INL/EXT-14-33211

September 2014

Advanced Small Modular Reactors

Prototype Development Capabilities of 3D Spatial Interactions and Failures During Scenario Simulation

Steven Prescott
Ramprasad Sampath
Curtis Smith
Tony Koonce

Idaho National Laboratory

INL/EXT-14-33211

Advanced Small Modular Reactors

Steven Prescott
Ramprasad Sampath (Centroid PIC)
Curtis Smith
Tony Koonce

September 2014

Idaho National Laboratory
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Overview

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. This report addressed the methods, techniques, and resources used to develop a prototype for using 3D modeling and simulation engine to improve risk analysis and evaluate reactor structures and components for a given scenario. The simulations done for this evaluation were focused on external events, specifically tsunami floods, for a hypothetical nuclear power facility on a coastline. The approach that is describe in this report is applicable to the evaluation of different reactor sites and types, including current reactors and proposed small modular reactors. The examples that are presented are generic and are not representative of any existing or proposed reactor facility. As advanced small module reactor models are created for analysis, we will incorporate these into the analysis framework described herein.

CONTENTS

FIGURES

ACRONYMS

API	Application programming interface
FLIP	Fluid-Implicit-Particle
FPS	Frames per second
GB	Gigabyte
GHz	Gigahertz
GPU	Graphics Processing Unit
HPC	High Performance Computing
INL	Idaho National Laboratory
INI	Initialization
PIC	Particle-In-Cell
SPH	Smoothed Particle Hydrodynamics
SSC	System Structures and Components
NOOA	Oceanic and Atmospheric Administration
CDF	Cumulative Distribution Function
GEV	Generalize Extreme Value

Prototype Development Capabilities of 3D Spatial Interactions and Failures During Scenario Simulation

1. BACKGROUND

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. [5]

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 targeted some of these standard methods and tools used in other fields to analyze flooding events for risk analysis of nuclear facilities such as small modular reactors. These tools have then been incorporated into PRA analysis software to improve results for external event analysis.

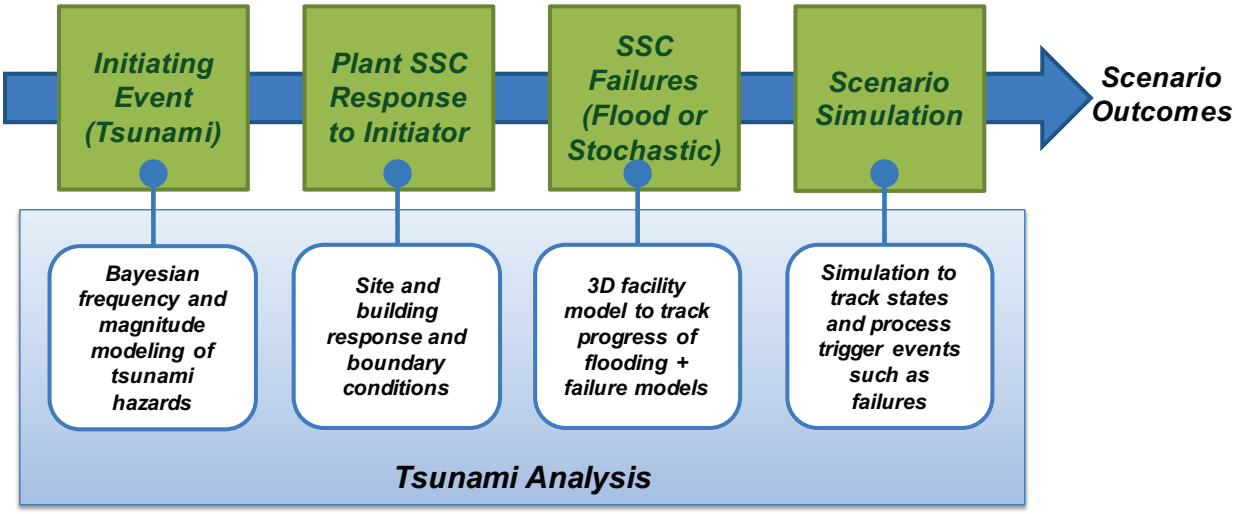


Figure 1: Steps of the Analysis to Model Tsunami Initiating Events

2. Bayesian Modeling for Extreme Event Initiators

The first step in the analysis is to perform Bayesian analysis of initiating events, including the frequency of tsunami events. The goal of the Bayesian modeling for extreme event initiators (e.g., tsunamis) is to gain insight as to the expected tsunami height for various return periods given historical data, and to make predictions as to the expected tsunami heights for longer return periods at a given plant location. The desired result is a distribution of expected tsunami heights for various return periods (i.e., yearly, every 10 years, every 100 years, etc.) based upon observed historical events.

Historical tsunami data for this study was obtained from the National Oceanic and Atmospheric Administration (NOAA). The data is available for download from the NOAA website,

http://www.ngdc.noaa.gov/hazard/tsu_db.shtml. The download file contains details for all known tsunamis world-wide from 2000 B.C. to present, including; maximum wave height, magnitude and location of seismic initiating event, and other attributes. The validity and accuracy to be expected from historical events are also graded to allow for scrutiny and segregation of the data, if desired.

The case analyzed and discussed in this section includes all known historical tsunami events affecting the Fukushima Prefecture region of Japan. For the case study described in this report, we assume that our hypothetical nuclear power plant will be located on the north-east coast of Japan, in the Fukushima Prefecture region. In order to obtain these data from the large data file from NOAA, all tsunamis affecting Japan were retained from the data file. The data were then segregated by the location of the seismic event that generated the wave. Any data that initiated within appropriate latitude and longitude that the Fukushima Prefecture could be impacted was retained for analysis and the other data were discarded. The latitude and longitude used for parsing the data was 35 – 40 degrees north and 141 – 144.5 degrees east, respectively. This resulted in data ranging from the years 1611 to 2013 (a span of 403 years). A sample of the data retained is illustrated in Table 1.

Table 1. NOAA Tsunami Data (example).

YEAR	MONTH	DAY	EVENT VALIDITY	FOCAL DEPTH	PRIMARY MAGNITUDE	LATITUDE	LONGITUDE
1938	11	13	4	60	6	36.7	142.1
1938	11	6	4	20	7.1	37	142.2
1938	11	30	4	50	7	37.5	142.2
1938	11	5	4	30	7.7	38.2	142.2
1938	11	5	4	30	7.6	37.552	142.214
1938	11	6	4	0	7.5	36	142.3
1959	1	22	4	40	6.8	38.297	142.373
1959	10	26	4	10	6.7	38.1	142.4
1960	3	23	4	20	6.7	39.5	142.5
1960	3	20	4	20	7.5	37.812	142.619
1961	1	16	4	20	6.4	38.64	142.75

Bayesian Inference was performed using these data to determine the expected maximum wave height for the one year, 10 year, 100 year, and 1000 year return periods. Several distribution types were used in this Bayesian Inference, including: gamma, exponential, and lognormal. The gamma and lognormal distributions did not produce sensible distributions for the 1000 year return period and were then rejected from further analysis. The exponential distribution yielded viable results for all four return periods, with increasing uncertainty as the return period grew, as expected. The distributions used for each wave height for the four return periods are given in Table 2, along with their associated 5^{th}, 50^{th}, 95^{th} percentiles and mean values. The cumulative distribution function (CDF) for the four return periods are given in Figure 2, and wave height for each return period are illustrated in Figure 3.

Table 2. Wave Height Distributions for Each Return Period.

1 year return period: exponential distribution with $\lambda_{mean} = 1.90 \pm 0.095$			
$5^{th} = 0.03$ m	$50^{th} = 0.36$ m	Mean = 0.53 m	$95^{th} = 1.57$ m
10 year return period: exponential distribution with $\lambda_{mean} = 0.249 \pm 0.039$			
$5^{th} = 0.21$ m	$50^{th} = 2.79$ m	Mean = 4.02 m	$95^{th} = 12.06$ m
100 year return period: exponential distribution with $\lambda_{mean} = 0.044 \pm 0.020$			
$5^{th} = 1.18$ m	$50^{th} = 15.9$ m	Mean = 22.9 m	$95^{th} = 68.7$ m
1000 year return period: exponential distribution with $\lambda_{mean} = 0.026 \pm 0.025$			
$5^{th} = 1.99$ m	$50^{th} = 26.9$ m	Mean = 38.8 m	$95^{th} = 116$ m

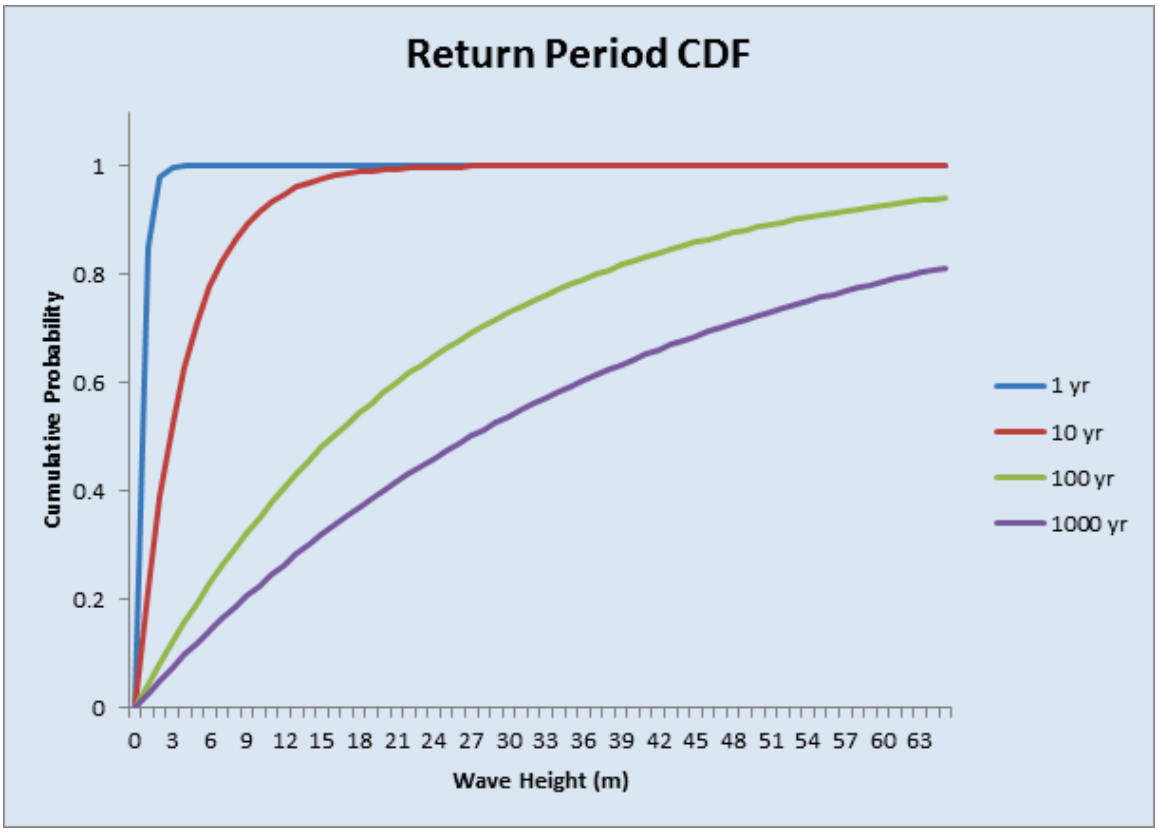


Figure 2: Return Period Cumulative Distribution Functions.

