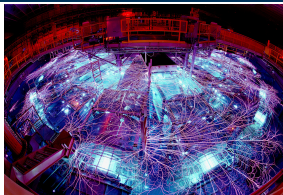


Exceptional service in the national interest



Sandia
National
Laboratories

SAND2016-11365C



Multiple Simulators in the ADAPT Framework

Versatile Discrete Dynamic Event Tree Creation

Zachary Jankovsky^{*+}, Matthew Denman⁺, Tunc Aldemir^{*}



THE OHIO STATE
UNIVERSITY

*The Ohio State University +Sandia National Laboratories

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-xxxxC.



Acknowledgments

- This work was supported by the Laboratory Directed Research and Development program from the Sandia National Laboratories.

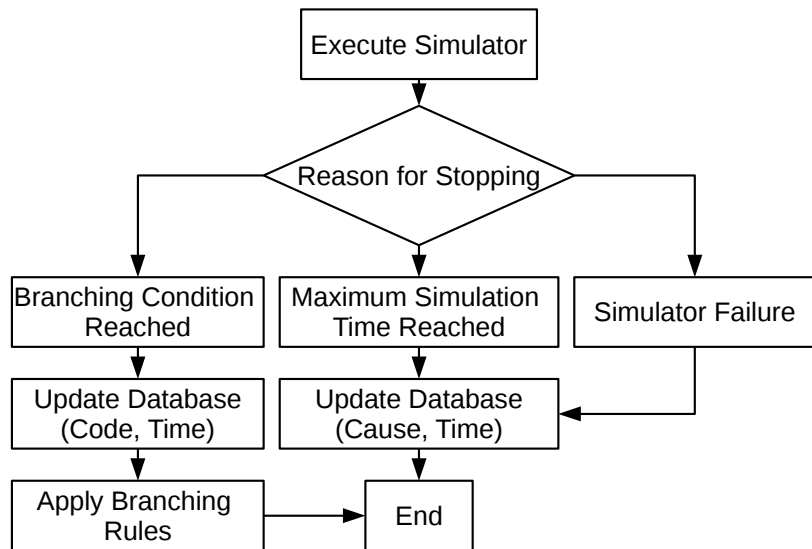
- Background
 - Discrete Dynamic Event Trees (DDETs)
 - DDET Tools
- Motivation
 - Why Multiple Simulators?
- Code Modifications
 - Branching Rules File
 - Wrapper File
- Sample Problem
- Results
- Concluding Remarks

- Traditional Probabilistic Risk Assessment (PRA)
 - Combined fault tree / event tree analysis
 - Binary success / failure branching
 - Does not traditionally capture timing effects
 - May be nested to capture time dependency, e.g.:
 - E1: Action performed before $t=2hr$ success / failure*
 - E2: (given E1 failure) Action performed between $t=2hr$ and $t=3hr$ success / failure*
 - E3: (given E2 failure) Action performed after $t=3hr$ success / failure*
 - Prescribed order of events
 - Practitioner may miss important sequences or include non-realistic sequences

- Dynamic PRA (DPRA) using discrete dynamic event trees (DDETs)
 - Begin root branch with simulator plant state file at $t=0$
 - Typically just before initiating event occurs
 - Tree grows according to branching rules, e.g.:
 - Branch on scram success / failure when demanded
 - Branch on number of high pressure injection trains available when demanded
 - Branching triggers set in simulator input (e.g. MELCOR)
 - Simulator programmed to stop either at predetermined time intervals (ADS) or upon reaching a trigger (ADAPT)
 - Branching occurs only when appropriate, and always when appropriate (according to simulator parameter values)

- Accident Dynamic Simulator (ADS), 1996
 - Hard-coded link to RELAP5/TRACE for plant model and
 - Hard-coded link to Information, Decision, and Action in a Crew (IDAC) for operator model
- Reactor Analysis and Virtual Control Environment (RAVEN), INL, 2012
 - Control system for RELAP-7 plant model
 - Also linked to RELAP5-3D, MAAP, MELCOR
 - Single branching code for DDET operation
- Analysis of Dynamic Accident Progression Trees (ADAPT), SNL/OSU 2006
 - Initially linked to MELCOR
 - Intended to be simulator-flexible
 - Subsequently linked to RELAP5, MAAP4, SAS4A
 - Now allows arbitrary number of arbitrary branching codes

