DPL™ 9
Fault Tree
User Guide

www.syncopation.com

Printed in the United States of America.

Revised April 2018.

Table of Contents

1. Introduction

1.1 Welcome to DPL 9 Fault Tree

This *DPL 9 Fault Tree User Guide* describes the features and functionality of DPL 9 Fault Tree. It begins with an introduction to fault trees, although some familiarity with fault tree concepts is assumed.

This manual is intended to be read while working with the DPL Fault Tree software. The tutorials in many of the chapters follow-on from previous chapters. In particular, the tutorial beginning in Chapter 2 continues through Chapter 5.

A few conventions have been used in the text of the tutorial chapters. An instruction to you in a tutorial will be contained in a bulleted paragraph with an arrow, as follows:

⇨ Please do this step now.

Information to be entered in dialog boxes is typically contained within double-quotes. Do not include the double-quotes when entering the information.

DPL ribbon tab names, groups, and commands are separated by vertical bars (|). For example, "FAULT TREE | Node | Binary Event" refers to the Binary Event command in the Node group on the FAULT TREE tab of the ribbon as show in Figure 1-1.

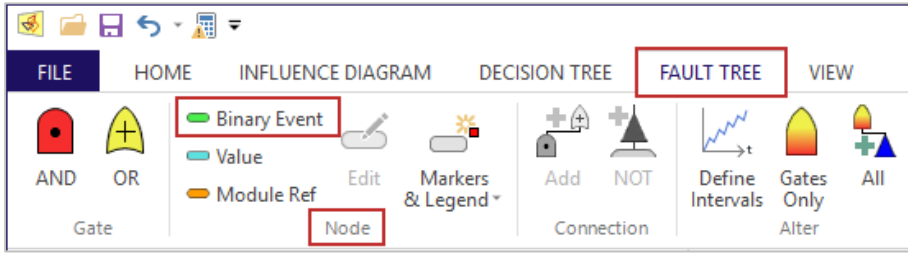

Figure 1-1. Manual Convention: FAULT TREE (Tab) | Node (Group) | Binary Event (Command)

Within this user guide the word "Options" following a vertical bar refers to the dialog box launcher ( ), which is located at the bottom right corner of the given tab group. For example, "HOME | Run | Options" refers to the

dialog box launcher for the Run group on the HOME tab of the ribbon and will launch the Run Settings Dialog.

If there are multiple tabs within a dialog, then the item following the last | refers to the desired tab. For example, FILE | Options | Outputs refers the Outputs tab of the Options dialog accessed from the FILE tab.

1.2 Contents of this Manual

A brief outline of the contents of this manual follows.

Chapter 2 introduces fault trees and their component parts.

Chapters 3 contains a tutorial on how to build a fault tree in DPL.

Chapter 4 provides an overview of how to create, manipulate, and interpret a circuit diagram view of a fault tree.

Chapter 5 covers two main fault tree outputs: cut sets and partial derivatives.

Chapter 6 discusses the use of true/false costs in fault trees. Furthermore, this chapter covers fault tree inversion and how it can be leveraged with tree/false costs.

Chapter 7 provides information on embedding modules both within fault trees and decision models.

Chapter 8 provides information introducing time intervals into a fault tree and how to generate a time series output.

In DPL Fault Tree, fault trees may be used in conjunction with DPL decision models. Chapter 9 covers incorporating fault tree modules into decision models. If you intend to use this feature and are new to DPL, you should review the contents of the *DPL 9 User Guide* before proceeding to the features documented in this chapter. You may also wish to complete the tutorials contained in that manual. The *DPL 9 User Guide* also contains information on how to install DPL and how to access resources and get help.

2. What is a Fault Tree?

A fault tree is a hierarchical model that is used to analyze the probability that an event will occur. The event is typically a low probability, high consequence risk or outcome such as the failure of a critical system or a breach of security. This probability or outcome is referred to as the top-level event. In a fault tree, a top-level event is broken down into lower level component faults or failures (also called events). These lower level events are combined together with gates to calculate the probability of higher level events until you reach the top event. There are two types of gates in a DPL Fault Tree: AND Gates and OR Gates which behave similar to the corresponding logical operators. The fault tree provides a graphic representation of the breakdown of events and the gates that connect them. The fault tree also dictates the mathematical operations necessary to combine together events to calculate the overall probability of the top-level event.