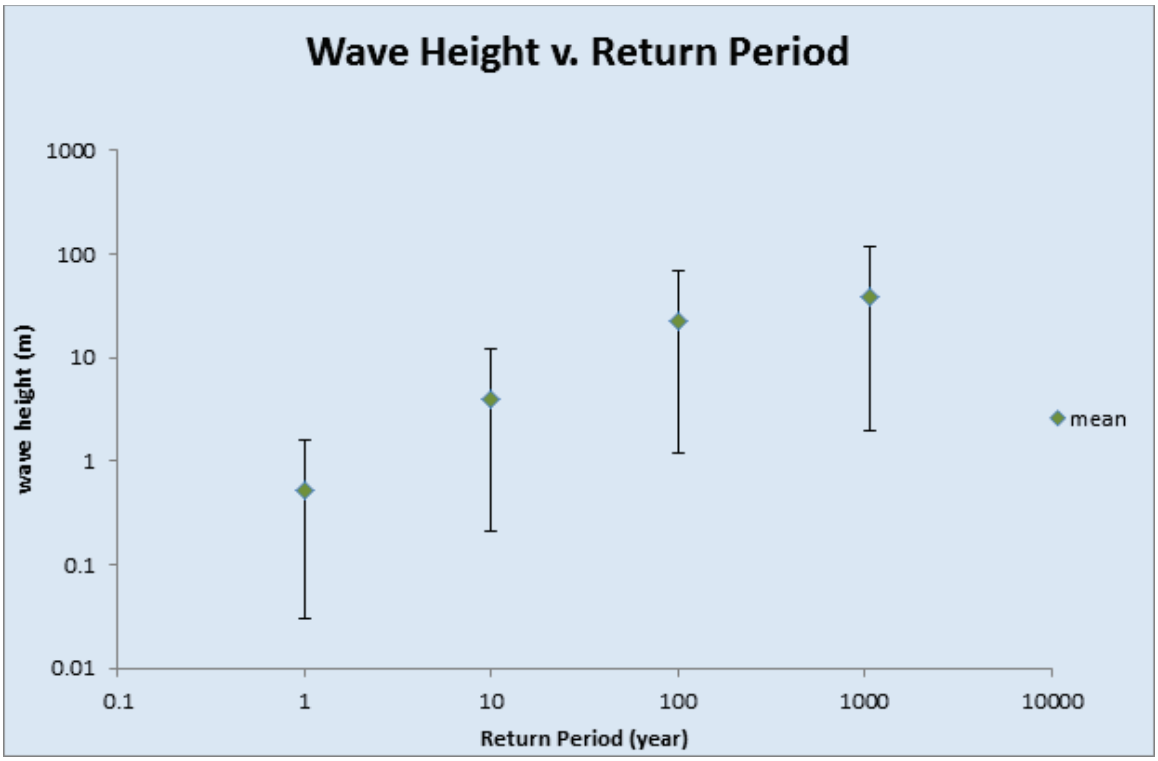


Figure 3: Wave Height for Various Return Periods (based on historical data).

The historical data was sufficient in developing wave heights for observed return periods, but unable to predict maximum wave heights to be expected for longer return periods. An "extreme value paradigm" is used in order to extrapolate observed data for estimation of the predicted return period wave heights for longer periods (i.e., 10,000 years and 100,000 years). Specifically, the generalized extreme value (GEV) family of distributions is used to perform an asymptotic extrapolation of the known data. For this hypothetical example, we created an OpenBUGS script to perform the Bayesian inference, building upon the Generalize Extreme Value (GEV) modeling as described in (Kelly & Smith, 2011). Additional details on the GEV model and assumptions may be found in Chapter 13 of [2]. The OpenBUGS script that was produced is shown below.

```
Modeling extreme value of tsunami height using GEV
model {
for(i in 1:N) {
tsu.h[i] ~ dgev(mu, sigma, xi)
}
sigma ~ dgamma(0.0001, 0.0001)
mu ~ dflat()
xi ~ dflat()
}

data
list(tsu.h = c(25, 4.5, 38.2, 29, 38.9), N = 5)
```

The GEV distribution is given below and uses three parameters; mu, sigma, and xi. Mu can be thought of as a location parameter, sigma is a scale parameter, and xi is the shape parameter of the distribution.

$$GEV(z) = \exp\left\{-\left[1 + \xi\frac{(z-\mu)}{\sigma}\right]^{-\frac{1}{\xi}}\right\}$$

When xi > 0 the asymptotic extrapolation approaches a maximum return period, but there is no upper bound on the predicted maximum wave height; xi < 0 the asymptotic extrapolation produces a maximum predicted wave height with no maximum return period.

The first attempt to employ GEV Bayesian Inference was with the annual maximum tsunami height for years 1611 to 2013. Not every year had a recorded tsunami and some years had two or more small tsunamis. Since seismic events along faults happen every day, given they are small in magnitude, each year that did not have a recorded tsunami was recorded as 0.1 m (10 cm), which is well below high tide and any measurable wave height. The maximum tsunami height within a year was taken for all years that had more than one recorded tsunami. This produced 403 data points for the inference analysis. The posterior parameters were; $\mu = 0.2426$, $\sigma = 0.5033$, and $\xi = 3.497$. This resulted in predictive wave heights of 3.76 m, 1.4E+6 m, and 4.5E+9 m for the 10 year, 100 year, and 1000 year return periods, respectively. These results do not appear to be representative of historical observations and are a result of the relatively large xi value.

The second attempt to perform the GEV Bayesian Inference used the maximum wave height for ten year periods (instead of every year). There were some decades in which no recorded tsunami occurred and others in which several tsunamis were recorded. The same approach as above was used to adjust the data as for the annual return period. The posterior parameters using the 10 year return period data were; $\mu = 1.239$, $\sigma = 5.293$, and $\xi = 4.649$. Once again, xi was very positive resulting in no upper bound for the predicted maximum tsunami wave height. The main reason the 10 year and one year return period data results in a xi greater than zero is because of numerous years in which no tsunami is recorded or only a small tsunami occurs. This causes an artificial growth in predicted wave height when years with large tsunamis are added to the data base.

The third attempt to perform the GEV Bayesian Inference method was as before except using data for the maximum wave recorded for each 100 year period. This resulted in no zero values used in the analysis since all the data ranged between 4.5 – 38.9 m. The posterior parameters here were; $\mu = 23.3$, $\sigma = 29.58$, and $\xi = -1.645$. The 100 year return period data resulted in a negative value for xi which results in an asymptotic extrapolation from observed data to larger return periods. These parameters predict a mean value of the maximum possible tsunami height for any return period beyond 100 years to be less than or equal to 41 m. Figure 4 gives the predictive wave heights for larger return periods along with the previous results from above.

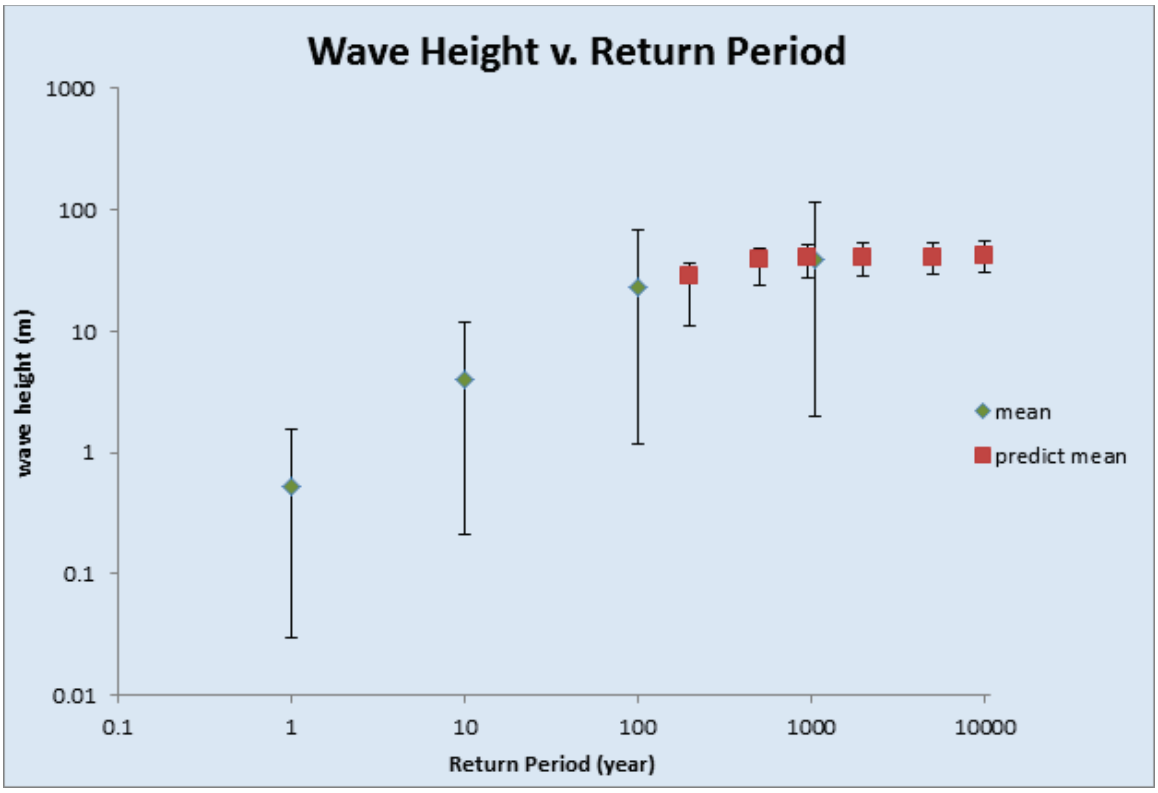


Figure 4: Wave Height for Various Return Periods.

3. SSC Failure Modeling

Traditional PRA software uses static Fault Trees and Event Tree sequences to calculate probability. While simulations could provide additional information for this approach, it cannot be directly incorporated into the solving methods. To do this, we need a dynamic method to do the PRA.

3.1 Time Based State Model

The solution to the dynamics issue is to use a state model of the system -- this allows the model to have time based dependency and behavior. This modeling is done using an event and action driven state diagram approach. At any given moment, the model is in a set of "States." Each state can have "Actions" it performs upon entering that state and "Events" that trigger an action or set of actions. The set of "Current States" changes over time until a terminal state is reached. Once a terminal state is reached, the "Current States" list is evaluated and logged as one iteration of the model (the black dot in Figure 5). After running many iterations, this model converges on a failure probability and is equivalent to traditional static PRA methods.

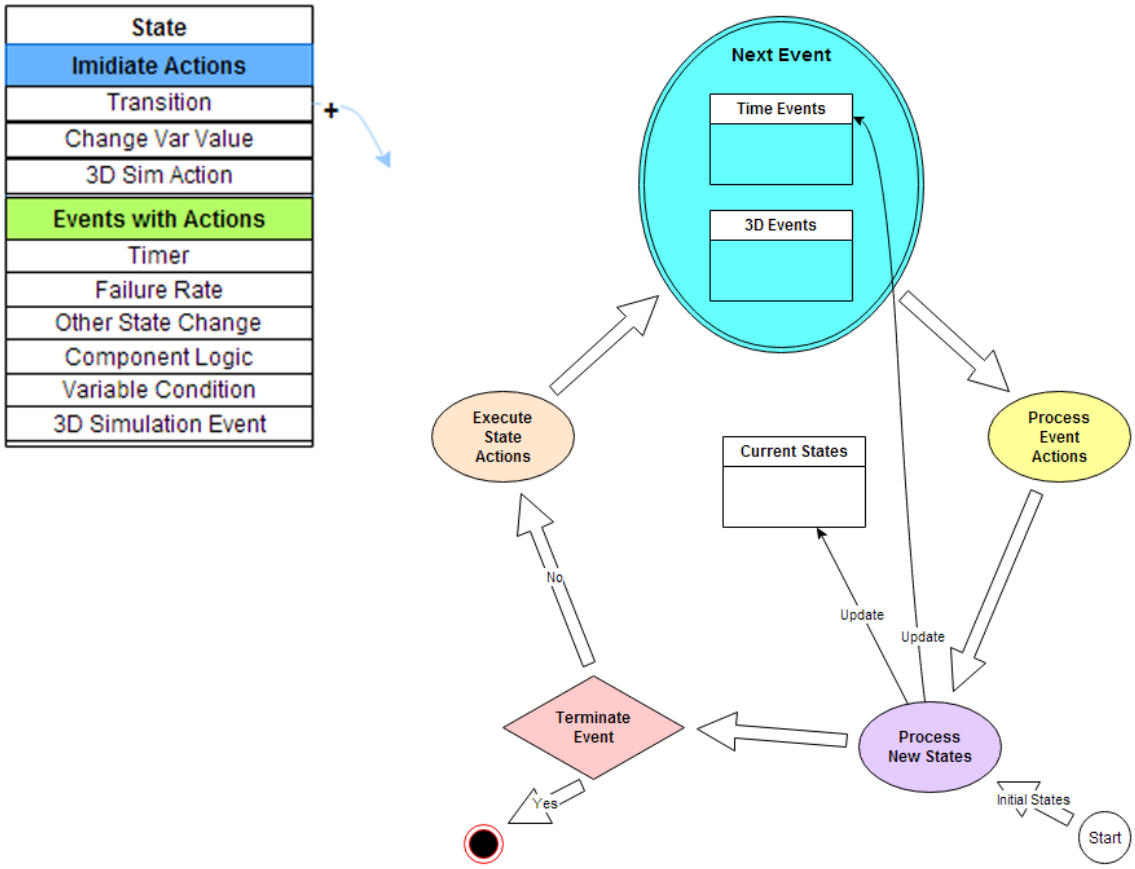


Figure 5: State Action and Event options (Left), flow of State processing (Right).