ADAPT Branch Behavior, Current

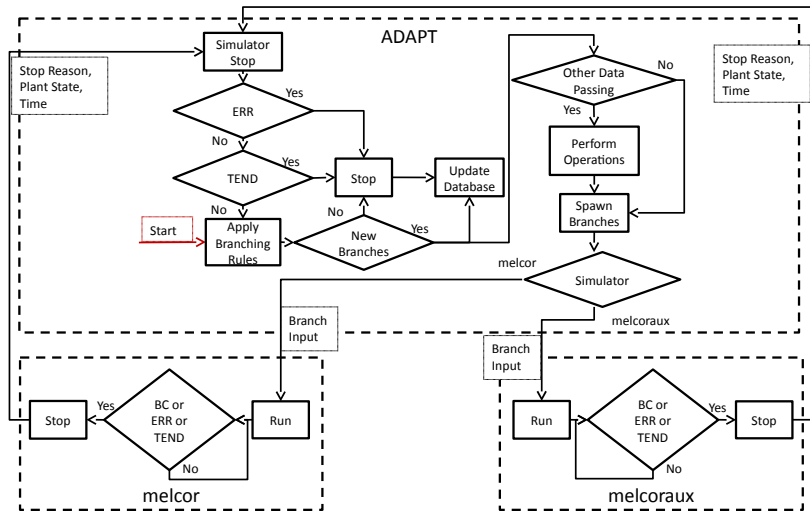


- Mechanistic handling of events
 - e.g. seismic-induced component failure
 - Traditional: assign component failure distribution
 - Mechanistic: link to structural mechanics code, branching on uncertain loading and capacity
- Combine temporally-separated steps of the problem
 - May create seamless Level 1/2/3 dynamic analyses (e.g., MELCOR and MACCS)
 - Allows accounting for the impact of timing of events on final consequence measures
- Detailed analysis of select phenomena
 - e.g. control room habitability
 - Reach decision point in system-level code
 - Run habitability code, branching on ventilation status and operator tolerance of dose
 - Return to system-level code with results of each branch

- Principle: each branch runs one and only one simulator
- Database
 - Add indication of simulator run to branch information table
- Branching Rules File
 - Generalize from 1 to ∞ template simulator input files, e.g.:
 - Old: *InputFile dsc.cor.tpl*
 - New: *InputFile 1 dsc.cor.tpl*
 - Enumerate simulators and their executables, e.g.:
 - *SimulatorExecutable 1 melcor*
 - *SimulatorExecutable 2 melcoraux*
 - Define simulator to run after each branching condition, e.g.:
 - *BranchingSimulator 90017 2*

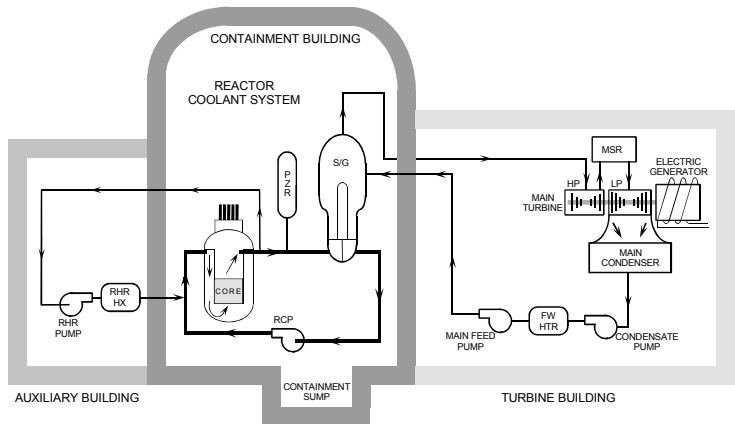
- Wrapper File
 - Script of actions to perform for each branch
 - Simulator now passed as input argument
 - If-then-else statements used to determine simulator-dependent behavior
 - e.g. different execution syntax and post-processing
 - Preparation of inputs for next branch dependent on simulator to be run
 - Informed by interpretation of Branching Rules File
 - Must consider all directions of transfer of control from Branching Rules File
 - e.g. 1 to 1, 1 to 2, 2 to 2, 2 to 1

