INL/CON-14-33680
PREPRINT

INCORPORATING DYNAMIC 3D SIMULATION INTO PRA

PSA 2015

Steven Prescott, Curtis Smith, Ramprasad Sampath

May 2015

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

INCORPORATING DYNAMIC 3D SIMULATION INTO PRA

Steven Prescott
Curtis Smith
Ramprasad Sampath

Idaho National Lab: P.O. Box 1625, Idaho Falls, ID, 83415, Steven.Prescott@inl.gov

Through continued advancement in computational resources, development that was previously done by trial and error production is now performed through computer simulation. These virtual physical representations have the potential to provide accurate and valid modeling results and are being used in many different technical fields. Risk assessment now has the opportunity to use 3D simulation to improve analysis results and insights, especially for external event analysis. However, the static nature of traditional PRA methods hinders the direct use of time dependent dynamic simulations. This paper first briefly discusses how 3D simulation methods can be used to improve the modeling approach. In addition, we show how a state based PRA model, based on discrete event simulation, is equivalent and in some cases better than results from traditional fault tree evaluation methods. Finally, how to successfully incorporate physics based 3D simulation events at runtime to enhance overall results.

I. INTRODUCTION

Through physics and complex mathematical models, we have a good understanding how our world around us behaves. However for anything other than small problems, these models become complicated when attempting to represent reality. The field of computational physics applies numerical approximations and decomposes a problem into a large number of simple mathematical operations that can be solved using a computer. (Ref. 1)

Many fields of study use computational physics to do calculations, from protein folding for medicine to realistic effects in visualization. With the expansion of computation power and distributed computing, larger and more complex problems are able to be solved. We have used these physics methods and tools to analyze flooding events and dynamically determine component failures. These tools have then been incorporated into PRA analysis software to improve results for a tsunami external event analysis. The same principles can also be used for other external events.

By using these simulation tools, the modeler only has to determine the likelihood of an event without having to also predict the results of that event. The use of 3D simulations not only reduces errors from unforeseen secondary effects, but also determines when or sequence of any failures.

To make full use of time dependent simulation results, a modification to traditional PRA calculation methods is needed. A state based PRA modeling method was implemented based on the same principles as three-phased discrete event simulation (Ref 9). It can fully represent traditional models, and seamlessly incorporate events from a continuous physics based simulation.

For this example we started with the same design as the "Demo" project from SAPHIRE, consisting of 16 basic events and 2 fault trees, and then added in tsunami initiating events. The four pumps and two diesel generators were also included in the 3D model to fail with water contact during the simulation.

II. PHYSICS BASED 3D SIMULATION SETUP

A 3D Simulation for risk analysis consists of several parts, the 3D facility model; components in the model; the events of concern on those components; and a scenario to run against the model.

II.A. 3D Facility Model

A 3D Facility model is used by the simulation for collision data and object properties during the simulation. It defines items that either exist or are planned for, such as terrain, buildings, tanks, openings, etc. The closer these objects match to their real world counterparts, the more accurate the simulation results.

The terrain is a key factor for any external flooding scenario such a dam break, flash flood, or tsunami. A planed terrain can be constructed by a modeler, or if it is an existing area, automated tools can be used. For example, the Google Altitude API can be used to query

points for a defined area (Ref. 4). This point cloud can then be converted into a polygon mesh and used as a base for the rest of the modeling.

Deferring levels of resolution of a facility need to be made in order to optimize the simulation calculations. A lower resolution but large area model is needed for full facility simulations. Then any buildings of interest need a higher detailed interior model with all doors, vents, and physical structures. Items need to be assigned physical properties such as mass or anchoring pins so that they can react correctly to situations in the simulation.

II.B. Model Components and Events

All key components and the events of concern must be included in a model. Events of concern could be water contact, water submersion level, impacts, or movement. Data such as impact force, pressure, or position could be gathered or fed back to a parent application during a simulation.

For example, a water resistant electrical pump would be modeled and linked with water submersion event. When water comes in contact with the pump, that data is communicated back to the attached application with the time of the event.

The events done for this test were only water contact events. However, with simulated physics, other factors such as debris impacts, flow obstructions, or erosion could also be modeled.

II.C. Scenario Simulations

Once a model is built and components with events added, a scenario can be constructed. Many different scenarios can be developed using the same base model. A scenario consists of a timeframe, and physics based influences added to a model, such as rain fall, a wave machine, or wind. When a simulation starts, physics algorithms determine all consequential or secondary reactions over the timeframe of the simulation.

III. SIMULATION TOOLS

There are different 3D physics simulation engines using a variety of methods. Each has its advantages or disadvantages depending on the intended use.

