# GRAPHICAL EXPERT SYSTEM FOR ANALYZING NUCLEAR FACILITY VULNERABILITY

C. David Sulfredge, Robert L. Sanders, Douglas E. Peplow, and Robert H. Morris
Oak Ridge National Laboratory[*]
Oak Ridge, TN

## ABSTRACT

Nuclear facilities are a vital part of the nation's infrastructure providing approximately 20% of the nation's electricity and representing a vast investment of national resources. They also have the potential for serious radiological releases following an incident, which could adversely affect public health and contaminate large areas. Following the September 11th attacks, it has become clear that nuclear facilities rank very high among possible targets that must be safeguarded from terrorists. Conversely, mission planners need a targeting tool to estimate what sort of damage would be required to defeat operations at a nuclear facility and how serious the collateral consequences of an attack might be. The Oak Ridge National Laboratory (ORNL) is currently developing a novel graphical expert system, the Visual Interactive Site Analysis Code (VISAC), to address both of these needs.

For a single software package to meet these design goals, VISAC must integrate the three concepts of target geometric modeling, damage assessment capabilities, and event/fault tree methodology similar to that used in the nuclear industry for evaluating accident consequences. As currently written, VISAC is a Java-based graphical user interface (GUI) that can analyze a variety of accidents/incidents at nuclear or industrial facilities ranging from simple component sabotage to an attack with military or terrorist weapons. The list of damaged components from a scenario is then propagated through a set of event/fault trees to determine the overall facility kill probability, the probability of an accompanying radiological release, and the expected facility downtime. An initial version of VISAC is now being used for test calculations.

## ABOUT THE AUTHORS

**C. DAVID SULFREDGE** is a mechanical engineer with ORNL, where he works on facility vulnerability and various types of thermo-fluid modeling. He has developed numerous nuclear facility models together with event/fault tree algorithms for estimating downtime and calculating the probability of facility kill or radiological releases. He received his Ph.D. in mechanical engineering from the University of Kentucky in 1993.

**ROBERT L. SANDERS** is a nuclear engineer at ORNL, where he is involved with studying the vulnerability of various facilities (nuclear as well as chem/bio) to blast damage. His responsibility includes development of the facility vulnerability database. His previous experience includes modifying and enhancing MELCOR to meet the requirements of a fast-running PRA tool for the analysis of nuclear accidents.

**DOUGLAS E. PEPLOW** is a nuclear engineer in the Nuclear Science & Technology Division at ORNL. His current interests are in computational methods for facility vulnerability as well as Monte Carlo methods for radiation transport. He received his Ph.D. in nuclear engineering from North Carolina State University in 1999.

**ROBERT H. MORRIS** is the program manager for the facility vulnerability program in the Computational Sciences and Engineering Division at ORNL. He is a registered professional engineer with more than 25 years experience in the nuclear industry, including technical and project management, systems engineering, configuration management, on-site engineering support, and computer program development and applications.

[*] Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the Department of Energy.

# GRAPHICAL EXPERT SYSTEM FOR ANALYZING NUCLEAR FACILITY VULNERABILITY

C. David Sulfredge, Robert L. Sanders, Douglas E. Peplow, and Robert H. Morris
Oak Ridge National Laboratory
Oak Ridge, TN

## INTRODUCTION

In any country possessing nuclear facilities, they represent a vital part of the national infrastructure and a major investment of national resources. For example, in the United States 104 commercial nuclear power plants (69 Pressurized Water Reactors and 35 Boiling Water Reactors) currently supply about 20% of the nation's electric generating capacity (U.S. NRC, 1997). Thus nuclear power plants are prime assets that must be safeguarded in wartime and protected from terrorists in the aftermath of the September 11th attacks. Incidents at nuclear facilities also have the potential for serious releases of radiological materials, which can adversely impact public health and lead to long-term contamination of the surrounding environment. Public concern about terrorist strikes on U.S. nuclear facilities following September 11th has been shown by numerous accounts in the national media.

Nuclear facilities are also important targets for offensive operations because of their potential to be used for creating weapons of mass destruction. Hence mission planners need a targeting tool to estimate what type of attacks could defeat a nuclear facility, how serious any collateral consequences might be, and how long the facility would be out of operation. Oak Ridge National Laboratory (ORNL) is currently working to develop the Visual Interactive Site Analysis Code (VISAC), a novel graphical expert system that can answer such questions. VISAC can be used for both security-related analysis of U.S. plants or warfighting scenarios against nuclear installations ranging from power reactors to reprocessing plants.

## BACKGROUND OF VISAC

As currently written, VISAC is a Java-based graphical user interface (GUI) that can analyze a variety of accidents/incidents at nuclear facilities. Using Java allows VISAC to be platform independent, so that it can be run on any of the operating systems now in common use. To meet its design goals, VISAC must integrate the three concepts of target geometric modeling, blast damage assessment, and event/fault tree consequence analysis into a single code. Facility geometric information for both critical component locations and structural details of nuclear facility buildings is stored by VISAC in text files compatible with the BRL-CAD format popular in the weaponeering community (U.S. Army, 1991). This choice of format allows VISAC target models to be used by many other codes such as EVA-3D (Young, et al., 1995) and MEVA (Dunn, et al., 1999) that also rely on BRL-CAD geometric input. VISAC uses algorithms adapted from the EVA-3D/MEVA weapons effects codes for damage assessment. These correlations in turn date back to empirical test data developed by the National Defense Research Committee (NDRC) during the 1940's (White, et al., 1946). Correlations are available for breaching of concrete walls by a blast and the shock overpressure propagated through air spaces exposed to an explosion.