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.

The state diagrams can be defined based on PRA modeling practices of components with various failure types, fault trees, and event trees. For simplicity, most of the following examples and diagrams will be describing the example described in (Rasmuson, 1992)).

Each component has various states that it can be in such as Standby/Off, Active/On, and Failed. The paths from one state to another are dictated by events. For example, a failure type is handled through an event. A failure to start would use an event placed on the standby state. When a start system request is made, it either moves to the "On" state or the "Failed" state determined by the probability associated with it (see the red circled event in "E-PUMP-B_Standby" of Figure 6). A failure to Operate/Run would be represented with an event in the Active/On state going to the Failed state, the event produces a time based reaction by sampling the probability associated with it (see the red circled event in "TANK_Active" of Figure 6).

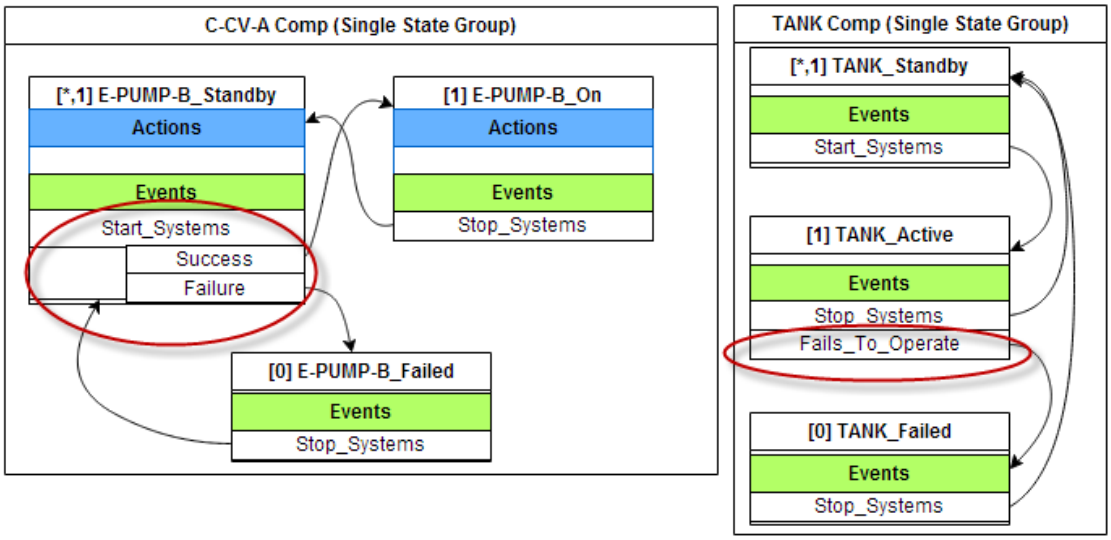


Figure 6 – Example for state transitioning for components and types of failures. (Left) Fails to Start (Right) Fails to operate or run.

If the component has both failure types then it is just a combination of the two events in the corresponding states (as shown by E-PUMP-B in Figure 7).

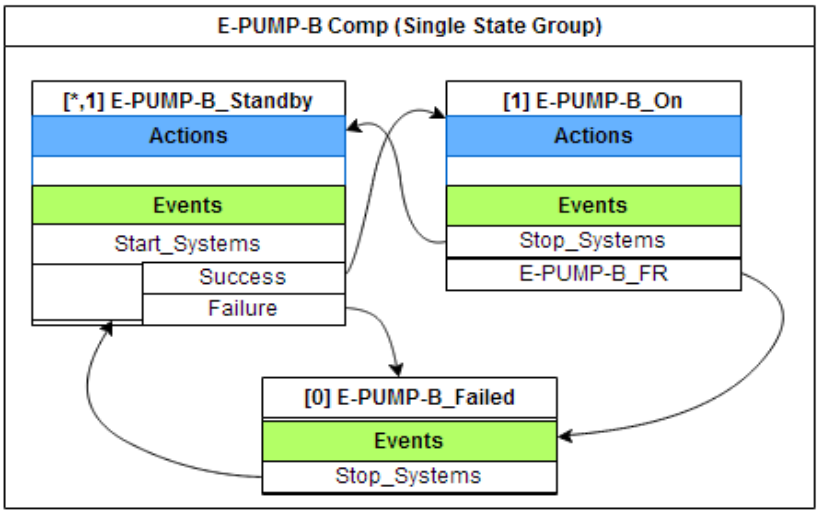


Figure 7 - Example of State flow for a component that has both "fails to start" and "fails to run".

When adding states for a component, each state is flagged as either a success or failed state for the component (designated by the [0] or [1] before the state name, see Figure 8). This flag allows for an evaluation of the component by other events in the model.

Once all component states have been modeled, systems can be represented using component logic events. A system failure event uses a logic diagram of the components and evaluates its success or failure to trigger the event's actions (see Figure 8). This is similar to a typical PRA model except this is based upon simulation at a component level. The states for the fault tree system could be as simple as an active and failed state with movement between them triggered by the previously described event (see Figure 8).

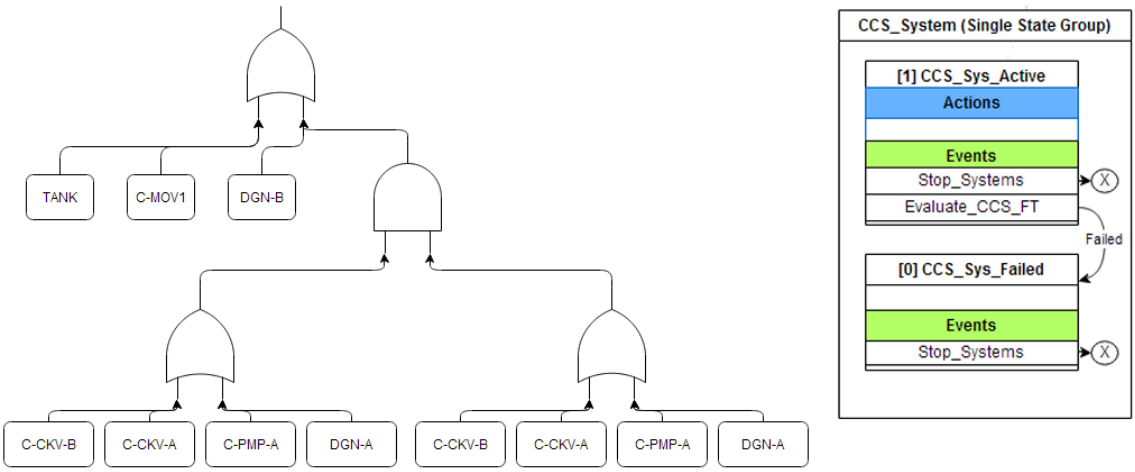


Figure 8 - Example of Logic Evaluation for triggering an event

With all the systems modeled, the next thing is to track the critical or end states of the plant. This is done by starting with a normal operations state with initiating events transitioning to other plant states and

those leading to desired end states. It also needs a terminal state to end the simulation with a mission-time state and other final states transitioning to this terminal state.

In the demonstration example we have the following states (see Figure 9).

Normal_Op – the starting state for the simulation, with the initiating event IE-LOSP transitioning to the LOSP state. It also has an immediate action of going to Stop_Systems in order to reset everything if it goes back into normal operations. ("+" indicates adding a *new* state instead of leaving this one)

LOSP – Indicates that the system is without off-site power. When entering this state, we need to immediately activate all the systems of the plant and monitor events that constitute a small or large release.

Stop_Systems – A temporary state use to turn off the evaluation of any component and systems. (The empty action with the arrow to a circled "X" indicates an immediate exit of the State.)

Start_Systems – A temporary state used to trigger any failures for components and to evaluate systems.

Small_Release – An end state that results are needed for the overall analysis. The only way to leave this state is to have a large release which supersedes the small release.

Large_Release – An end state that results are needed for the overall analysis. This state also triggers an end of the simulation.

MissionTime – A starting state used to shut down the simulation after the mission time has elapsed.

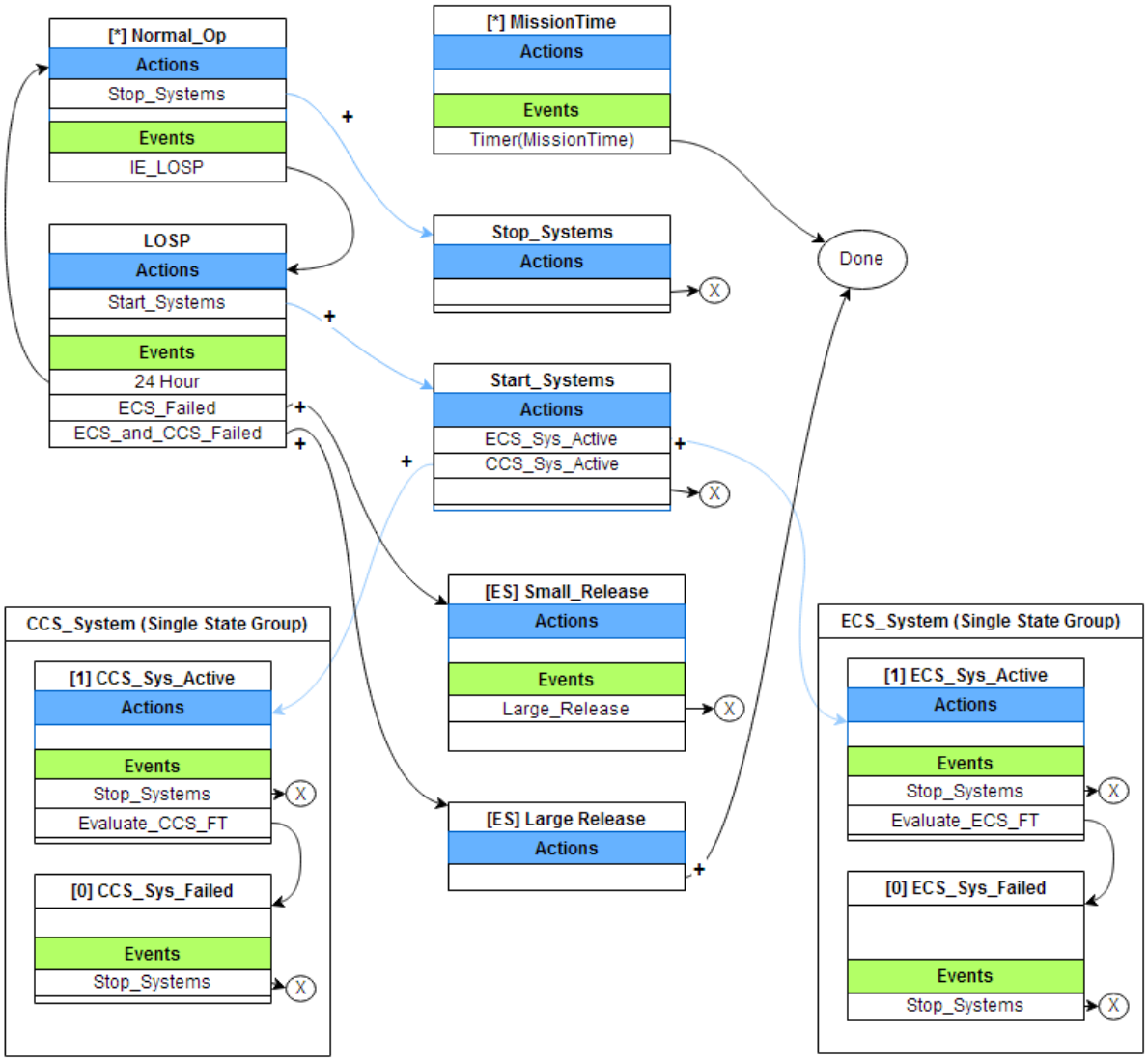


Figure 9 - State diagram example for evaluating Systems and tracking End States of the demonstration example.

3.2 User Interface

A simple user interface to visualize and set up the state diagrams is in development. It organizes the various parts of a simulation into basic components and allows the user to construct state diagrams by dragging, dropping, and linking those components. Starting a simulation on the server and then displaying the results is also easily done through a web interface.