III.A. Houdini FX

For this research, we initially started with a software package called Houdini FX. (Ref. 6) This application is a dynamic and widely used 3D simulation environment for visual effects. It also has an API for custom modifications which allowed us to communicate with it through other applications during each frame of the simulation. This feature makes it useful for incorporation into risk analysis modeling since the scenario evolution can be controlled (e.g., a failures can be triggered) during the calculations being performed.

The Houdini package worked well for smaller simulation such as water flow inside of a room. However, it had two issues with larger simulations. First, the solver that is used in Houdini was grid based Fluid In Particle (FLIP) method. Particle-In-Cell (PIC) PIC/FLIP based solvers are extensively used in visual effects and produces visually interesting dynamic motion because it uses custom particle advection methods to combat numerical diffusion problems resulting in diffuse fluids. However, using a Smoothed Particle Hydrodynamics (SPH) solver with physics based modifications to handle boundaries guarantees conservation of mass with computation of pressure from weighted contribution of neighboring particles (Ref. 2).

When Houdini's FLIP solver was used to generate wave like a tsunami, the wave would quickly lose momentum and diffuse out. We tried to overcome this numerical diffusion by using higher resolution grids. But, in using higher resolution grids, we ran into the second problem. The Houdini engine was not able to support the memory requirement needed to run the larger simulations. Moreover even running higher resolutions to a maximum of what could be handled, this work around still did not produce a wave which preserved energy and had excessive numerical diffusion.

III.B. Neutrino Solve Engine

To combat the problems we encountered with Houdini's FLIP fluid solver, we decided to try "Neutrino." The Neutrino fluid solver [Developed by Neutrino Industries from initial work done by Nadiar Akinci's PhD thesis (Ref 7)] is based on Smooth Particle Hydrodynamics with a pressure solver to handle incompressible fluids. The Neutrino fluid solver also factors in accurate boundary handling, and adaptive time stepping to help to increase accuracy and calculation speed (Ref. 3).

For this research, we collaborated with Neutrino Industries and they provided the use of their solve engine and made custom modifications to the code base to help with analysis. Neutrino was able to handle not only the memory requirements needed for large simulations, but provided more accurate fluid movement with less numerical diffusion which preserved the solitary tsunami wave momentum required for our simulations.

Neutrino also provides a variety of tools to measure parameters during a fluid simulation. This includes the wave height at a specific point, the average pressure and average velocity in a certain area/volume, as well as the flow rate across a certain area/volume.

IV. SIMULATION TESTS

To test the capabilities of the simulation tools for PRA purposes, we combined both a large scale problem and a smaller subset. The first consisted of a large tsunami wave inundating a facility and then dynamic internal flooding of a room caused by random tsunami heights. Components and their failure monitoring were included in the smaller simulation.

IV.A. Facility Model

To begin, we needed to model a facility. As a starting point we used a publicly available reactor facility model that was based on Fukushima (Ref 8). This base model was then converted into a collision model for the main facility simulations. As testing evolved, we then used a custom application that queried Google's Altitude API (Ref. 4) to generate a terrain map for the facility. (See Appendix A.I) The facility was then added to the terrain map to provide more accurate simulation behavior. (See Figure 1)

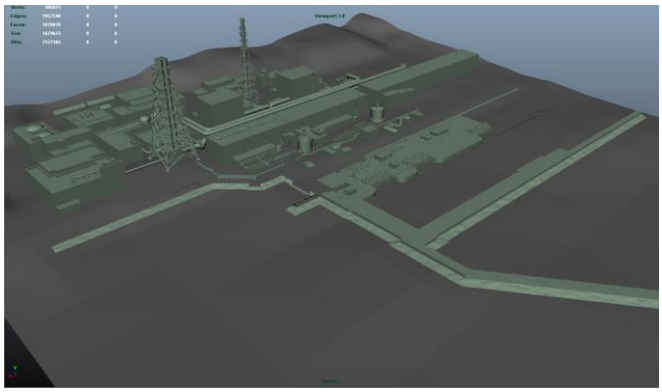

Figure 1. Nuclear facility model on a terrain map.