Event tree/fault tree methodology has been applied for decades in the nuclear industry for consequence assessment and determining the probability of reactor core damage during accident sequences (U.S. NRC, 1975). Basically, event trees track the progress of an accident sequence and define the safety systems that can be applied to avoid undesirable consequences. At each branch of an event tree, an underlying fault tree shows the critical components (connected by AND or OR logic gates) necessary for that safety system to function. VISAC adopts this modeling approach to calculate the probability of core damage and potential radiological release associated with attack scenarios at nuclear facilities, and adds another "facility kill" fault tree to determine the probability that the plant would be forced off line by the strike.

Ordinary event/fault tree calculations typically involve very small component failure probabilities, so that certain mathematical shortcuts based on cut set analysis such as the rare events approximation (McCormick, 1981) or the Esary-Proschan approximation (Isograph, 1996) are useful for evaluating sequence probabilities. On the other hand, in vulnerability analysis critical component failure probabilities tend to be high, so these approximations are no longer valid. Therefore VISAC had to introduce some unusual evaluation techniques based on Bayesian analysis and Monte Carlo methods to solve for the sequence probabilities efficiently. VISAC allows a user to set up the system logic model on a detailed basis using an arbitrarily large number of event trees and fault trees referenced to basic events that are critical components in the facility model. A coarser "building level" analysis is also possible, in which only the facility buildings are basic events in the logic model, and failure of a building implies loss of function for all critical components and systems located in that building.

The cut sets of VISAC's fault tree for facility kill are also vital for estimating facility downtime (DT). Each cut set of the facility kill fault tree consists of a group of equipment without which the plant cannot operate. The overall expected downtime from an attack scenario is thus given by assigning a kill probability ($P_i$) and a downtime ($DT_i$) to each cut set kill path and summing over the cut sets:

$$DT_{net} = (1/P_{kill}) \Sigma P_i * DT_i \quad P_{kill} \neq 0 \qquad (1)$$

where the overall expected downtime has been normalized by $P_{kill}$ (the overall facility kill probability) to make it conditional on facility kill. Equation (1) assumes a strictly serial repair process where damaged components are expected to be fixed sequentially one after the other. Parallel downtime algorithms have also been constructed that assume workers at the plant can repair the equipment associated with multiple cut sets of the facility kill tree simultaneously (Morris, et al., 1998). VISAC gives the user downtimes calculated by both serial and parallel approaches for each scenario.

Incident scenarios that can be modeled in VISAC range from simple component sabotage to an attack with military or terrorist weapons. VISAC is supplied with a library of facility models that can be edited through the VISAC GUI in both geometry and logic to match a wide variety of nuclear installations. The next three sections give some examples of VISAC's model editing capabilities and the range of vulnerability scenarios at nuclear facilities that can be analyzed with the code.

## EDITING FEATURES OF VISAC

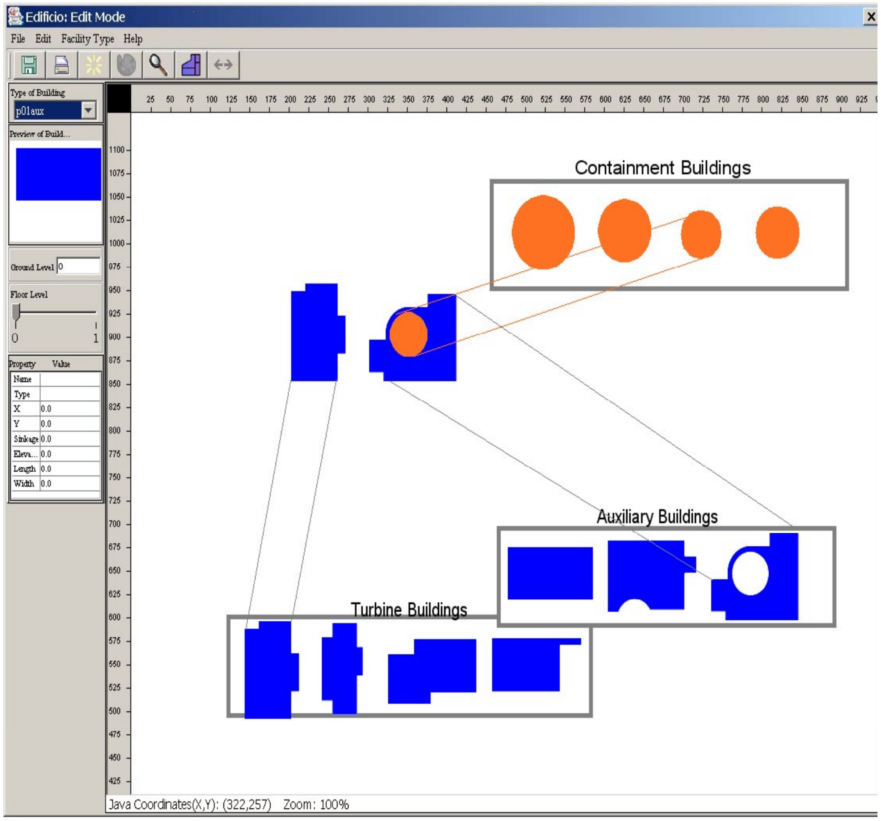
VISAC editing features are divided into two areas. The first is for the facility geometry model and the second is the event/fault logic model that describes both the required equipment for operation and the equipment required for safe shutdown. For the facility model, VISAC utilizes the standard geometry package BRL-CAD. This package was chosen for its compatibility with many other codes [e.g., EVA-3D, MEVA, PATRAN, ADINA, EPIC-2, PSARC, and NASTRAN (U.S. Army, 1991)]. A full description of the many uses of BRL-CAD can be found on ARL's BRL-CAD website http://ftp.arl.army.mil/brlcad/. However, unlike many CAD packages, BRL-CAD uses solid geometry; therefore, it is almost impossible to convert models from typical geometry packages that use surface orientation to BRL-CAD. All BRL-CAD model generation typically must be performed using the primitive graphical user interface MGED provided with the BRL-CAD package (U.S. Army, 1991). MGED is a GUI intended to provide a means of viewing the model and adding or modifying geometric bodies within it; nonetheless, its editing capabilities are less than desirable.