The motivation for modeling a risk with a fault tree is based on the observation that the probabilities of each lower level, component event can be more reliably assessed than the probability of the more complex top-level event.

Fault trees are unique in their ability to gracefully handle graphic representations of large-scale problems. A fault tree provides a means to better understand the risks involved in a given situation and how they relate to each other. This leads to a more accurate risk assessment and a quantitative method for assessing the merits of actions that may reduce risk.

2.1 Events

An event in DPL Fault Tree is something which either occurs or does not, i.e., it is either true or false. Events can be represented by binary nodes (see Section 2.2), gates (see Section 2.3) or modules (which will be covered later in Chapter 0). The output of any event is the probability that it will occur.

It is customary to define events such that their true states are the lower probability outcome. For example, one would define "Backup Generator Fails" rather than "Backup Generator in Service".

2.2 Basic Events

A basic event is represented by a binary node in DPL. It is called a binary node because the event either occurs or it does not, i.e., it is either true or false. Binary nodes are flattened green ovals in DPL Fault Tree. See Figure 2-1.

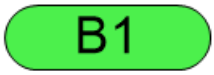

Figure 2-1. Binary Node

In a fault tree, a predecessor to an event is a lower-level event that feeds into the event. A basic event has no predecessors. I.e., the probability of its occurrences is assessed directly. It is not the combination of lower-level events.

2.3 Gates

Gates combine together the probabilities of the events that feed them. The events that feed a gate are referred to as its predecessors. All predecessors to a gate must be events. The gate is the successor event (or successor) to all of its predecessors. A gate is also an event and the output of the gate is the probability that it occurs.

Gates may have an unlimited number of predecessors.

2.3.1 AND Gates

AND Gates are red half ovals with a black dot in the middle in DPL Fault Tree. See Figure 2-2.

Figure 2-2. AND Gate

An AND gate is true if all of its predecessors are true (similar to the logical operator). The probability that an AND gate is true is the product of the probabilities of its predecessors.

2.3.2 OR Gates

OR Gates are yellow shield-shaped nodes with a plus sign in the middle in DPL Fault Tree. See Figure 2-3.

Figure 2-3. OR Gate

An OR gate is true if any of its predecessors are true (similar to the logical operator). The probability that an OR gate is true is one minus the probability that none of its predecessors are true. The probability that none of its predecessors is true equals the probability all its predecessors are false which is the product of each of its predecessors being false.

2.4 Connections

Connections are black lines that indicate which nodes are predecessors and successors in the fault tree. Figure 2-4 shows an AND gate with two connections into it, i.e., the gate has two predecessors. A node is a predecessor if a connection comes out of the top of the node; a node is a successor if a connection goes into the bottom of the node.

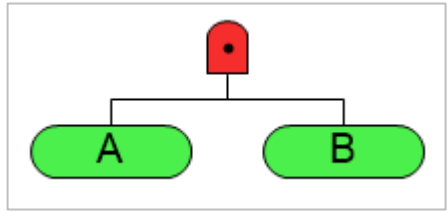

Figure 2-4. AND Gate with Connections

The AND gate in Figure 2-4 is true of both event A and B are true. The probability of the AND gate being true is the probability A is true times the probability B is true. A and B have no predecessors and are basic events.

2.4.1 NOT gates

NOT gates behave like the logical not operator. A NOT gate is represented by a blue triangle on the connection as shown in Figure 2-5. A NOT gate is true if the predecessor is false. I.e., when calculating the probability that a

successor to an event connected with a NOT gate is true, the probability that the predecessor is false is used which is one minus the probability of the predecessor.

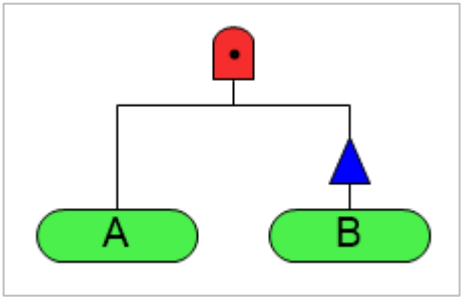

Figure 2-5. AND Gate with a NOT gate

The AND gate in Figure 2-5 is true of event A is true and B is false. The probability of the AND gate being true is the probability A is true times one minus the probability B is true (i.e., the probability B is false).

3. Building a Fault Tree in DPL

The next few chapters of this guide focus on building and analyzing a simple fault tree starting from a blank Workspace that represents a water shortage problem. You'll first assess the likelihood of a local water shortage and will later update the fault tree to incorporate the likelihood of disruptions to imported water sources as well.