IV.B. Tsunami Simulation

The tsunami simulation consisted of a facility modeled on the coast, similar to Fukushima incident. In the beginning a small slice of the facility with one reactor was used for testing. This progressed to using the south half of the facility including support buildings and terrain. A boundary container was constructed around the desired facility area and into the ocean to contain the water within the desired scope. This container was filled with water particles using volumetric operations and Boolean operations to remove any particles inside solid geometry. This served as a start point of the simulation (i.e., still water). A wave machine was added to the model to initiate the water movement in the simulation. (See Figure 2) To accurately generate the wave, the wave piston movement was based on Goring's 1978 numerical wave model (Ref 10). (Also See Appendix A.II)

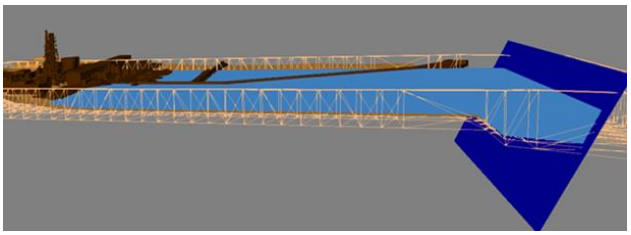

Figure 2. Wave piston setup with initial conditions.

Several simulations with various wave heights were run for the single reactor slice of the facility and for the larger half facility model. (See Figure 3) These simulations not only showed the water behavior and inundation depending on the varying waves, but were used to gather water levels and flow rates on doors and openings leading inside facility buildings. The two facility simulations used millions of particles depending on the resolution of the water particles. Over 12 million particles where used for a ½ meter resolution single reactor simulations and 80 million particles for a ½ meter resolution half facility model.

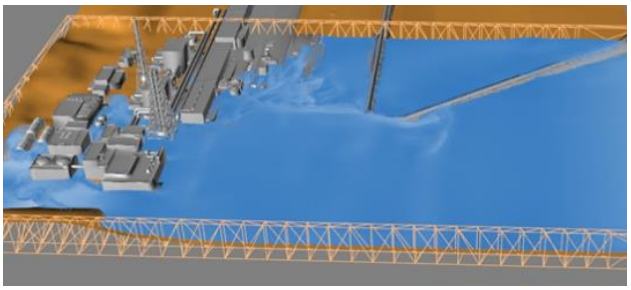

Figure 3. Resulting flooding from tsunami wave simulation

For 12 million particles, each frame (simulated at 24fps) required about 2-3 minutes on a 24 thread 12 core dual processor Xeon 2.8 GHz machine. These large scale simulations provide great result data, but can't be used for large run scenarios. Instead, a few simulations with varying parameters can be done and then extrapolate data points for areas between those simulations. For our tsunami simulations, three wave heights were generated and the flooding level into one building was measured. (See Figure 4)

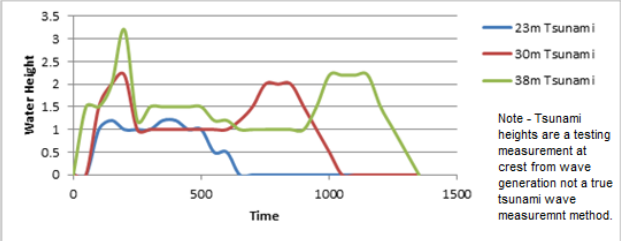

Figure 4. Measured water heights from different tsunami waves

IV.C. Internal Building Flood Simulation

By using smaller yet more detailed models we can detect individual component failures. This also allows for the running of many simulations with minor parameter variations. For testing we constructed a room with several components and a flooding water source. (See Figure 5) Through parameters the water flow rate can be adjusted throughout the simulation.

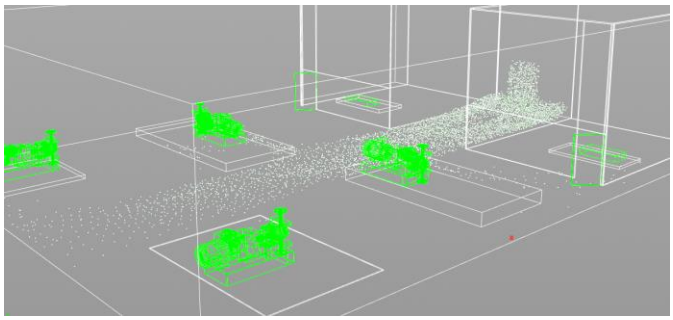

Figure 5. Dynamic interior room flooding example

This allows for direct incorporation into multiple run methods, with runtime communication between the parent application and the simulation.

V. STATE BASED PRA MODELING

Traditional PRA software uses static Fault Trees and Event Tree sequences to calculate probability. While simulations can and do provide additional information for this approach, you are limited by not having a true interaction between the static PRA model and the dynamic Simulation. To accomplish this, we need a dynamic PRA method. We developed a state diagram method using discrete event analysis that can also start up 3D simulations with desired parameters. The PRA simulation sends and receives events from the 3D simulation affecting the current states of the PRA model thus affecting the overall results during runtime. The PRA model can also change the behavior in the 3D simulation. This allows the PRA model to dynamically effect the simulation based probabilistic failures of components or systems.