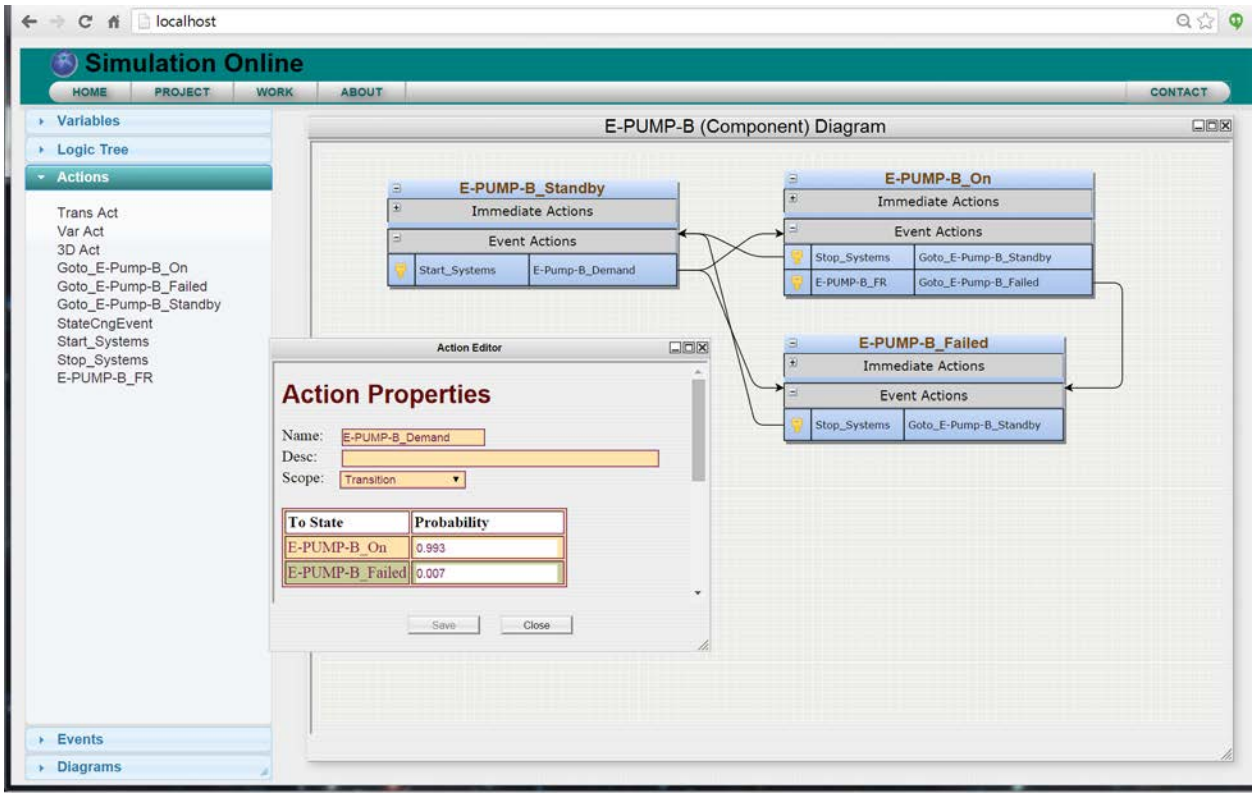


Figure 10: Web based user interface for setting up the State Digram model.

4. 3D Setup

Before 3D simulations can be used in the risk analysis, several pieces of the model need to be created. These pieces consist of a 3D facility models, terrain model, scenarios to run on the model (different initiating events), and component interactions or failures to monitor.

4.1 3D Facility Models

Before any simulations can be done, a model to run the simulation on must be constructed. Models used for simulation consists of two parts, the visual model, and the collision model. The visual model consists of an accurate representation of the physical appearance of the facility and usually has image maps applied to the 3D structures to make them appear realistic. For our purposes, little emphasis is placed on the visual appearances, as long as an analyst can easily understand where and what items are such as buildings and components. More important is the collision model and, in our case, it is almost identical to the visual model except with additional invisible boundary items to bound calculations within the desired risk analysis scope.

There can be many different models and level of detail in those models. For example, a large low-detail facility model might be constructed and used for the external event evaluation. (See Figure 11) Then, for buildings with components of interest, higher detail interior models could be created. (See Figure 12) Simulation data from the first larger model can be used to run and get data from the higher detail models.

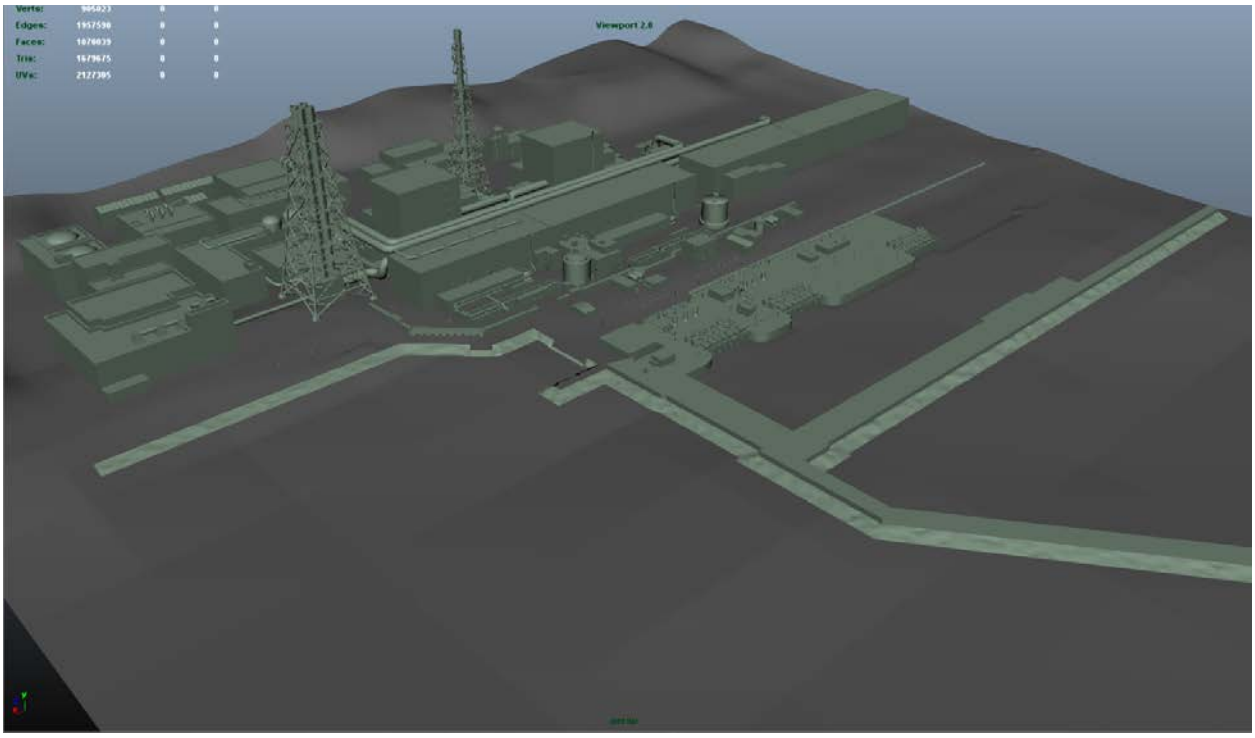


Figure 11: Large scale, low detail facility model.

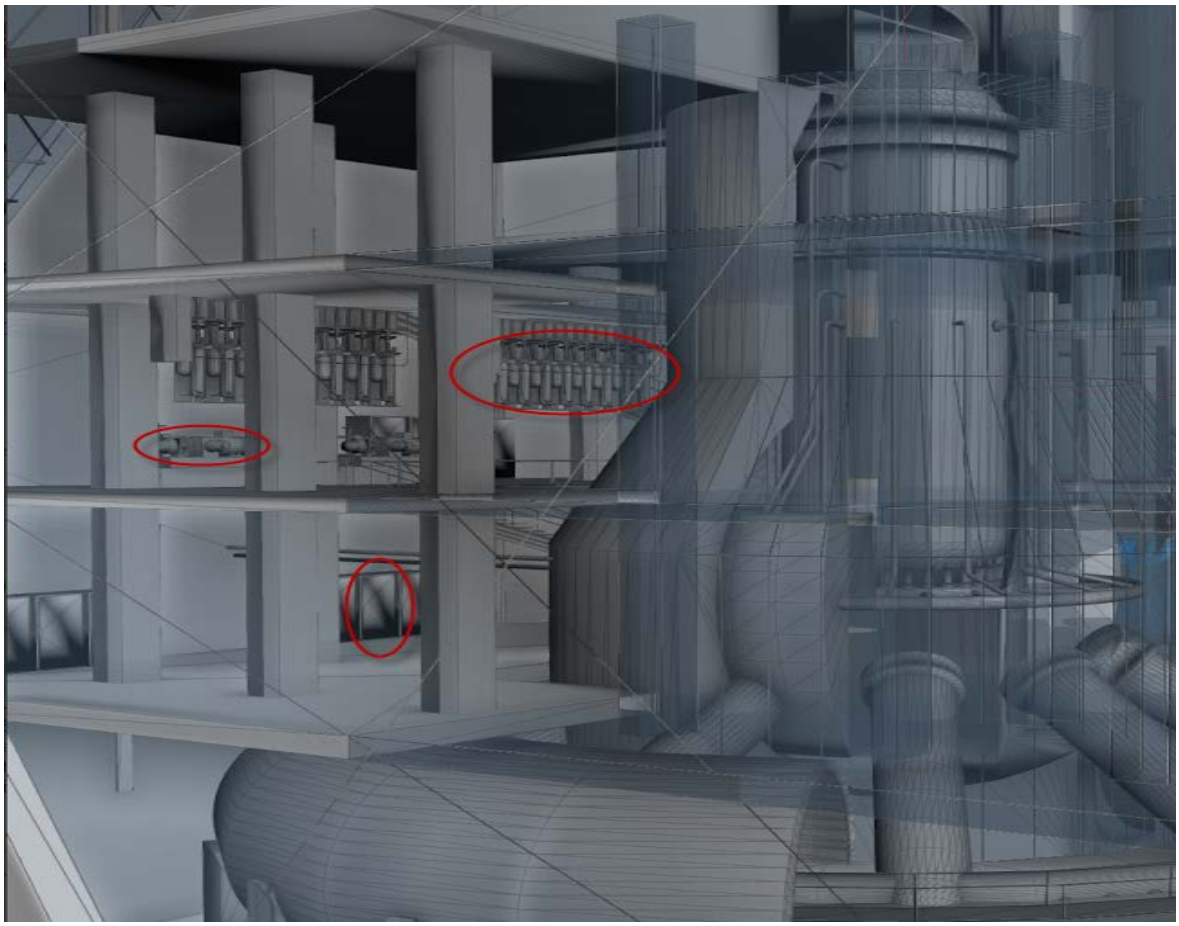


Figure 12: High detail internal building model with critical components modeled.

4.2 Terrain mapping

For our demonstration problem, initial testing was done with a flat gradient on the facility and simple gradual slope for the ocean floor. However, to produce accurate results, a terrain map showing the topography of the area is needed. To obtain this information, a standalone applet was created using Google's Elevation API. (See Figure 13) The Elevation API allowed for us to retrieve elevation levels from a set of points in a rectangular area anywhere on the surface of the earth. The user inputs latitude, longitude, distance and resolution data for the desired location. Using this information, the application gives a visual representation of the defined area on a Google map (and which can also be saved as a 3D model for incorporation into the facility 3D model). For our testing we used a location with a scale of five meters.