3.1 Creating a Named OR Gate

⇨ Open DPL Fault Tree.

DPL Fault Tree will load with the Workspace Window on the left and an empty Fault Tree Window on the right-hand side of the screen. Further, the Fault Tree tab is active as show in Figure 3-1.

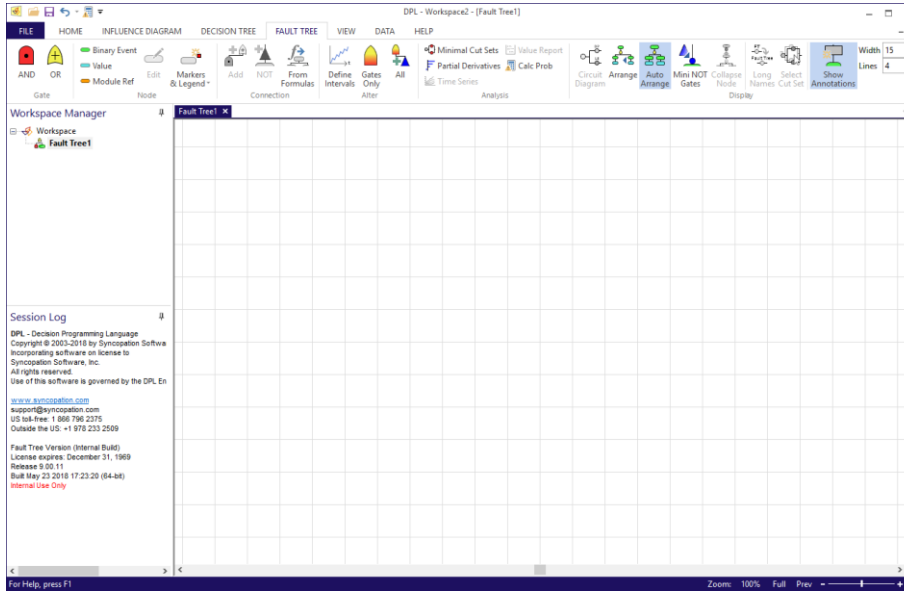

Figure 3-1. Blank DPL Fault Tree Workspace

A local water shortage will occur if there is a local rainfall shortage **or** if the local reservoir is contaminated due to a natural disaster. If either one of

these two events occur there will be a shortage. Consequently, the first element you'll add to the fault tree is an OR gate.

⇨ Click Fault Tree | Gate | OR.

Your cursor will move down to the Fault Tree Window and will have a semi-transparent OR Gate beneath it.

⇨ Single-click to place it near the top, center of the Fault Tree Window.

You can provide a long name, short name and an annotation for a gate via the Fault Tree Node Definition dialog. Defining any of these names for a gate is optional in most scenarios. For our purposes, you will provide a short name for the OR gate.

⇨ Double-click on the OR gate to edit it.

The Fault Tree Node Definition dialog opens for the gate. The General tab is available for providing long and short names, an annotation, and comments for the gate as shown in Figure 3-2. You'll provide just a short name.

⇨ Enter a short name of "LOCAL" as shown in Figure 3-2 and click OK to close the dialog.

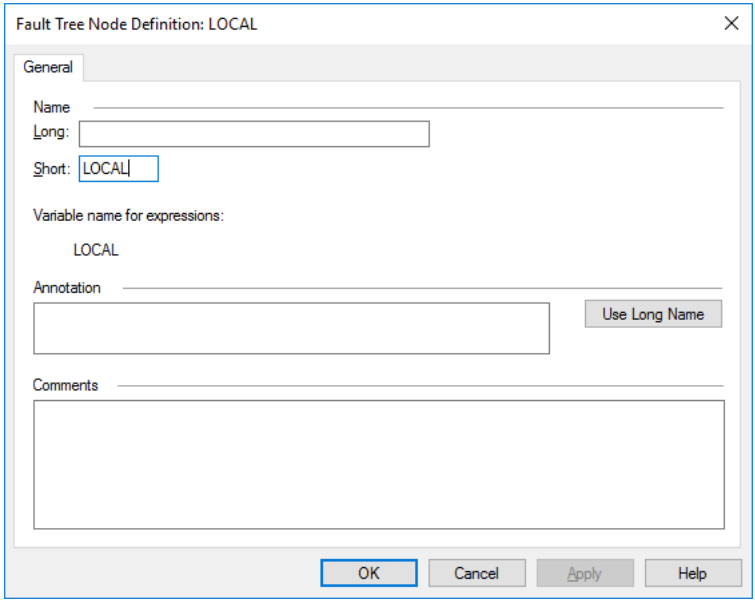

Figure 3-2. General Tab of Fault Tree Node Definition Dialog for OR Gate

The gate and node connected above it are colored magenta, indicating that they are selected.