Because of the editing limitations found within MGED, a high-level geometry editing package has been developed within VISAC. The purpose of this editor is to provide a means of constructing a nuclear facility from basic building blocks within a few minutes instead of the days or weeks that it would normally take by using MGED.

This timesaving can be accomplished by assuming that most nuclear facilities are constructed of the same basic building types. For the construction of a simple Pressurized Water Reactor (PWR), several buildings would be chosen from a menu. Typically, the buildings that would be chosen are -- a containment building (that houses the reactor), an auxiliary building (that houses the support equipment), a turbine building (that houses the power production equipment), a diesel generator building (to provide backup power, sometimes located within the turbine building), a transformer building (can also be located within the turbine building), and an intake structure (to provide cooling water for the turbines). An example of this selection process is provided in Figure 1. Not shown in the figure is the ability to rotate the entire building and to show the internal structures (x-ray view). These features will be revealed in Figure 2.

It is interesting to note that all of the containment buildings shown look similar; however, each represents a significantly different design (e.g., one represents a small ice condenser unit while another represents a large dry containment). Also, separate containment buildings are provided for the three major U.S. vendors: Westinghouse, Babcock & Wilcox, and Combustion Engineering.

Once a facility is constructed, several modifications can be performed to make the facility match the specific site of interest. These adaptations include moving, adding, and rotating components within the buildings. Components can also be exchanged amongst the various buildings. Figure 2 shows a containment and auxiliary building. The components on the third floor of the auxiliary building are shown in red. As can be seen, the auxiliary building has been rotated 90$^o$ from



Type of Building: p01aux

Ground Level: 0

| Property | Value |
|---|---|
| Name | |
| Type | |
| X | 0.0 |
| Y | 0.0 |
| Sinkage | 0.0 |
| Eleva... | 0.0 |
| Length | 0.0 |
| Width | 0.0 |

Java Coordinates(X,Y): (322,257) Zoom: 100%

Figure 1. Building a simple nuclear facility from menu boxes.

what is visible in Figure 1. Each red component is intended to represent a piece of equipment that will be used in the fault trees. The red circles at the top of the auxiliary building, which correspond to emergency water storage tanks, are required if the intake structure becomes unavailable.

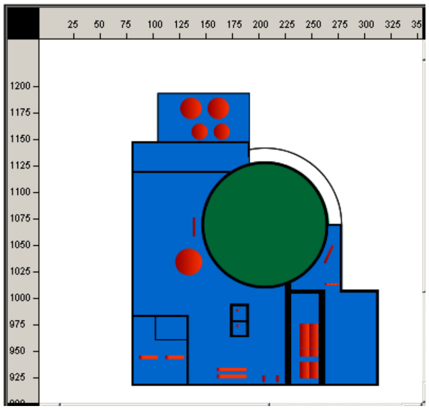


Figure 2. Components on the third floor of auxiliary building.

Figure 3 illustrates movement of these tanks along with several pumps (the boxes shifted from the bottom of the building to the top). The figure also demonstrates the addition of several components that represent control consoles, indicated by the black arrow. Lastly, Figure 3 illustrates the addition of a grid that can be used to aid the user in the proper placement of buildings.

Once the facility has been constructed using VISAC's geometric editor, the fault trees and event trees can be created/modified. Like the geometry model, a library is provided with the code that contains event and fault trees for various types of nuclear facilities. All that is required is for the user to choose a basic facility and then customize it to match the specific facility of interest. This editing of the event/fault trees is performed in a similar graphical fashion to the editing of the geometry model.

When the VISAC event/fault tree editor is started as shown in Figure 4, any of the event trees may be viewed by using the tabs. The bar consisting of a line of rectangles (name boxes) at the top of the event tree panel, represents the underlying fault trees. The menu bar at the top of the window allows users to perform the following basic tasks:

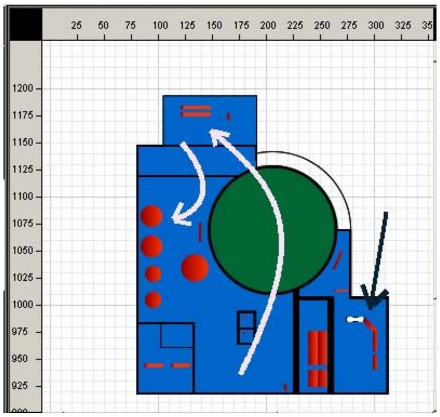
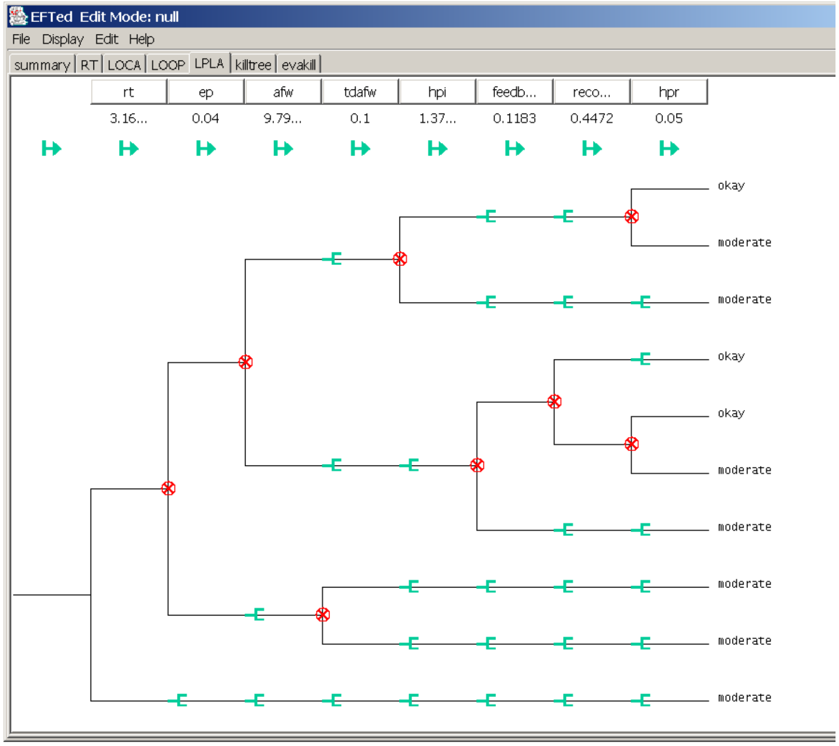


Figure 3. Movement of water storage tank and creation of new components.



| rt | ep | afw | tdafw | hpi | feedb... | reco... | hpr |
|---|---|---|---|---|---|---|---|
| 3.16... | 0.04 | 9.79... | 0.1 | 1.37... | 0.1183 | 0.4472 | 0.05 |

Figure 4. Typical event tree editor window.

- Add/Edit consequences, which are placed at the end of each sequence
- Add a new Event Tree, adds a new tab
- Edit Event Tree properties (its name and its description)
- Delete an Event Tree, removes a tab

Within the event tree that is shown in Figure 4, top level systems that branch the event tree can be added or deleted, changing the sequence dependencies on the fault trees given by the name boxes at the top of the screen. When a new top level system is added (increasing the number of branches) the number of possible sequences (paths through the event tree) will also increase. The event tree for "LPLA," a simultaneous loss of off-site power and loss of coolant accident, is shown in Figure 4. The graphical editing targets (green and red icons in the figure) can be toggled on/off using the "Display/Toggle Edit Targets" menu bar.

The following editing actions can be made to an event tree utilizing the graphical editing targets within the tree display: Clicking on a green arrow can add or delete an underlying fault tree; selecting the green fork splits the branch (thus adding a dependency on the failure probability of the underlying fault tree to the selected sequence); and lastly by choosing the red circled “X,” a branch can be deleted (removing the dependency of the sequence on the underlying fault tree). When a new branch (sequence) is created, the user must also assign a consequence (i.e., assign whether the sequence results in a desirable outcome or an accident). The top level underlying fault trees may be viewed and edited at any time by clicking on the name box located at the top of the screen.

The editing features of the fault tree window provide the basic functions found in most graphical fault analyzers [e.g., see Fault Tree Plus (Isograph, 1996)]. A sample fault tree for editing is shown in Figure 5. The first such function is copying a portion of a fault tree to another tree or for the creation of a separate new tree. When copying a gate, the gate will move to its

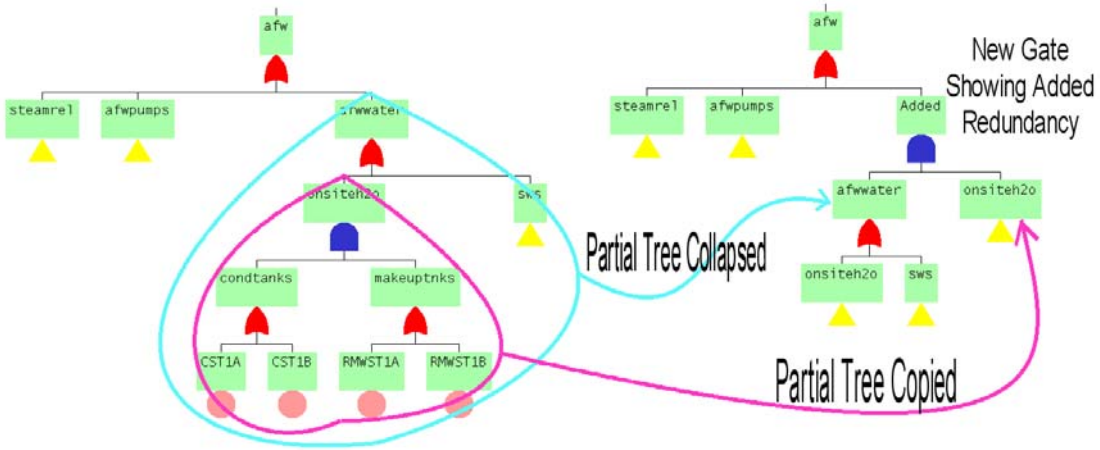


Figure 5. Illustration of modifying Auxiliary Feedwater Fault Tree.

own fault tree window and both the old and new locations of that gate will be replaced by the TRANSFER gate symbol, as shown in Figure 5 (light blue line). Copying can be performed utilizing the typical cut and paste process or by duplicating the selected portion of the fault tree, as illustrated in the figure (violet line). The cut option is complemented by the ability to delete individual or whole sections of the fault tree.