V.A. State Based Design

The state based model implements a variation of three phase discrete-event simulation where you jump to the next chronological event; execute any required or immediate events; and then check for any conditional events. (Ref 9) Designing a system consists of a layout of states with defined event triggered actions as links or movement between states. A state has two sections, a list (0 or more) of immediate actions and a set of event driven or conditional actions. When entering a new state, all immediate actions are executed in order and the state then stays in the current states list until an event triggers an action that moves it out of that state. (See Figure 6 and Figure 7) We are able to synchronize the continuous 3D simulation by sending the time of the next state event to the 3D simulation and requiring it to return back either that event or any previously occurring 3D event before proceeding. In conjunction the simulation pauses after sending any event it sends and waits for a "Continue" from the state simulation.

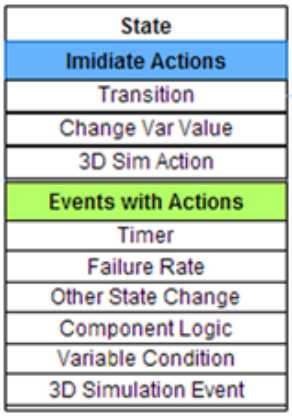

Figure 6. A options for a single state of a state diagram

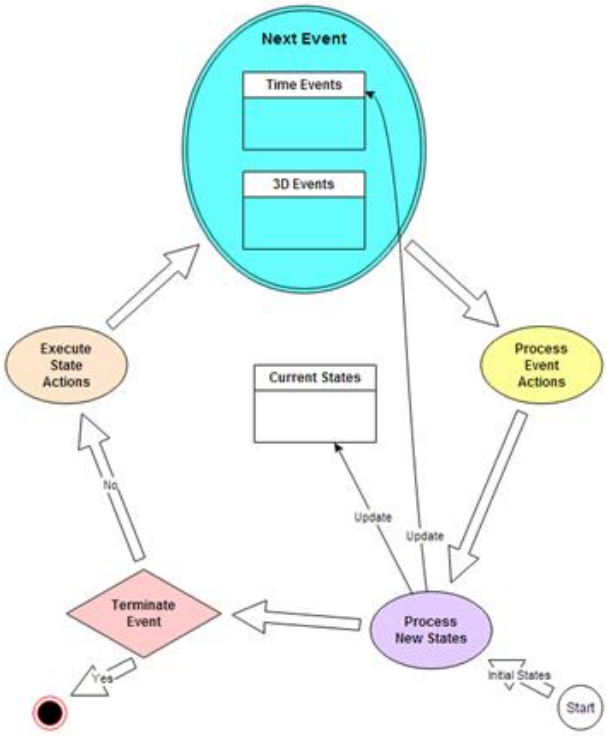

Figure 7: Flow of how State Diagrams are processed

V.B Definitions

State: a logical representation for the condition of a component or system. (4 types)

1. **Start** – A state that is to be placed in the current state list when the model begins a simulation.
2. **Standard** – A normal state representing no special conditions.
3. **Key State** – Marks a state that is to be tracked for final probability calculations. (All "End States" from a traditional PRA model should have a corresponding "Key State")
4. **Terminal** – Marks when a simulation ends. (If this state is encountered then the simulation ends)

Component Group: a group of states that together define the valid states of a component. Only one of these states can be in the "Current States" list at any given time. Each of these states must have a success or failed flag indicating if the component is in an "OK" or "Failed" condition.

Variables: named values that can be set by "Actions" or evaluated by "Events". (3 Types)

1. **3D Simulation** – value for the associated component in the 3D simulation.
2. **Component** – available for all to read but only "Actions" in a "State" associated with that component can change the value.
3. **Global** – available for all to read the value and "Actions" to set it.

Action: (3 types).

1. **Transition** – Start or move to a new state or states. It is probabilistic if it contains more than one to state.
2. **Change Value** – Change the value of a variable.
3. **3D Sim Action** – Send a message to the 3D simulator.

Event: A condition based item that when met executes its assigned actions. (6 Types)

1. **Timer** – executes when time has passed.
2. **Fail Rate** – executes when the sampled time (based on the failure rate) has passed.
3. **State Change** – executes when the associated state is in the list of current states.
4. **Component Logic** – executes if the defined logic for a set of components is met. (Similar to evaluating a FT in PRA without probabilities)
5. **Variable Condition** – executes if a variable meets the user defined condition.
6. **3D Simulation** – executes if the associated 3D component fails.