ADAPT Branch Behavior, Expanded



Sample Problem (1/3)

- MELCOR Model Split:
 - Containment / Turbine Buildings
 - Auxiliary Building



- Residual Heat Removal System (RHR) Interfacing System Loss of Coolant Accident (ISLOCA)
 - Both isolation motor-operated valves (MOV) held open in cyber-induced controller failure
 - Sticking open due to mechanical failure occurs only on demand
 - Cyber-induced failure may occur at any time
 - May lead to large pipe break outside containment
 - May lead to RHR heat exchanger tube rupture, introducing reactor coolant system (RCS) water into component cooling water (CCW)
 - Possibility of failing CCW, upon which many engineered safety features depend
 - May lose safety injection (SI) pumps

Sample Problem (3/3)

- MELCOR model split to allow flexibility in handling possible contamination of auxiliary building
- For sample analysis, allowing only one-way transfer of control
 - Containment / Turbine model runs to maximum simulation time, then Auxiliary model runs
 - Disregards potential feedback of RHR leak on systems in other model
 - In some scenarios, this may be appropriate
 - Contamination may impede operator actions to isolate and mitigate ISLOCA
- Final analysis will support two-way transfer of control
 - Single MELCOR model
 - Linked habitability model (RADTRAD)

- Running each simulator:

```
if [ $THIS_EXECUTABLE = "melcor" ]; then
echo E | nice $MEL_ROOT/$EXE $FILE_PRI
else
if [ ! -f dscaux.rst ]; then
cp $MEL_ROOT/ss2_adapt.gen $MEL_ROOT/ss2_aux.gen \
$MEL_ROOT/ss2_materials.gen $MEL_ROOT/ss2_misc.gen \
$MEL_ROOT/ss2_RNdch.gen $MEL_ROOT/ss2_RNtrack.gen .
$MEL_ROOT/$EXE_MEG $FILE_AUX
fi
echo E | nice $MEL_ROOT/$EXE $FILE_AUX
fi
rc=$?
if [ $rc -ne 0 ]; then
exit $rc
fi
```

Results (2/2)

- Total of 851 branches, 328 in the auxiliary model
 - 384 unique sequences
- Simulator marked in database for each branch
 - Shown in visualization tool

(54021) [$p=6.6e-17$] sb 2 melcor
(code: 90015)
(simtime: 86400.10)
(sec: 197084)

(54390) [$p=6.6e-17$] sb 1 melcoraux
(simtime: 86400.40)
(sec: 13762)

(54022) [$p=3.99896e-09$] sb 1 melcor
(logical_fail: 1)
(simtime: 20425.40)
(sec: 70705)

- Recent Expansion of ADAPT
 - **Multi-simulator support**
 - Dynamic importance measures
 - Slicing of DDET conditional on user-input rules
- Use of multi-simulator DDETs with ADAPT will expand the insights that may be gained using DPRA
- Addresses advanced PRA objectives*:
 - *Reduces reliance on unnecessary modeling simplifications and surrogates (i.e., more phenomenological)*
 - *Makes process and results more scrutable*
 - *Leverages advances in computational capabilities and technology developments, but is computationally tractable*
 - *Allows for ready production of uncertainty characterization*

* Don Helton, *Scoping Study on Advancing Modeling Techniques for Level 2/3 PRA*, USNRC, May 2009

- Tuesday - Current Topics in Probabilistic Risk Analysis I
 - 1:00p: N. Martin, Pruning of Discrete Dynamic Event Trees Using Density Peaks and Dynamic Time Warping
 - 1:25p: M. Denman, Preliminary Cyber Informed Dynamic Branch Conditions for Analysis with the Dynamic Simplified Cyber MELCOR Model
 - 2:40p: Z. Jankovsky, Dynamic Importance Measures in the ADAPT Framework
- Wednesday - Computational Methods
 - 10:05a: Z. Jankovsky, Conditional Tree Reduction in the ADAPT Framework
 - 10:30a: Z. Jankovsky, Extension of the ADAPT Framework for Multiple Simulators
- Wednesday - Nuclear Installations Safety: General I
 - 1:00p: J. Cardoni, Severe Accident Modeling for Cyber Scenarios