Several options are available to the user to add new branches to a preexisting fault tree. The simplest is the addition of a basic event, which corresponds to a new component in the graphical model builder, to a preexisting AND or OR gate. The user can also construct new gates in preexisting trees by building upon the various basic events. The basic events added can be either from the library geometrical model or components added during the geometric model editor phase described earlier. It is important to remember that a component is only available to the fault tree editor after it has been added to the geometric model. Currently, only the OR and AND gate operators are available to the user when adding new gates. To prevent trees from becoming excessively large for viewing, a TRANSFER gate can be added. This gate will bring up a separate continuation window for that fault tree. Using either the TRANSFER function or the event tree pane, new fault trees can be created.

## VISAC MODEL VIEWING OPTIONS

VISAC contains several different options to view the geometric model. The first utilizes the viewing features of the geometry editor; however, no modifications of the model are allowed. The second is a 2D viewer that provides slices (cut planes) through the facility in three separate windows, XY, XZ, and YZ. The view is controlled by three slider bars, represented as blue lines in the cut planes, that allow viewing of any combination of X, Y, and Z. Each cut plane shows the materials in that slice. Standard colors for a VISAC geometry display (not all of which are used in Figure 6) are: Void-white, soil-green, concrete-gray, rebar-yellow, rock rubble-light gray, steel-blue, critical components-red, broken critical components-cyan, holes generated by the blast program-dark gray, anything else (air)-pink. A specific area of the model can be zoomed by drawing a box around it. This will change the images in the selected view and the other views sharing those coordinate axes. Figure 6 shows an example image from this viewer.

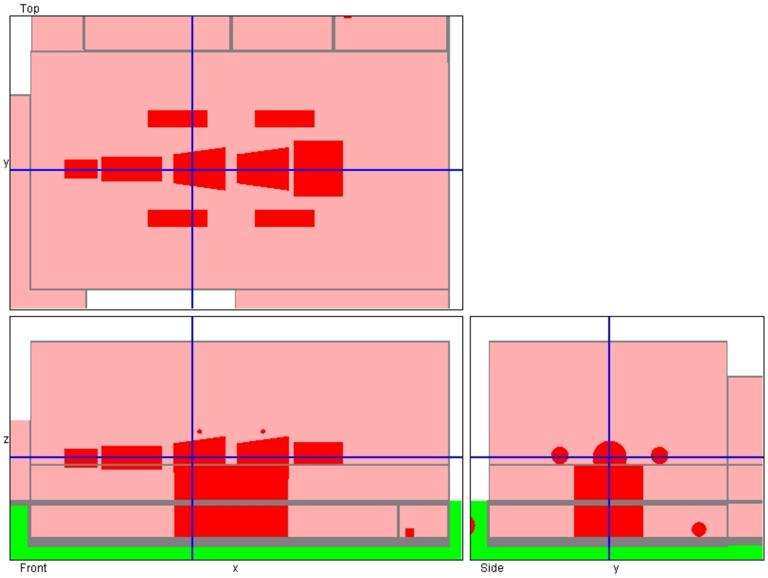


Figure 6. Example plant image with the 2D viewer, showing a triple orthographic projection of a reactor turbine building.

The last viewer option is a ray-tracing mode that provides true three-dimensional viewing of the model as illustrated in Figure 7. Because of the model complexity, this is the slowest option; however, it can produce excellent images at full resolution. The user can move the viewing point around the plant as desired. Zooming in and out can be performed on the overall image or by drawing a box around a selected area.

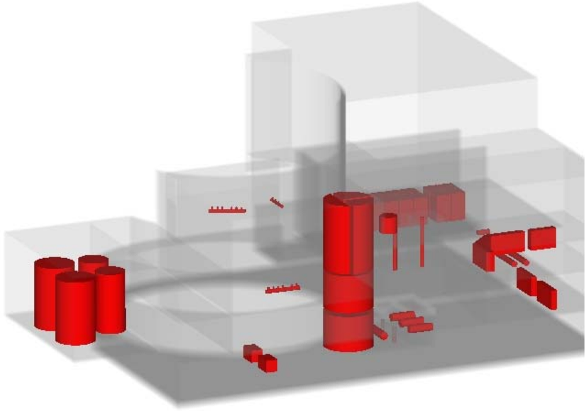


Figure 7. Example 3D viewer image.

To help see internal components, users can make use of a cutter box to remove a section of the geometry. The cutter box defines an area where nothing will exist. Another way to help visualize the interior of facilities is to make some of the materials translucent. Each material can be assigned a color by the user and an alpha (opacity) value. An alpha of 1 makes a material opaque and an alpha of 0 makes a material completely invisible. An alpha between 0 and 1 will make a material partially transparent. For example, with alpha=0.1, a material will be transparent up to 10 ft thickness. After 10 ft, that material will appear opaque. The geometries for the VISAC library plants have collections of regions called groups. These groups can be selected or de-selected for display.