V.C. Modeling Design

A project model consists of many diagrams and those diagrams can be grouped into logical areas such as Components, Systems, and Plant Response. Traditional PRA models could be translated or imported into a state model with a similar design layout. For example, most basic events for a component can be modeled as a single component diagram with three states. These three states are "Standby", "On" or Running, and "Failed" and a component must always be in one and only one of these states. The starting state is "Standby" and when the system starts, it shifts to "On" or "Failed" depending on if the component failed to start. Any events that cause the item to stop running also move the state to "Failed". (See Figure 8) A component diagram also has a Boolean evaluation value with each state so that the component can be evaluated at any time by other events. It evaluates to the Boolean value of the current state. (See Figure 9)

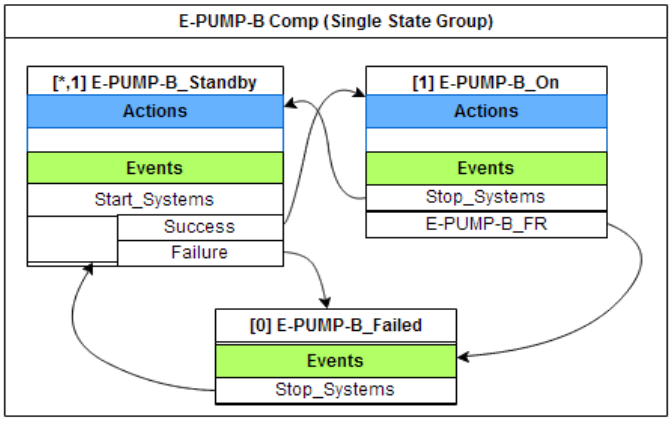

Figure 8: Example of State flow for a component that has both "fails to start" and "fails to run".

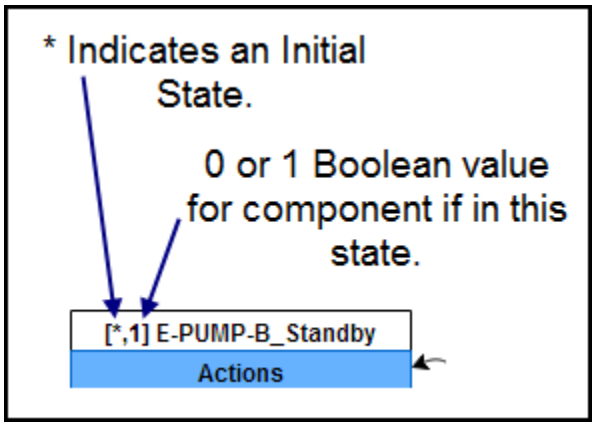

Figure 9. Key for state startup and evaluation

Once all components of are modeled, a Fault Tree from traditional modeling can be directly converted into System diagram with two states and a special event used to evaluate Boolean logic. The two states are "Active" and "Failed", with a "Component Logic" event in the active state evaluating the assigned logic whenever a component diagram state changes. If the logic ever evaluates to false, then the current state shifts from "Active" to "Failed". (See Figure 10) This is similar to a typical PRA model except the logic does not contain any probabilities, just references to the Component diagrams.

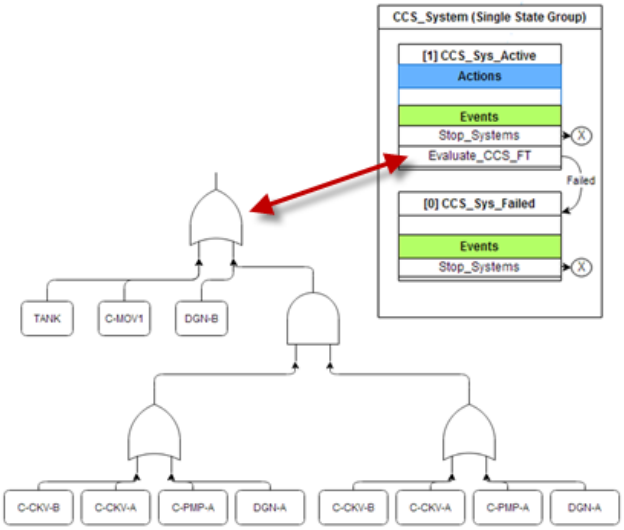

Figure 10: Example of Logic Evaluation for triggering an event.

Finally we have Plant Response diagrams. These diagrams are the main items to be evaluated, similar to Event Trees. This diagram has starting states such as Normal Operation and a Mission Time state. Other states can do general evaluation and movement or be Key states. The Normal Operation starting state contains a Failure Rate event with information formerly held by an initiating event in traditional modeling. The Mission Time starting state uses a timer to end the evaluation if we are past the desired mission time. Key states represent states that we are concerned about when the evaluation ends, such as End States from traditional modeling.