⇨ Click in whitespace to de-select the OR gate.

If you provide a short name for a gate, the short name will be displayed as a binary node connected directly above the gate as shown in Figure 3-3. We'll say more about naming conventions and the fields on the General tab of the Node Definition dialog in Section 3.2.1 below. You may notice that when you select the named gate, the connection and the binary node above it are selected as well.

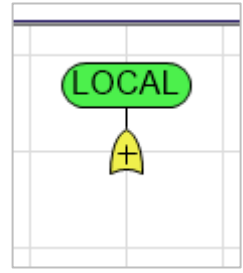

Figure 3-3. OR Gate with a Short Name of LOCAL

3.2 Adding and Connecting Binary Nodes

Next you will define and connect two inputs to the OR gate in the form of binary events: one for Local Rainfall Shortage and one for Local Reservoir Contamination.

⇨ Select Fault Tree | Node | Binary Event.

Your cursor will move down to the Fault Tree Window and will have a semi-transparent Binary node beneath it.

⇨ Click in white space beneath the OR gate to place the node.

3.2.1 Naming Conventions for Nodes

Once placed the Fault Tree Node Definition dialog will open for the binary event with the General tab active as show in Figure 3-4. By default, the node has been provided a long name of Binary 1 and a short name of B1.

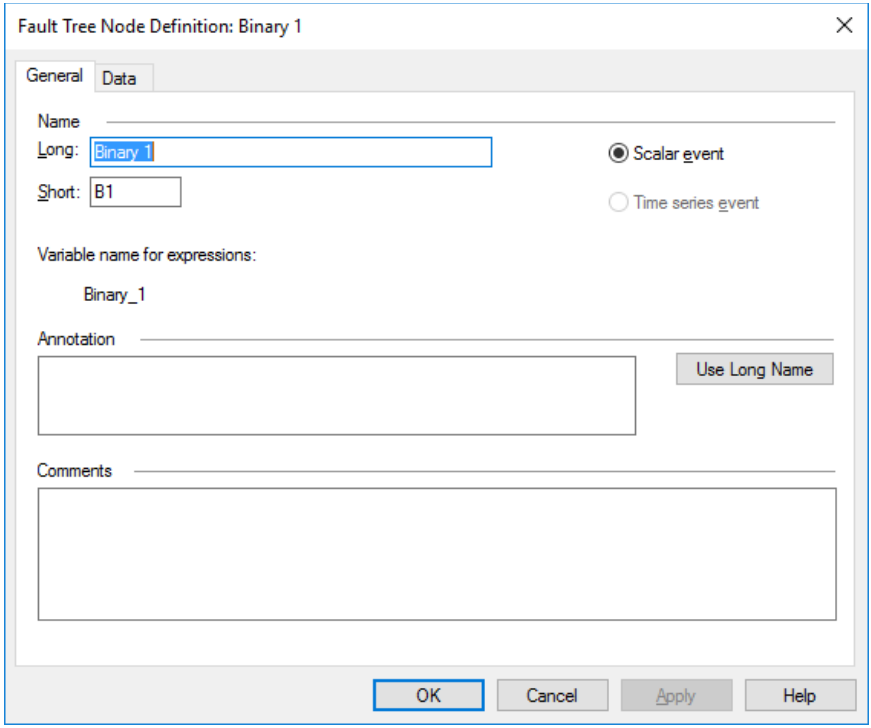

Figure 3-4. General Tab of the Fault Tree Node Definition dialog for a Binary Node

Keep in mind that not every field in the Fault Tree Node Definition dialog requires an input. A short name is required for all binary and value nodes and its usually good practice to supply a long name as well. Beyond that the data requirements vary depending on how the node is being used in the Fault Tree. Read on to see how each of these fields is used for nodes within the fault tree.