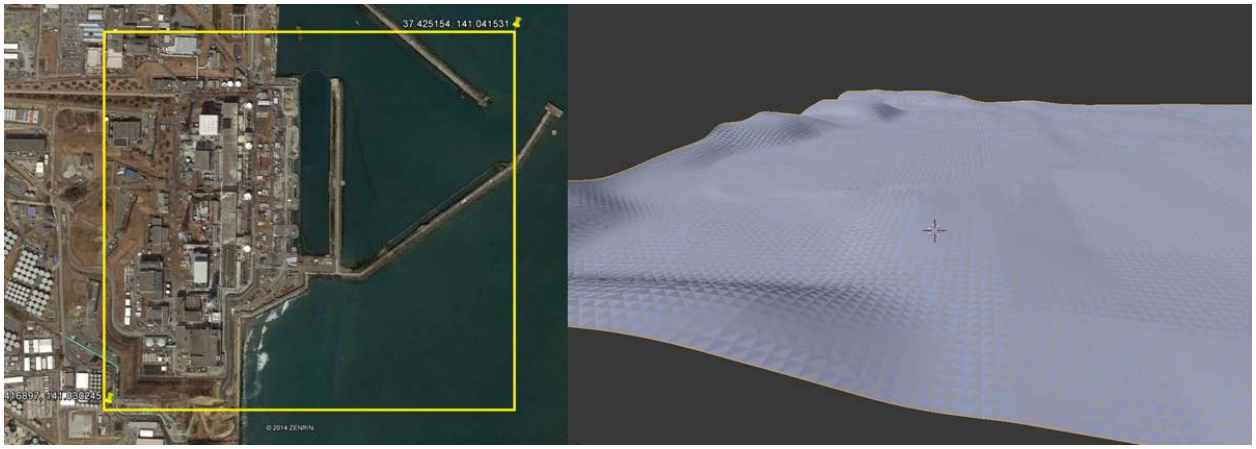


Figure 13: The 3D map generated from the Fukushima site.

Using the data from the inputs, the terrain map application then calculates all the points within the rectangular area using the Haversine formula (See Appendix A) and proceeds to query the Google terrain database for elevation data for each point that was calculated. After all the queries are complete, the application is able to export all the elevation data gathered and convert it into a set of representative points. The exported file is saved as Wavefront Technologies' most common geometry interchange file format: the OBJ file format. This can then be combined with a facility model for a scenario to be simulated. An accurate terrain map was used for second stage simulations described in Section 7.3.

4.3 Scenarios

Once the models are complete, desired scenarios are setup in the 3D environment such as a tsunami wave. Just as there are different levels of detail for the 3D models, different levels of resolution can be used for the scenarios. For example, to do the tsunami wave, a large body of particles would be used for the ocean with a wave generator. Those particles can be fairly large (1/2 – 1 Meter) and still be an adequate representation of water flow around the facility. However, for simulation inside a building, the particle size would need to be smaller in order to be a valid representation. Once a scenario is set up, parameters can be adjusted, such as water flow rates or wave height/length/direction, in order to get variations of the same scenario.

4.4 Component Failures

Within the 3D model, significant components are identified and mapped with failure properties. For example, an electrical panel could be susceptible to shaking (from earthquakes or water-carried debris), water contact, and impacts. These components are also linked with the external state model items, so that any component failure information can be transferred and affect the states. (See Figure 12)

5. Simulation Engines

A simulation engine is what gives makes items in the simulation behave as they would in real life. It uses mechanistic models coupled with probabilistic aspects to predict the behavior and interactions of items in the environment. There are different 3D physics simulation engines using a variety of methods.

Each has its advantages or disadvantages depending on the intended use. In this report, we describe a couple of approaches, and provide demonstration cases where these may be applicable for risk analysis.

5.1 Houdini FX

For this research, we initially started with a software package called Houdini FX. 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. (See Figure 14) This feature makes it useful for incorporation into risk analysis modeling since the scenario evolution can be controlled (e.g., a failure can be triggered) during the calculations being performed.

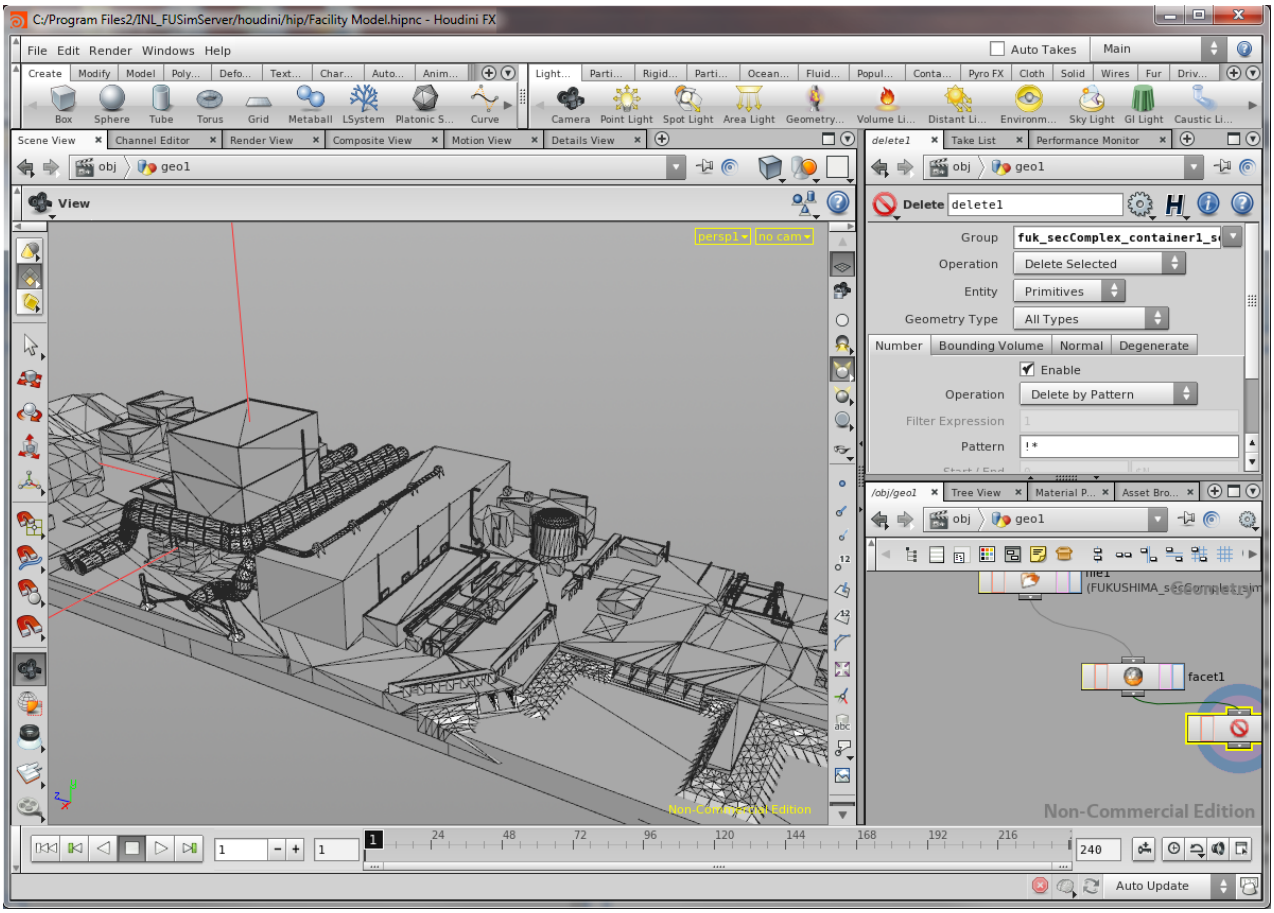


Figure 14: Houdini FX user interface

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. 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. 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

[1]. Although this option was present in Houdini to use its SPH solver, it was not suited for our needs. When Houdini's FLIP solver was used to generate a solution wave like a tsunami wave, 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 it had excessive numerical diffusion

5.2 Neutrino

To compare the results from Houdini as well as 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 is based on Smooth Particle Hydrodynamics with a pressure solve 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 [2].

For this purpose, Neutrino Industries collaborated with the Idaho National Laboratory (INL) by providing the use of their solve engine and making 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 simulation. Neutrino's simulation framework was flexible and provides a python based expression system (see Figure 2) to accurately model the movement of the wave machine based on Goring's 1978 numerical wave model (see Appendix A). Neutrino also provided a variety of tools to measure parameters in a section of the fluid simulated. 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.

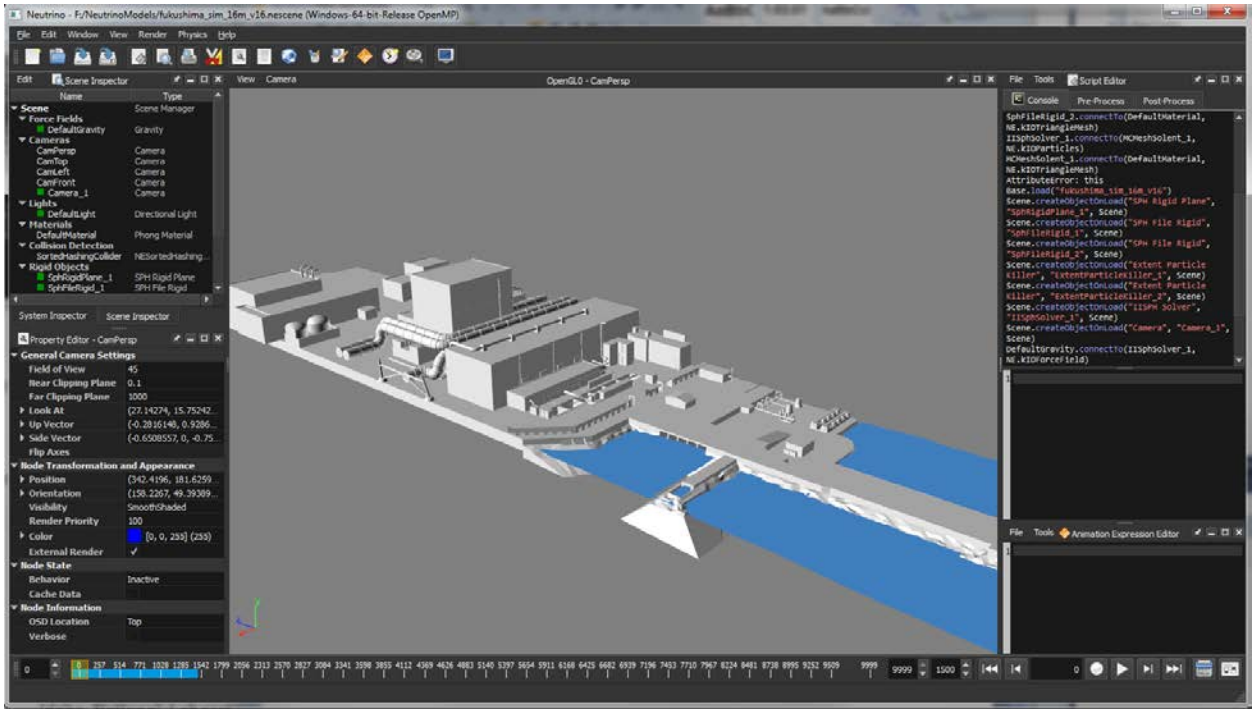


Figure 15: Example of the Neutrino modeling framework.

6. Linking Simulation with PRA

The demonstration case PRA model described in section 3.1 was modified to incorporate the failure events from a simulation. Before these modifications are made, baseline results are calculated to do a comparison of how much the simulation results affect the overall probability.

6.1 State PRA modifications

From the simulation, any component that comes in contact with water needs to translate into failure in the state diagram model. This is done by additional failure paths to the state diagram model for all components with a corresponding 3D item. For example, the "3D_Sim_Flooded" event is added to the "E-Pump-B_On" state to move to "E-Pump-B_Failed" when that message is received from the simulation. (See Figure 16)

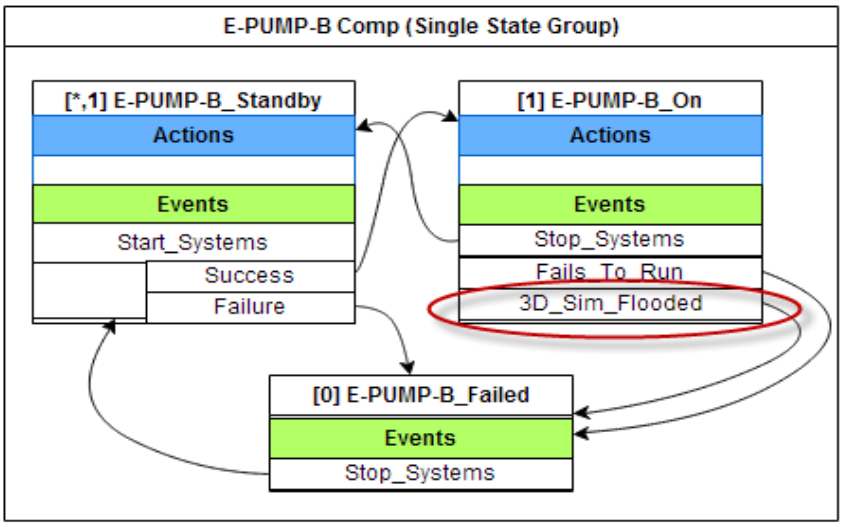


Figure 16 - Example of adding a 3D event to move to a failed state.