For example if an evaluation is needed for "loss of offsite power" (LOSP) that failure rate is added to the normal running start state for the Plant Response diagram. If that event occurs, the state shifts to the LOSP state. The LOSP state immediately starts new states to evaluate the related systems and waits for any events that trigger new key state. When the evaluation ends, all Key states that are in the current state list are logged with the time of when that state was entered. (See Figure 11)

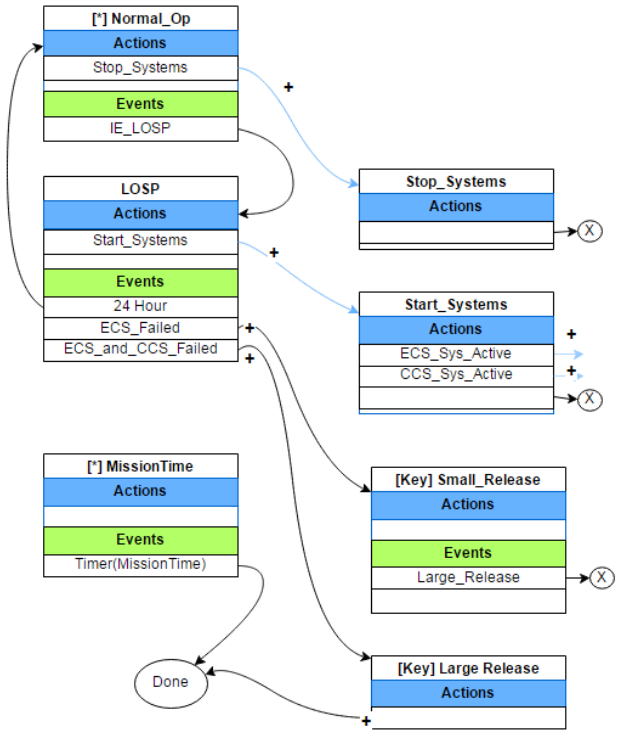

Figure 11. Plant Response diagram for LOSP example

V.D. 3D Simulation Integration

Integrating 3D simulation into the state diagram model is just a matter of starting the simulation when needed and receiving events that can trigger a state change. For each component that can have a failure from the 3D simulation, we must add a 3D simulation failure event in the component state diagram. (See Figure 12)

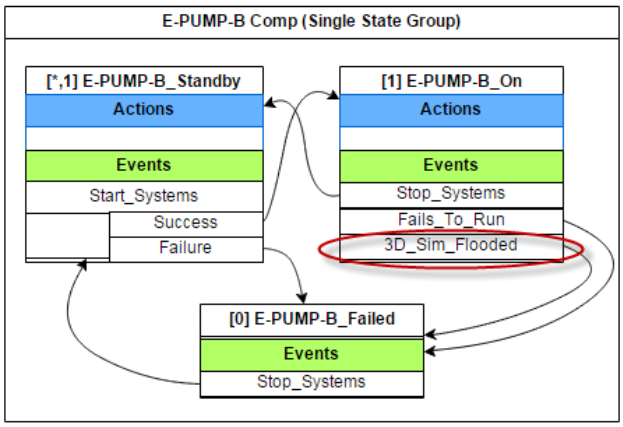

Figure 12. Addition of 3D Simulation event into a component diagram

Then in the Plant Response diagram we need a "3D Sim Action" to start the simulation. As an example we will add a Tsunami initiating event probability to the diagram show in Figure 11. This event triggers a move to the Tsunami state, which possibly triggers LOSP, starts the 3D simulation, and starts evaluation of the systems. (See Figure 13) Evaluating the state diagrams now consists of evaluating the next state diagram event or waiting for the next 3D simulation event.

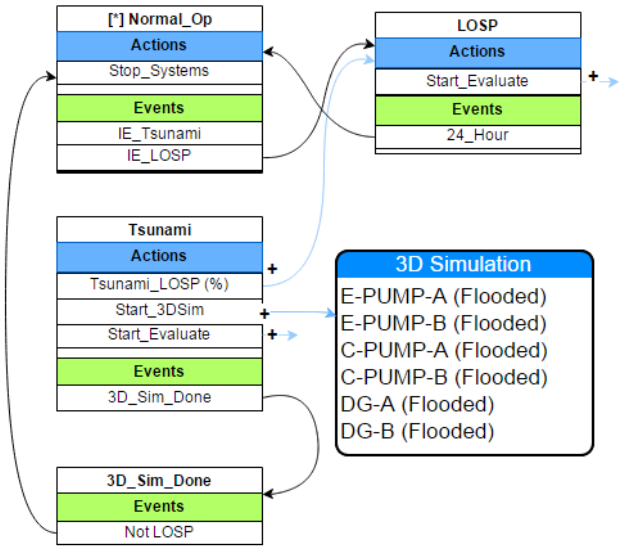

Figure 13. Plant Response diagram incorporating a 3D Simulation.