Long Name

The first field listed within the General tab is to enter a long name for the node. This allows for a name that is more descriptive than the short name for use in results, diagrams, and dialogs involving the node. DPL creates a variable name for the node based on the long name provided to use internally. Note that providing a long name for a node is optional. DPL will use the short name in lieu of a long name if one isn't defined.

Short Name

A short name is required for all binary and value nodes. A short name is required for a gate node if it is the top event in a fault tree or it will be used as a reference node. For binary and value nodes, it is displayed within the node in the Fault Tree Window. For gates, it is displayed above the gate as a binary node. Consequently, the short name is restricted to fit inside the node; depending on the font approximately 6-8 characters can be displayed within a node. Nodes within the fault tree are fixed in size to keep large trees a manageable size. DPL will initially generate a short name based on the long name entered. You can accept this default or provide a name of your choosing.

Annotations

Text entered here appears in the fault tree in a gray box above the node. This can be used to further describe the node. It often makes sense to have the annotation be the same as the long name. You can click the "Use Long Name" button to the right of the annotation text box to copy the long name to the annotation field.

Annotations are useful for making the tree more readable, but it also increases the viewing size of the tree. You can toggle the display of all annotation boxes on/off via the Fault Tree | Display | Show Annotations button. Further, you can specify the width and height of the annotation boxes in the fault tree by changing the numbers within the Width and Lines text boxes in Fault Tree | Display group as highlighted in the Figure 3-5.

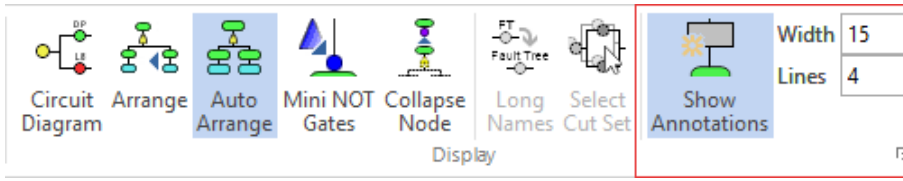

Figure 3-5. Annotation Controls within Fault Tree | Display

Comments

Lastly, there is a box to enter comments for the node. Text entered here will not appear in the fault tree display. You can view comments for a node if you hover your mouse cursor over the node and "Show Tips" is turned on within the View | Tips group. The tip box will also list the long name, probability, and cost data if supplied for the node.

Notice that there is also a Data tab within the dialog. The Data tab is where you enter probability and cost data for the node, if appropriate. Costs will not be incorporated in the fault tree and because this is not a basic event, you do not specify a probability of occurrence for it. DPL will assign the appropriate probability according to the connections placed below it. The information contained in the Data tab of the Fault Tree Node Definition dialog will be discussed in more detail in Section 3.2.3.

⇨ On the General tab, enter "Local Rainfall Shortage" for the long name.

⇨ Notice that DPL has generated a short name of "LRS" from the long name. You will accept this default.

⇨ Click the "Use Long Name" button to copy the long name to the annotation field. The dialog should look like Figure 3-6.

