever the system enters this state, the valve is forced to close and never reopens. Thus the system is bound to fail LOW. Finally, Table 5 shows that, given the stated initial conditions, the minimum time necessary for the system to fail LOW is 3 seconds and the minimum time for the system to fail HIGH is 5 seconds.

3.9. 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 (or equivalent features in other ET/FT analysis tools) using text files or graphically [3]. While DETs have been generated above, it is recommended to import all trees into SAPHIRE as fault trees (FTs), do to the simplicity involved in the FT logical format. This may be done since the DETs may be represented as a series of AND events, which may be modelled as FTs.

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:

- 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
- 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 DETs 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 sequence ordering 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. Some post-processing of the cut sets may be performed internally in SAPHIRE through the FT Recovery Rules editor. The Recovery Rules editor allows the user to search through the cut sets and edit or remove inconsistent cut sets.

4. CONCLUSION

This paper shows how it is possible to model digital control systems for PRA purposes using Markov/CCMT. In particular, it shows how it is possible to perform the analysis starting from a detailed description and understanding of the system under consideration. The inputs are the control laws, the system topology, the control logic and the analysis of the failures and effects performed for all the components of the system.

The analysis of the system is performed merging two separate models, one for each of the two types of interactions. Type I interactions take into account the dynamic of the system and how the process and the controller affect each other. These are modeled using the CCMT which describes the dynamics of the process through transitions between cells that compose the CVSS. Type II interactions, on the other hand, take into account the interactions among the components of the controller. These are modeled thorough Markov transitions diagrams, one for each component.

Discrete hardware/software/firmware states are defined and transitions between these states are deduced from the control logic of the system, as well as from the failure modes and effects analysis performed on each component.

Starting from a finite set of initial points distributed in the CVSS, the Markov/CCMT follows the trajectories of these points. At each time step and for each point, new points are generated, each having the same location in the CVSS but different system configuration in accordance with the Markov models built previously.

The generated trajectories are equivalent to event sequences which can be simply converted into dynamic event trees. These event trees can be incorporated into an existing ET/FT based PRA of a PWR using the SAPHIRE code.

Acknowledgements

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. A. Mangan, "Current State of Reliability Modelling 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. Russel, 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", NUREG-1150, U.S. Nuclear Regulatory Commission, Washington, D.C. (1990).

[5] D. Mandelli , T. Aldemir, J. Kirschenbaum, P. Bucci, D. W. Miller, M. Stovsky, E. Ekici, S. A. Arndt, "A Benchmark System for the Reliability Modeling of Digital Instrumentation and Control Systems", these proceedings.

[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)