V.E. State Model Evaluation

Results from this design are gathered from repeat evaluations of the model. From each repeated evaluation, the any Key states in the current state list, along with the time those states were entered, is logged as a result. The compilation of these results allows us to determine the probability of any outcome we are concerned with. When using equivalent models with no time related items, the results from the state model evaluation converge on the same result as other PRA methods. If there are time based relationships, such as the recovery of offsite power that is dependent on the when two failures occur, the state diagram run automatically takes this into account where traditional methods attempt a complex convolution adjustment to fix the deviation.

The state model evaluation also allows the user to determine the average or mean time of a particular outcome, enhancing the result set by giving an additional perspective.

VI. RESULTS

Information from the SAPHIRE "Demo" project was used as a testing model. The only change made was the initiating event frequency for LOSP, where it was reduced

from 2.3/yr to 0.1/yr to better reflect industry occurrences[1]. The state based simulation model was then evaluated 1,000,000 times to verify an accurate baseline with the SAPHIRE results. The results shown in Table 1 verify the state simulation model is statistically equivalent to the SAPHIRE model.

TABLE I. SAPHIRE and State Simulation Comparison

End State	SAPHIRE Results	State-based Simulation Results
Small Release	2.53E-3/yr	2.57E-3/yr
Large Release	7.85E-5/yr	7.86E-5/yr

Additional initiating events for different magnitude tsunami frequencies were then added to this baseline model. One initiating event was added for each of the 1, 10, 100, and 1000 year groupings with frequencies corresponding to statistical data for tsunamis. Each time a simulation is run, any tsunami events that occur are then sampled for wave height data. (See Figure 14) Historic data of Tsunami wave heights were used to determine a sampling equation. The methods used are worthy of a separate paper with debate of methods and thus not covered in this paper. It is also assumed for this analysis that any tsunami height above 3 meters also triggers a LOSP event in the simulation.

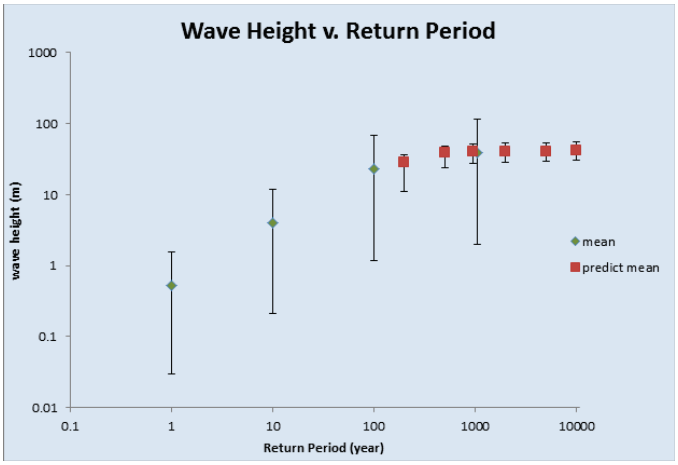

Figure 14. Results of the Bayesian Extreme Value Analysis Compared to a Simpler Poisson Model (bars represent 90% intervals while the point represents the mean)

The simulations were run with the added tsunami initiating events but without the 3D simulation and then again with the 3D simulation including the first as baseline for comparison. The results show a small increased failure from LOSP triggered by the tsunami (Col. 1 and Col 2 in Table II) and then a large increase caused by the 3D simulation failures (Col. 2 and Col. 3 in table II). Adding the tsunami initiating events only slightly increased the probabilities because it only increased the likelihood of the LOSP. If the system is normally very reliable without power, increasing the number of times it is without power only moderately increases the overall risk. However if critical components can be disabled by the water from the tsunami, it will dramatically increase the overall risk. (See Table II) Similar results could be achieved without 3D simulation, by building into the model a component failure dependency on the wave height of a tsunami initiating event. However determining the failure of those events due to secondary effects such as flooding can be difficult and more error prone than properly modeled physics based simulation.

TABLE II. Results with Tsunami Simulation Components

End State	State-based Simulation Baseline	With just Tsunami IE	With 3D Failure Events
Small Release	2.59E-3/yr	3.08E-3/yr	1.17E-2/yr
Large Release	7.86E-5/yr	9.33E-5/yr	5.23E-3/yr

VI.A. Additional Capabilities

We have shown one example of a tsunami based flooding scenario, with failed components from water contact. Incorporating 3D simulation failures can be used for many more external event models and component failure methods such as erosion, debris, or even human response failures due to inaccessible areas. Besides external event analysis, industry could use these methods to test design specifications of new facilities, or changes to existing ones against a set of likely scenarios to optimize those designs. Site locations could also be analyzed to determine the best building configuration or terrain modifications before construction begins. Models constructed for this analysis could also be used by other applications such as a simulation based training tool.