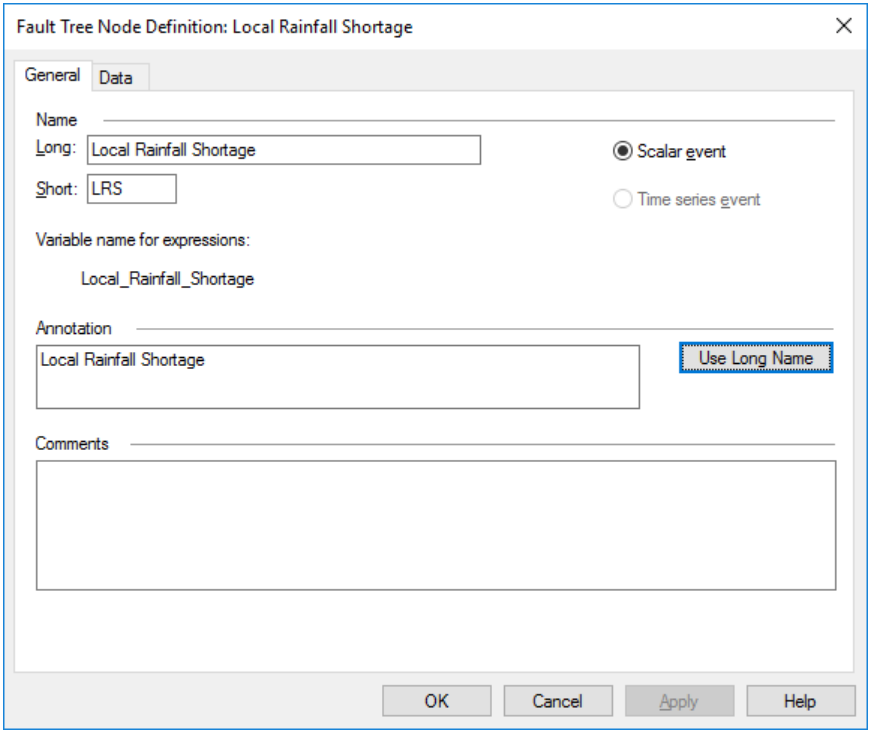

Figure 3-6. Filled in General Tab of Fault Tree Node Definition Dialog for Local Rainfall Shortage Node

⇨ Click OK to close the dialog.

⇨ Press Esc or click in white space to de-select the newly created node. Your Fault Tree should look like Figure 3-7

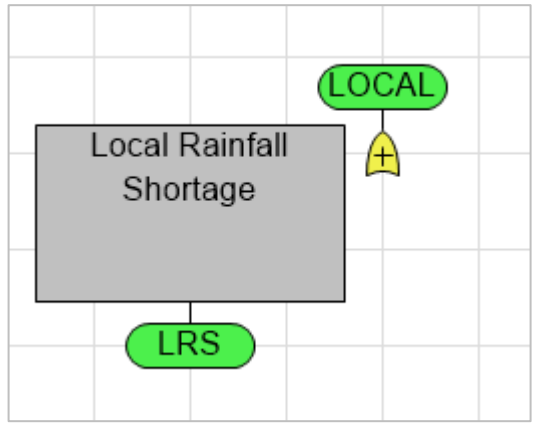

Figure 3-7. Local Rainfall Shortage Binary Node with Annotation Box

3.2.2 Two Methods for Connecting Binary Nodes to a Gate

You'll now connect this Binary event to the OR gate via the ribbon command.

⇨ Select Fault Tree | Connection | Add.

Your cursor will move down to the Fault Tree window and will turn into a choose predecessor cursor.

⇨ Click on the Local Rainfall Shortage node. This is the predecessor.

Your cursor will change to a choose successor cursor and will be anchored to the Local Rainfall Shortage node.

⇨ Click on the OR gate named LOCAL. This is the successor.

DPL will establish a connection between the two objects. Further, the predecessor node is centered beneath the OR gate. DPL did this for you automatically because the Auto Arrange setting is on by default within the Fault Tree | Display group. With Auto Arrange on, predecessors are evenly spaced left to right below their successor once a connection is made. You can select to have the fault tree arranged automatically as you build the Fault Tree (e.g., Auto Arrange on) or arranged only at a time of your choosing (e.g., Auto Arrange is turned off and the Arrange command is used). You'll continue with Auto Arrange turned on.

You'll now add the node for Local Reservoir Contamination.

- ⇨ Select Fault Tree | Node | Binary Event and place it to the left of the Local Rainfall Shortage node.
- ⇨ Enter a long name of "Local Reservoir Contamination", accept the default short name of "LRC" and copy the long name to the annotation field.
- ⇨ Click OK.

This time you're going to use the keyboard short-cut method to connect this new binary node to the OR gate.

- ⇨ While pressing the Shift key, single-click on the Local Reservoir Contamination node, the predecessor.
- ⇨ Click on the OR gate, the successor.

DPL establishes a connection between the two objects. Again, the two binary nodes were automatically arranged below the gate they are connected to as shown in Figure 3-8.