Next, an initiating event of a tsunami is added to the "Normal_Op" state which would trigger the movement to a tsunami state which starts the 3D simulation; the evaluation of the end states; and possibility of loss of power (see Figure 17). Finally an event transitions to a state which shuts down the 3D simulation and resets the state simulation after a given time frame.

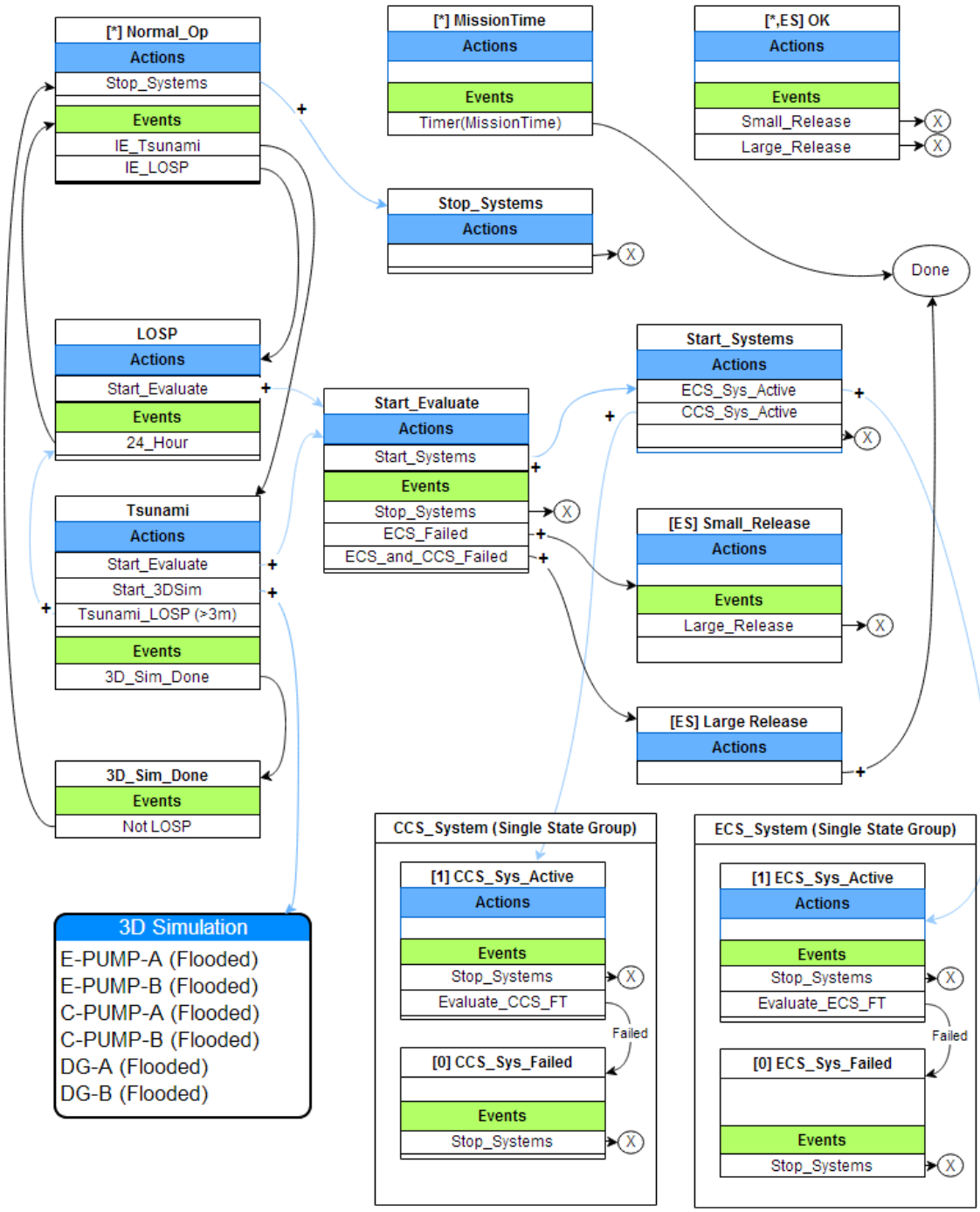


Figure 17 - State diagram model to incorporate 3D simulation results.

Each simulation that triggers a tsunami flood initiating event also computes flooding flow rates for the component room for the 3D simulation. These flow rates are determined by a sampling of the data given in Section 2. With the variation of flow rates in each 3D simulation, we get a distribution of the different effects on state simulation.

6.2 Initial Tsunami Simulation

A ¼ section of a facility was used to do initial tsunami testing. This simulation was used to test the feasibility of doing a simulation with a large number of particles (12 million), to generate a realistic wave, and to link data to State PRA simulations. Water levels at the bay door were measured for three different tsunami wave heights. This data was then used to extrapolate water flow levels for building flooding (Section 6.3) simulations of any tsunami height within the tested range (see Figures 17 and 18).

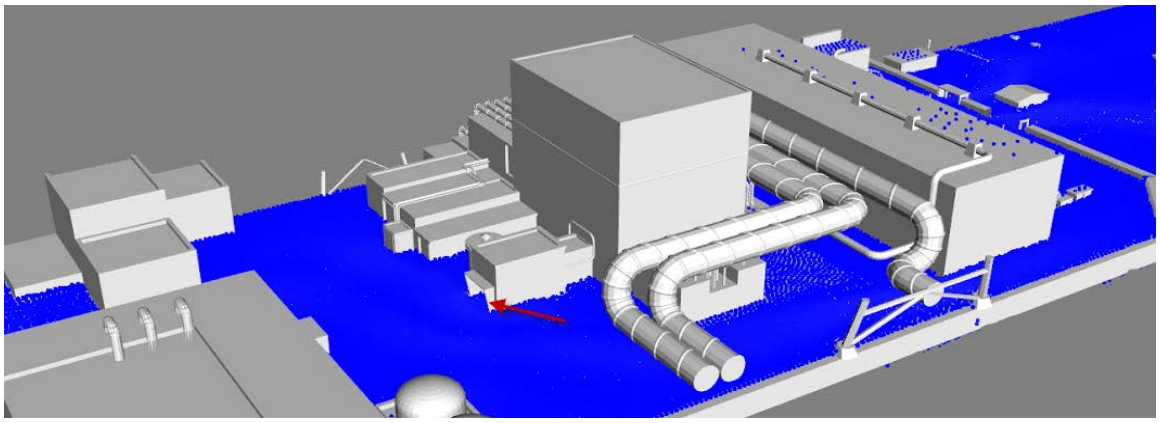


Figure 18: Water height levels over time for various tsunami waves measured at the bay door.

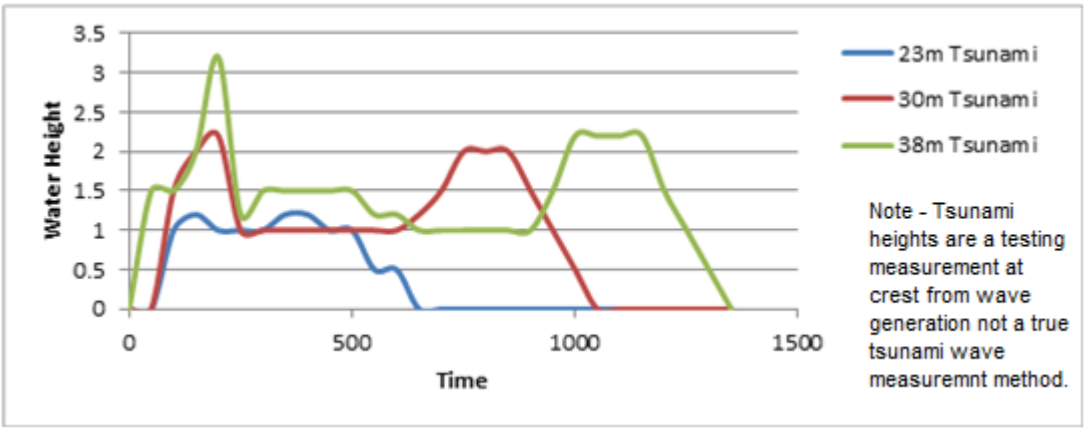


Figure 19: Measured water heights from different Tsunami waves.

6.3 Building Flooding

In the State PRA model, when a probabilistic run simulates a tsunami initiating event, it triggers a building flood simulation with flood rate data from the tsunami simulation. The building simulation is a flooding representation using a simple building with a couple rooms. A few components are placed in different locations of the building, which fail at different times depending on the flood rate. (See Figure 20) Each component failure is sent back to the State PRA model. This simulation is run multiple times, with various flood rates, depending on the tsunami data generated by each State PRA iteration.

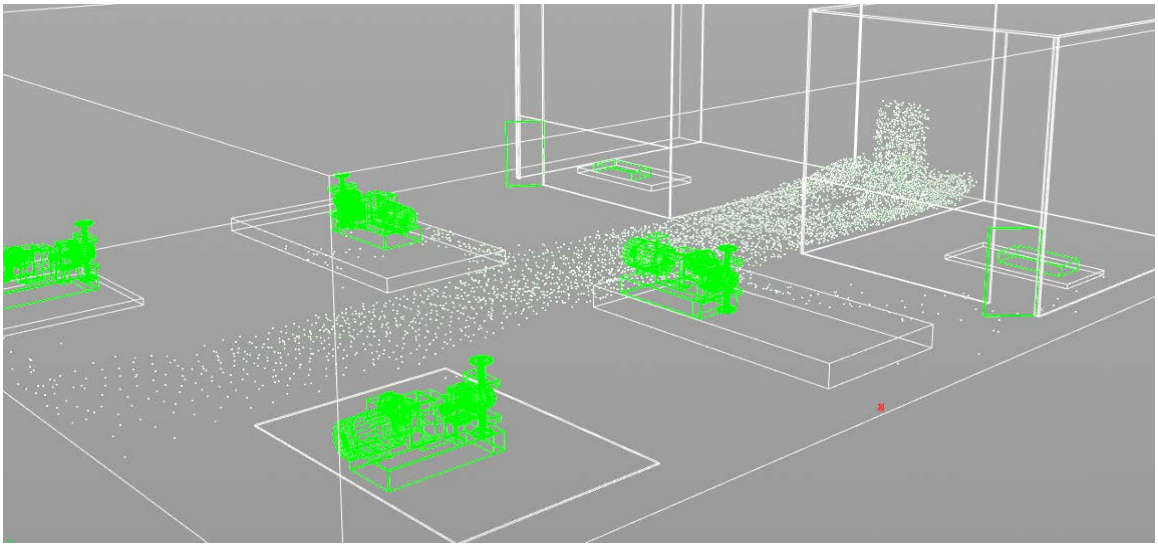


Figure 20 - A 3D model of a simple room with critical components from the Demo model.

6.4 State PRA Results

Information from the demonstration case project was used for the proof of concept simulation model. The only change made to the original example was the initiating event frequency for LOSP, where it was reduced from 2.3/yr to 0.1/yr to better reflect industry occurrences[a]. The state-based simulation model was then run 1,000,000 times to verify an accurate baseline with the SAPHIRE results. The results shown in Table 3 verify state simulation model is statistically equivalent to the SAPHIRE model. The higher the failure rate, the quicker the state based results converge on a valid answer.

Table 3. Baseline Simulation Results

End State	SAPHIRE Results	State-based Simulation Results
Small Release	2.53E-3/yr	2.59E-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 correspond to data given section 2.1. Each time a simulation is run, any tsunami events that could occur are then sampled for wave height data, also given in section 2.1. It is assumed for this analysis that any tsunami height above 3 meters also triggers a LOSP event for the simulation.

The simulations were run with the added tsunami initiating events but without the 3D simulation. The results show the increased failure caused by LOSP triggered by the tsunami and gives another reference point to determine any increase cause just by the 3D simulation.

[a] 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.

After incorporating the simulation failure events into the State Diagram PRA, it was run for 30,000 iterations when it started to converge on a result. In those runs, 215 significant tsunamis were encountered and simulated.

Table 4. Simulation Results Including Tsunami-cased LOSP Events.

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

7. Additional Simulations

7.1 Interior Flooding

Two interior flooding scenarios of the reactor building were performed as an example of simulation capabilities. The first consisted of flooding through the bay door into the interior of the reactor building. (See Figure 21) The second was to demonstrate water flow through the facility from a tank rupture. (See Figure 22) The Houdini FX module was used to simulate both scenarios. Because of the comparatively low number of particles, these internal flooding simulations took only a couple seconds per frame.