## EXAMPLE VISAC CALCULATIONS

VISAC uses three different types of incidents to damage components or systems:

1. component damage – user selects components and assigns failure probabilities (to simulate sabotage),
2. quick blast – a blast overpressure is propagated through the geometry model giving each critical component some fractional failure probability, and
3. quick blast from grids – user can select a point from a grid of previously calculated quick blast runs based on the desired results of the blast.

A series of incidents can be defined so that cumulative damage can be analyzed. One of the assumptions in VISAC is that the time interval between incidents is short compared to the downtime of the damaged equipment. Under this assumption, anything damaged in incident 1 will still be damaged for incidents 2, 3 etc, no matter how much time elapses between incidents.

The following examples will be based on a generic two-loop PWR model that does not represent any particular real plant. The logic model uses 151 critical components combined using 129 gates to form 19 top-level fault trees, each with some intrinsic failure probability (called simple failure) independent of any damage to the facility. Two overall consequences are calculated – facility shutdown and core damage. The core damage event tree is quite lengthy and is broken into a master and four slave trees. The master tree determines the probabilities with which each of the other trees are followed: RT (reactor trip), LOOP (loss of offsite power), LOCA (loss of coolant accident) and LPLA (LOOP and LOCA). The geometry model used in the example has five buildings that house the critical components.

### Component Sabotage

VISAC allows the user to directly select components of the logic model from either a listing of all equipment in the plant or from the 2D or 3D viewers. In this example, one of the plant's steam generators and the electric switchyard are both assigned failure probabilities of 50%. These probabilities are propagated up the fault trees through the AND and OR gates [like the LOOP gate that is shown failed by a red "X" in Figure 8 due to loss of the switchyard (S)]. The facility kill fault tree consists of an OR gate of many plant systems needed for continued plant operation, two of which are the steam generators and the switchyard. Based on independent failure probabilities of 50% for each of these two components, the facility kill fault tree then has a failure probability of 75%. The downtime associated with a steam generator is 9 months and with the switchyard is 2 months. Using the procedure outlined in the background section gives a parallel downtime estimate of 6.67 months and a serial estimate of 7.33 months.

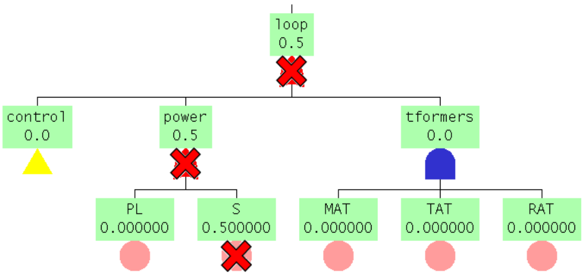


Figure 8. The loss of offsite power fault (LOOP) tree with damage propagated up from the damaged basic events.

Total probability of core damage is calculated to be 4.27%. As expected, this reflects an increase over the 0.35% probability of core damage from simple failures alone. The particular damage set just selected gives each of the four branches of the master core damage event tree a 25% probability (see Figure 9), so that RT, LOCA, LOOP, and LPLA outcomes are equally probable. The event tree for the LPLA branch is shown in Figure 10, with its most likely sequence leading to

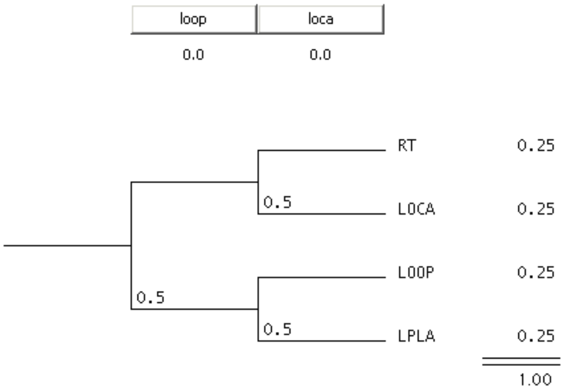


Figure 9. The master event tree for calculating core damage probabilities. This tree refers to four more complex trees.

core damage highlighted. Also shown in the figures are the probabilities associated with each sequence and the probabilities associated with each branch point in the event tree. The simple failure probabilities for each top-level system are shown beneath the system name.

In Figure 10, the sum of the sequence probabilities for the LPLA tree is 25%, the total probability of a LPLA event in the master tree from Figure 9. Also shown are the top-level gate simple failures (below the gate names) and the probability of system failure at each branch in the event tree.

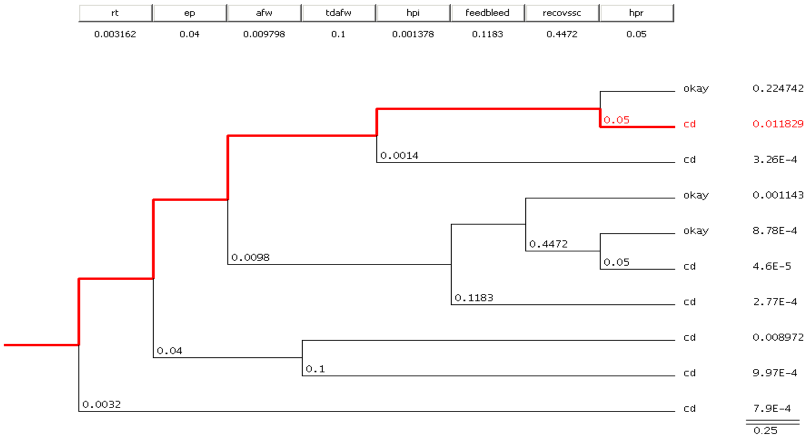
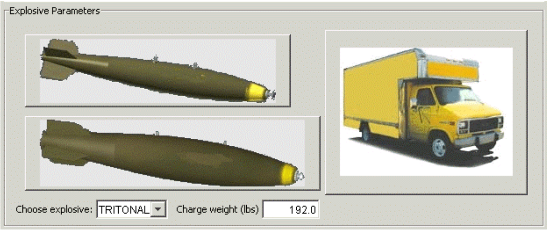


Figure 10. The LPLA (loss of offsite power and loss of coolant) event tree.