VII. CONCLUSIONS

Computers have been used for 3D modeling and simulation, but only recently have computational resources been able to give realistic results in a reasonable time frame for large complex models. In this report, we described the methods, techniques, and resources which are being developed at the INL to incorporate a 3D simulation engine into risk analysis methods. By combining PRA methods and 3D simulations, we are able

[1] The rate from NUREG-1032 was 0.12/yr while the rate in NUREG/CR-5496 was 0.058/yr – for this demonstration we used 0.1/yr.

to enhance result data and reduce the need for modelers to correctly predict the outcomes of hard to determine scenarios such as external events.

ACKNOWLEDGMENTS

We appreciate DOE for funding this research and furthering the advancement of PRA capabilities.

DISCLAIMER

.

REFERENCES

1. Wikipedia contributors, "Computational physics" Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Computational_physics. (accessed August 23, 2014)
2. Wikipedia contributors, "Smoothed-particle hydrodynamics" Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Smoothed-particle_hydrodynamics. (accessed August 23, 2014)
3. N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler and M. Teschner, "Versatile Rigid-Fluid Coupling for Incompressible SPH," *ACM Transactions on Graphics (Proc. SIGGRAPH 2012),* vol. 31, no. 4, 2012.
4. Google Maps Elevation API, Google, http://www.programmableweb.com/api/google-maps-elevation (accessed Nov 1, 2014)
5. Safinaz El-Solh, "SPH Modeling of Solitary Waves and Resulting Hydrodynamic Forces on Vertical and Sloping Walls". Master of Applied Science in Civil Engineering Report, University of Ottawa, Ottawa, Canada.
6. Houdini Software, Side Effects, http://www.sidefx.com/ (accessed Nov 10, 2014)
7. Neutrino Software Research, http://www.naadir.tk/research (accessed Nov 10, 2014)
8. Artist iljujjkin, Fukushima 1 Nuclear Power Plant, TurboSquid, http://www.turbosquid.com/3d-models/c4d-nuclear-power-plant-fukushima/594020 (accessed Nov 10, 2014).
9. Michael Pidd, "Computer simulation in management science" Wiley (1998)
10. Goring, D. G. Tsunamis – The Propagation of Long Waves Onto a Shelf. Doctoral Dissertation, Report No. KH-R-38, Keck Laboratory of Hydraulics and Water Resources, California Institute of Technology, Pasadena, California (1978).

A. APPENDIX

A.I. Altitude Point Cloud to Polygon Terrain Map

To obtain a terrain map for use in simulations, a custom application was developed. This application uses the Google Altitude API to obtain a point could for a given area. In order to correctly determine coordinate locations for querying points in a grid formation, a Haversine formula was used.

Haversine formula:
$a = \sin^2(\Delta\varphi/2) + \cos \varphi 1 \cdot \cos \phi 2 \cdot \sin^2(\Delta\lambda/2)$
$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$
$d = R \cdot c$
Where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km).
Note that angles need to be in radians.
[http://www.movable-type.co.uk/scripts/latlong.html]

The OBJ file format is generated first by defining points in a parameter space of curve or surface.
As an example, if "v 5 15 34.483" were to be generated, it would mean that a vertex (v for vertex) at rectangular point coordinate (5, 15) with z-axis (elevation) 34.483.

A.II. Goring Solitary Wave Generation – Numerical Model

Goring (Ref 10) proposed a model for the purpose of laboratory solitary wave generation. The surface profile (x,t) of a solitary wave can be described using the following equation:

$$\eta(x,t) = H\,sech^2\left[\kappa\left(Ct - X_0\right)\right] \qquad \text{(A-1)}$$

$$C = \sqrt{g(H+h)} \qquad \text{(A-2)}$$

$$K = \sqrt{\frac{3H}{4h^3}} \qquad \text{(A-3)}$$

Where C is the wave celerity or phase velocity, X_0 is the wave displacement, H is the wave height and h is the depth of the ocean. Applying equation A-1 to the wave maker piston results in

$$X_0(t) = \frac{H}{\kappa h}\left(tanh\left(\kappa\left(Ct - X_0\right)\right)\right) \qquad \text{(A-4)}$$

Using this equation one can solve for the wave piston displacement and wave piston duration using newton iterations resulting in

$$S = \sqrt{\frac{16Hh}{3}} \qquad \text{and} \qquad t_f = \frac{2\left(3.80+\frac{H}{h}\right)}{\kappa C}$$

Where S is the displacement and t_f is the time taken for the displacement.