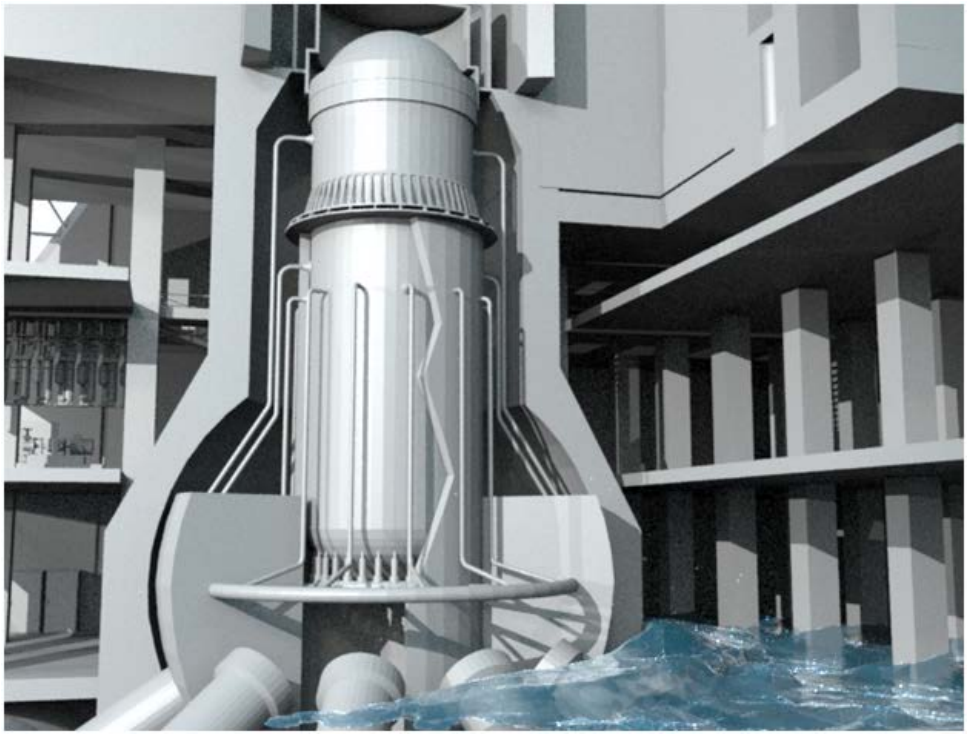


Figure 21: Reactor building flooding from the bay door.

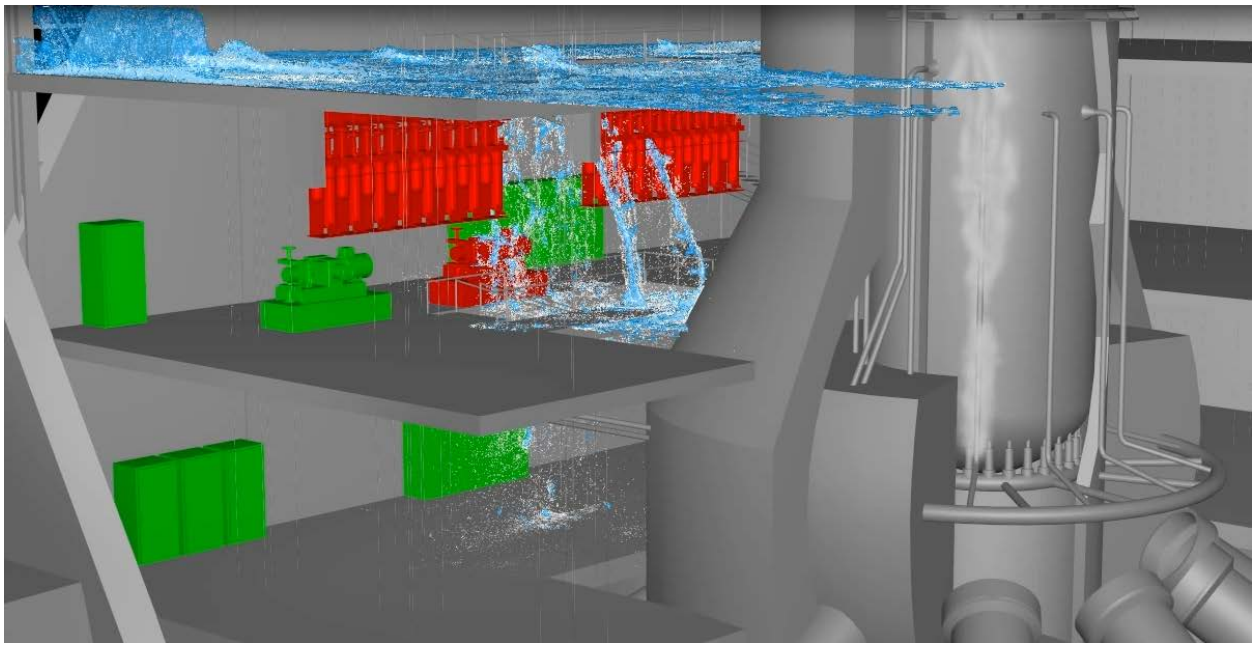


Figure 22: Tank rupture and water flow flooding example in a reactor building.

7.2 Facility modifications

Simulations can be an effective tool to optimize safety asset facility modifications. A demonstration simulation was done to show weak points of a facility for a given scenario. Water levels and pressures were measured on an exterior door and venting. This allows us to determine what level of tsunami would cause door failure and flooding into the air intake of the diesel generators. By making virtual modifications including a sealed reinforced door and moving air intake vents, we can protect these areas against a tsunami simulation of 19 meters, which failed the existing design. (See Figure 23)

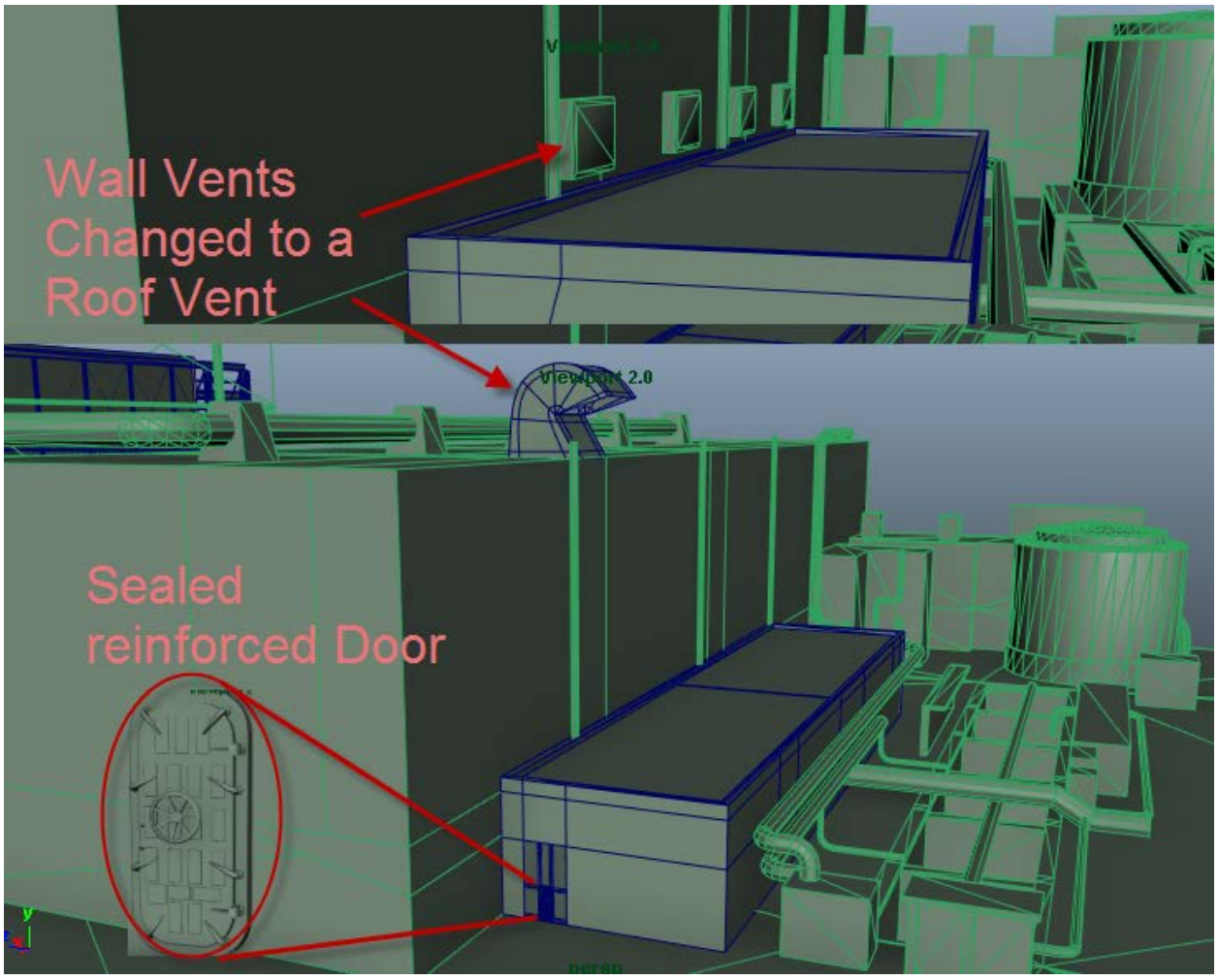


Figure 23: Facility Model Modifications

7.3 Larger Facility Simulation

The initial facility simulation model lacked accuracies such as actual terrain and wave formula representations. A second simulation was then constructed. It consisted of a ½ of the entire reactor facility including support building and representative terrain features. The ground terrain and ocean depth were obtained by using our custom mapping application as explained in section 4.2. Then, a boundary container was added to the model outside of the facility to represent the deeper part of the ocean. A bounding container was also modeled around the area of simulation with the measured ocean depth as its floor. The 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 piston machine which acted as a collision object was setup to follow Goring's 1978 model (see Appendix A) for generating waves. This wave piston (Figure 24) was placed at the far end in the "deep ocean." An additional rotational component of the wave piston was also generated using a sinusoidal equation to represent the wave part of the water.

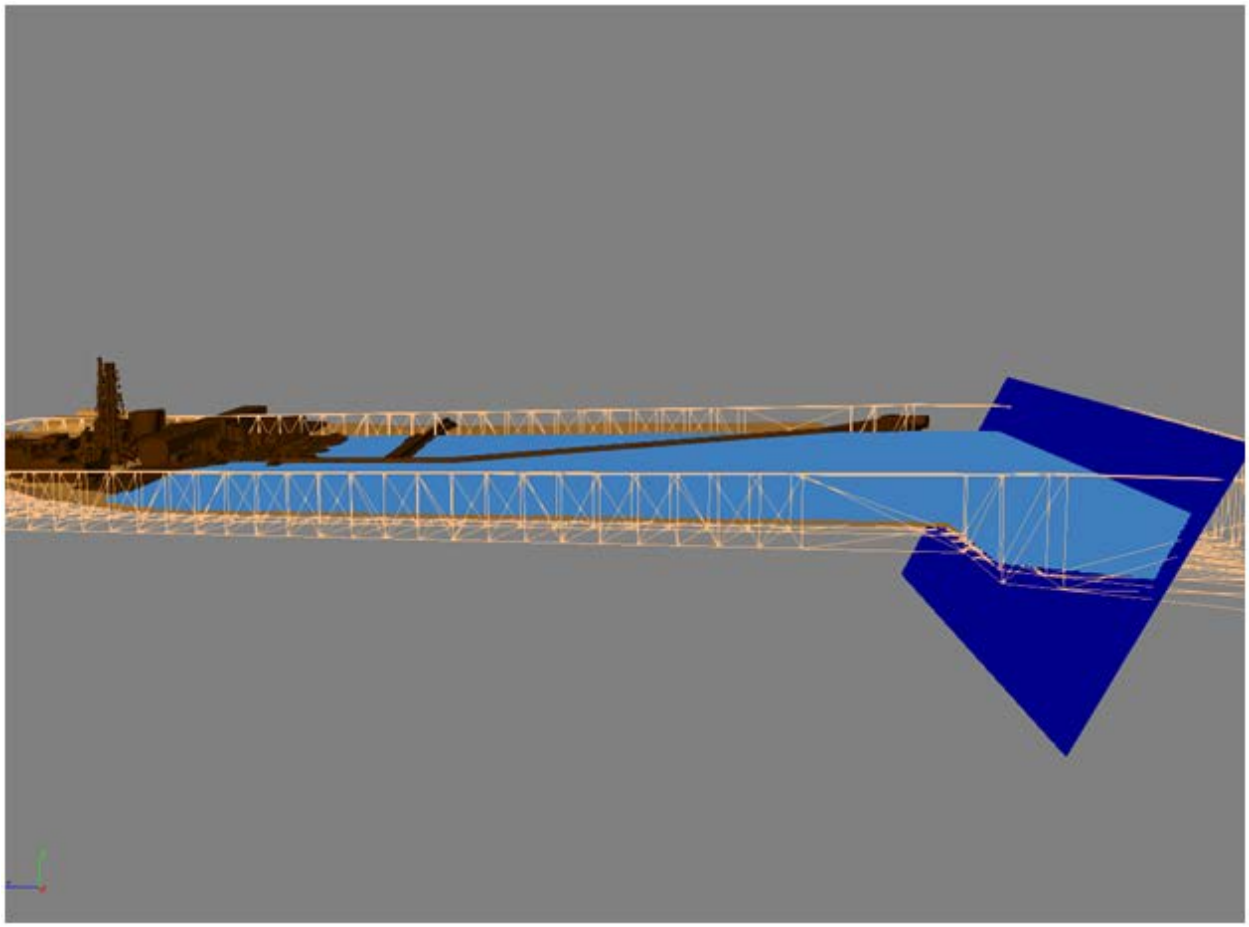


Figure 24: Neutrino Wave Piston Setup with Initial Conditions.