### Attack with Military Weapons

The VISAC Quick Blast utility allows the user to set up an incident by picking a blast location using one of the geometry display packages. The user can pick a point inside or outside a building. The charge type and weight are also input by the user or can be selected from a list of common military or terrorist weapons: a 500-lb general purpose (GP) bomb (192 lbs tritonal), a 2000-lb GP bomb (945 lbs tritonal), or a large truck bomb (3500 lbs TNT equivalent). Figure 11 shows the graphical weapon selection screen.

Three modes of calculating damage can be used. The user can try different modes and different calculation options for the application before saving the incident.



Explosive Parameters

Choose explosive: TRITONAL

Charge weight (lbs) 192.0

Figure 11. Graphical weapon selection screen in VISAC.

The three modes of calculating damage are:

1. ray-tracing mode - This mode sends out rays uniformly in different directions. Along those rays, walls are broken and critical components are damaged. With more rays, the damage description gets more accurate but takes longer to compute. The pressure of the blast is evaluated on the surface of the critical component where the ray strikes it.
2. EVA-3D mode - This mode emulates the EVA-3D weapons-effects code (Young, et al., 1995) and uses the rays only to break walls. Critical components in rooms affected by the blast are then tested for damage by evaluating the pressure of the blast at the centroid of the component. This mode tends to break more components since it can "see" around corners that the ray trace mode cannot. The drawback is that component damage is evaluated at the centroid, which may not be at the same overpressure as the point closest to the blast.
3. direct mode – This approach is similar to the ray-tracing mode, except that rays are only thrown out in the horizontal plane and towards the centroid of each defined critical component. For facilities with small critical components, this mode may be faster than using the standard ray tracing technique.

For this example, a 500-lb GP bomb is targeted at the facility control room, located in the lower left corner of the auxiliary building at the intersection of the blue crosshairs (see Figure 12). In this example, using 24 rays in the horizontal plane, the three different calculation methods give the same results – the four control room panels are destroyed with probabilities of 100%. The damage calculated by the quick blast utility can be viewed with the 2D or 3D viewer described earlier.

The list of damaged critical components is then input into the logical event/fault trees. The fault trees and event trees are then calculated giving results of 100% probability for facility shutdown and 100% probability for core damage.

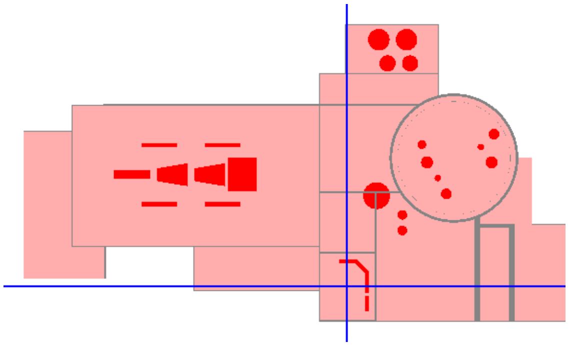


Figure 12. The quick blast target location, where the two blue lines intersect, is the control room.

### Blast Grid Analysis for Terrorist Truck Bombs

For any facility in VISAC, the user can define a grid of points and run the quick blast utility for all of them, producing a map of facility shutdown and core damage. The user specifies the spacing of the grid, all of the quick blast parameters, and whether to use points external to buildings, internal points, or both. Once all of the points have been computed, the user can then search for points that meet certain criteria. For example, points that have a high probability of facility shutdown with little or no probability of core damage may be of interest to a mission planner. Plant security could use the maps to determine where weak spots are and then take corrective actions.

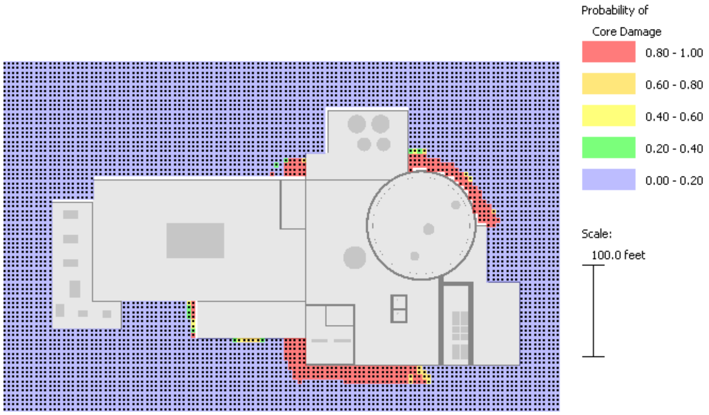


Figure 13. A grid of quick blast calculations for a large truck bomb using a detailed geometry.

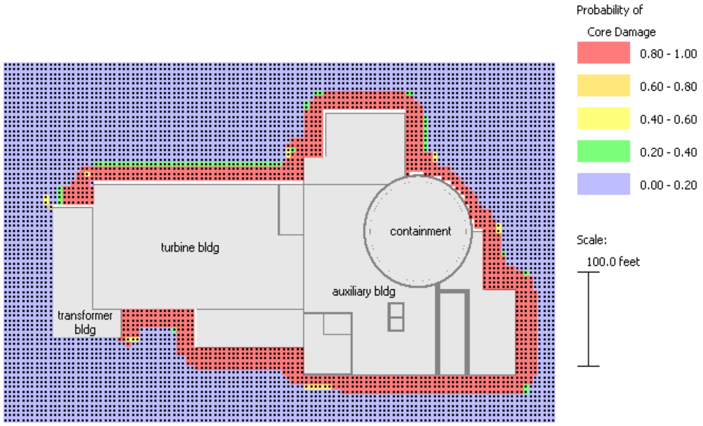


Figure 14. A grid of quick blast calculations for a large truck bomb using only a building level geometry.

For example, a grid can be calculated for a large truck bomb, exploding 10 ft above the ground at points external to the buildings of the hypothetical generic PWR model. This is shown in Figure 13. This map shows the areas where the truck bomb breaks through the walls and damages critical components. The blast points are black dots and the color indicates the probability of core damage for that grid location. Areas where core damage is likely are next to the containment, near the control room, and near other vital equipment in the auxiliary and turbine buildings. All these buildings can be located by comparing Figure 13 with Figure 14 where the building names are annotated.

The maps produced depend on the fidelity of the model - for basic level plants, the maps only indicate which buildings, if damaged, lead to core damage (see Figure 14). This figure shows that breaching any building of the hypothetical plant other than the transformer building will lead to core damage. Models at this level of detail do not give much insight into the problem of which parts of the plant need to be protected. The map created using the detailed model (see Figure 13) gives more useful information.

## FUTURE WORK

An initial version of VISAC is now being used for test calculations by ORNL. In the near future, several enhancements are planned to make the code more versatile and user-friendly. The VISAC facility library will soon be expanded to include a total of 16 water-cooled reactors, one gas-cooled reactor, and two reprocessing plants. Currently, only one general core damage outcome is available from many of the event/fault tree models for radiological releases. To improve the accident fidelity in both magnitude and timing, six new accident consequences are being incorporated into the VISAC event trees (large early release, large late release, large intermediate time release, small early release, small late release, and small intermediate time release). It is also planned to introduce additional ordnance details such as weapon CEP (Circular Error Probable) to attack scenarios involving military weapons.

Another goal for future work involves integrating VISAC with several other codes currently under development. MBLM/STEP includes models for transport of radioactive aerosols from an accident through the facility and an estimate of the quantities of such material released from the facility to the environment (Pierce, 1996). These codes should be particularly useful for quantifying releases from reprocessing plants. STEP also provides a link between MBLM results and HPAC (SAIC, 2001), a code for tracking atmospheric dispersion of any releases and estimating radiation doses to affected populations. The combination of VISAC with MBLM/STEP and HPAC will create a powerful tool for looking at all aspects and potential consequences of nuclear facility vulnerability.

## CONCLUSIONS

VISAC integrates the three concepts of target geometric modeling, blast damage assessment, and event/fault tree consequence assessment in a new way to analyze nuclear facility vulnerability.

VISAC's graphical editing routines can customize the library models supplied with the code to match a broad range of actual nuclear installations around the world.

VISAC can analyze a wide variety of vulnerability scenarios at nuclear facilities, including sabotage and an attack with military or terrorist weapons. For each scenario, the user can determine facility kill probability, the potential for a radiological release, and the expected downtime.

Work is currently underway to expand the VISAC facility library and integrate VISAC with other vulnerability codes.

## REFERENCES

Dunn, P. E., J. E. Madrigal, D. A. Parsons, J. C. Partch, D. A. Verner, and L. A. Young, *Modular Effectiveness/ Vulnerability Assessment (MEVA) Software User's Manual*, Applied Research Associates, Inc., Albuquerque, NM, April 23, 1999.

Isograph Ltd., *FaultTree+ Software Documentation*, 1996.

McCormick, N. J., *Reliability and Risk Analysis*, Academic Press, New York, 1981.

Morris, D. G., C. D. Sulfredge, R. H. Morris, R. L. Sanders, and W. R. Nelson, *Vulnerability Analysis of a Midsized Commercial Pressurized-Water Reactor*, ORNL/TM-13670, Oak Ridge, TN, October 1998.

Pierce, T. H., *MBLM Version 42c User's Reference Manual*, DTR-96-15508, San Diego, CA, September 1996.

SAIC, *Hazard Prediction and Assessment Capability (HPAC) User's Guide Version 4.0*, San Diego, CA, August 15, 2001.

U. S. Army Ballistic Research Laboratory, *The Ballistic Research Laboratory CAD Package, Release 4.0*, Aberdeen, MD, December 1991.

U. S. NRC, *Reactor Safety Study-An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, Report WASH-1400*, NUREG-75/014, 1975.

U.S. NRC, *Information Digest*, NUREG-1350, Vol. 9, 1997.

White, M. P., et al., *Effects of Impact and Explosion*, Summary Technical Report of Division 2, NDRC, Washington, DC, 1946.

Young, L. A., B. K. Streit, K. J. Peterson, D. L. Read, F. A. Maestas, *Effectiveness/Vulnerability Assessments in Three Dimensions (EVA-3D) Versions 4.1F and 4.1C User's Manual – Revision A*, WL-TR-96-7000, Applied Research Associates, Inc., Albuquerque, NM, November 29, 1995.