Using the new model we ran two simulations, a smaller wave and then a larger one, close to that which hit the Fukushima facility. The smaller wave simulation shows some vulnerable areas from water flow for a wave just over the sea wall height. The larger wave simulation can help to validate the accuracy of the results, by comparing the results to a real world event.

The way we originally measured the waves is different than done for the Fukushima tsunami event. We measured the peak of the wave at a cresting point in the ocean from the wave piston. The actual tsunami event was a measure of the wave height at the shore, not including any wave peaks. (See Figure 25) Using a similar measuring standard, we simulated an (approximately) 11m and 15m wave.

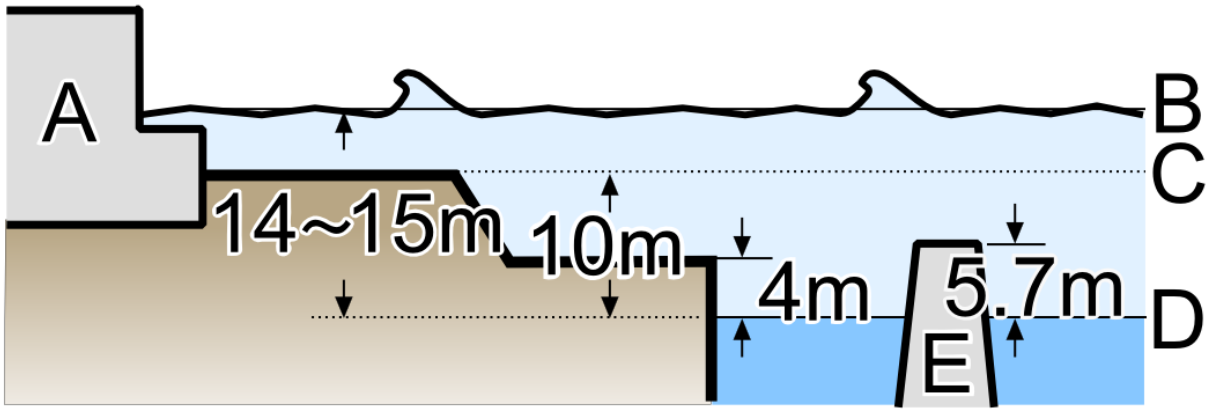


Figure 25: Fukushima tsunami height from Wikipedia.

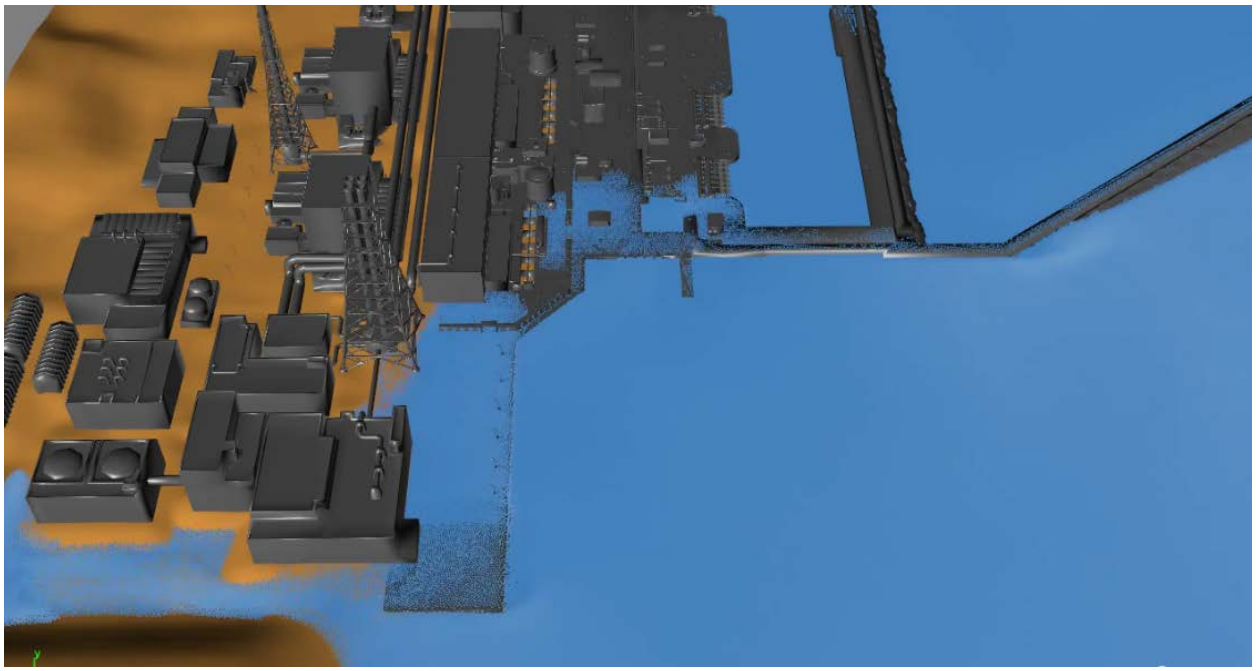


Figure 26: Half facility tsunami simulation 11meter wave inundation.

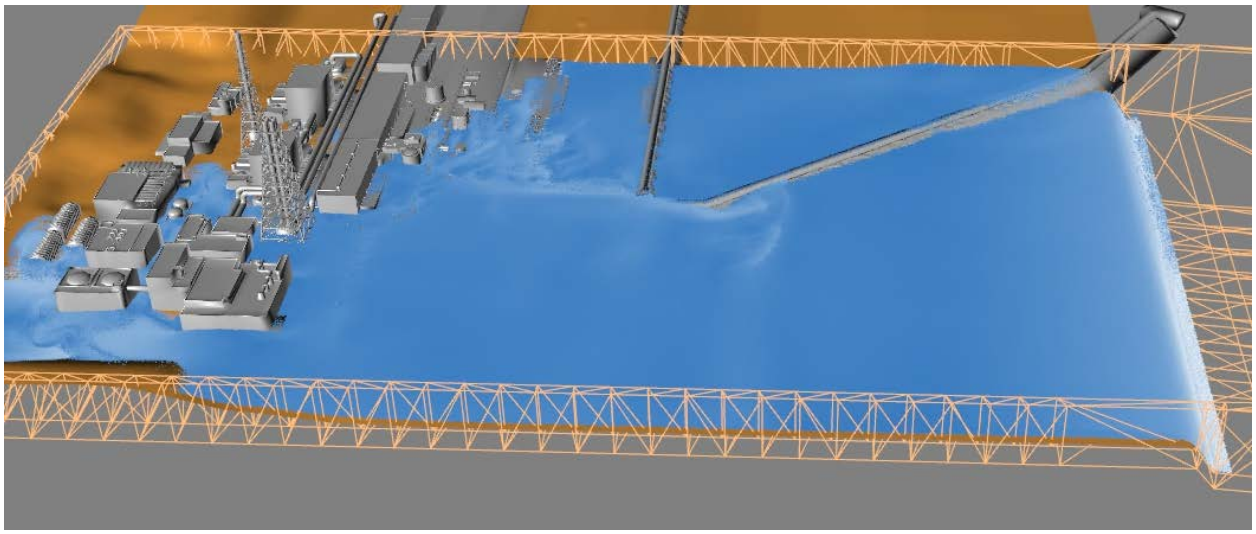


Figure 27: Half facility tsunami simulation 15 meter wave inundation.

8. Proposed Improvements

Because of shared memory requirements, these simulations were done on individual multi-core computers. Future work will investigate how to distribute the calculations in order to speed simulation results. For smaller simulations, such as inside the reactor, a GPU processor could be used to possibly give real-time results. This option is machine dependent and limited to small simulations depending on the memory capacity of the GPU. The second option is a dedicated co-processor which can add hundreds of additional cores to an existing machine. The principle is similar to the GPU but makes memory management much simpler because of the similar architecture to the main core and the code remains the same even if there is no co-processor on the machine. Lastly, distributed computing could be implemented to take advantage of clusters of networked computers. These High Performance Computing (HPC) systems could possibly run large simulations in close to real time (depending on the size of the system). However, all these options require modifications to 3D engine we are using in order to insure they work in unison with an effective memory management system.

9. 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 support a 3D modeling engine that can be used to represent risk analysis simulation for advanced small modular reactor structures and components. This capability is important since explicit, scenario-based analysis of plant safety will play a key role in licensing of small modular reactors, and significant emphasis will be placed on selection of events to be analyzed. Consequently, the need for extensive analysis, and representative models, supporting the safety case will be paramount.

10. REFERENCES

[1] http://en.wikipedia.org/wiki/Smoothed-particle_hydrodynamics

[2] 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.

[3] Goring, D. G. (1978). 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.

[4] Safinaz El-Solh (2012). 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.

[5] http://en.wikipedia.org/wiki/Computational_physics

APPENDIX A

A.1 Google Elevation API

Google's Elevation API places a limit on the number of possible query requests. Normally, a free user has up to request up to 2500 points per day. The API also tries to limit automated requests, so in order for the terrain map generator not to be blocked, time pauses were also added between requests. Optionally, Google provides an advanced subscription that can be to purchase which enables more requests. A terrain portion is determined by:

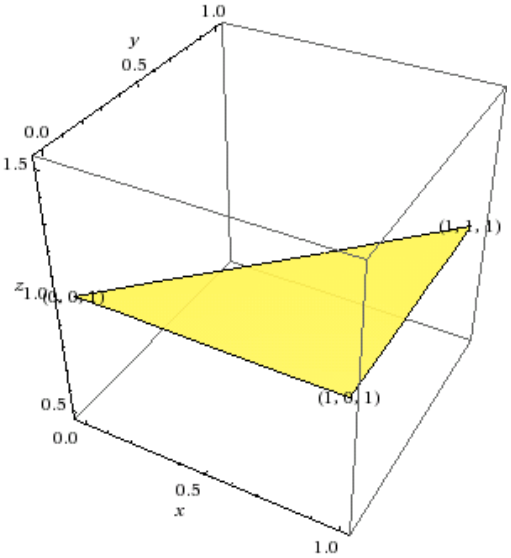
Haversine formula:
$a = \sin^2(\Delta\varphi/2) + \cos\varphi 1 \cdot \cos\phi 2 \cdot \sin^2(\Delta\lambda/2)$
$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$
$d = R \cdot c$
Where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km).
Note that angles need to be in radians.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

Secondly, the faces are generated to produce the texture from the vertices. Suppose the following were to be generated in the OBJ file:

v 0 0 1
v 1 0 1
v 1 1 1
v 0 1 1
f 1 2 3

This would generate a triangle connecting the three vertices numbered 1, 2, and 3. The number of the vertices is determined by their order from the top of the list going down.



(Visual Representation generated by WolframAlpha)

However the faces generated from the applet is generated as a rectangle and that only requires that the face generated have one additional vector number at the end.

A.2 Goring Solitary Wave Generation – Numerical Model

Goring (1978) 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:

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

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

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

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

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

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.