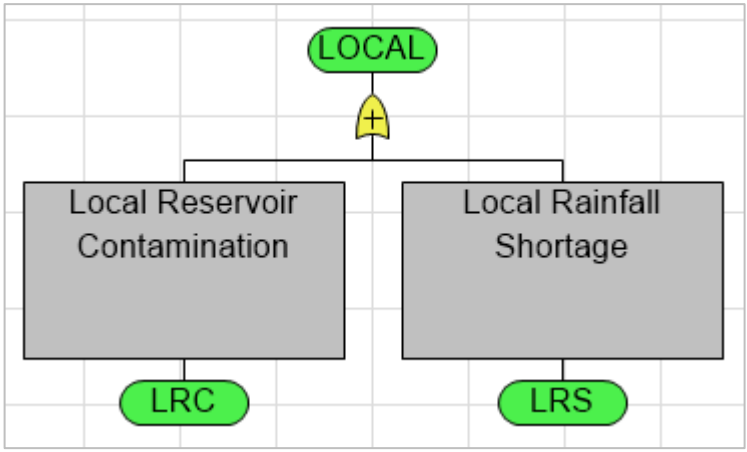

Figure 3-8. Named OR gate with Two Binary Event Predecessors

If the Local Reservoir Contamination and Local Rainfall Shortage nodes appear in a different order from left to right than what is shown in Figure 3-8, it is easy to switch the order of the nodes if Auto Arrange is on. E.g., if Local Reservoir Contamination appears to the right of Local Rainfall Shortage, simply drag the Local Reservoir Contamination node to the left side of Local Rainfall Shortage and release. The nodes will be reordered beneath the gate.

Throughout these tutorials the nodes in your fault tree that are inputs to a common gate may appear in a different order from the figures shown in this guide. While this isn't going to affect the results of the fault tree, you may want to reorder the nodes as necessary to match your tree to the figures in the guide to avoid any confusion.

3.2.3 Data Tab of the Node Definition Dialog

You now have a binary event in the fault tree that is concrete enough that it's probability of occurrence can be assessed directly. In other words, there are no predecessor events for this node. You've acquired information from the local weather service regarding the likelihood of a local rainfall shortage. You'll enter this probability of occurrence now within the Data tab of its Fault Tree Node Definition dialog.

- ⇨ Double-click the Local Rainfall Shortage node to edit it. Note that when editing an already existing binary node, the Data tab is active by default.
- ⇨ On the Data tab, enter a probability of "0.22" in the Probability column of the grid. See Figure 3-9.

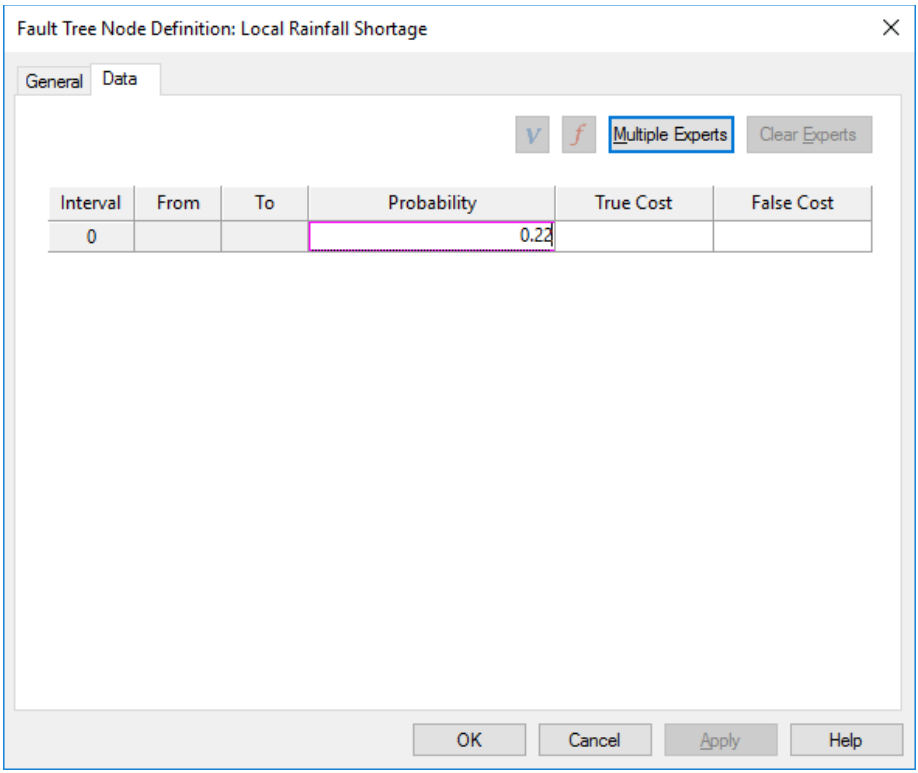

Figure 3-9. Data tab for Local Rainfall Shortage with Probability Entered

⇨ Click OK to close the dialog.

There is slightly more than a 20% chance that there will be a shortage of rainfall this year. The Local Reservoir Contamination event cannot be directly assessed, so you'll break this event down into components. You refer to events that have predecessors as "derived events".

3.2.4 Automatically Connecting Objects with Auto Arrange Enabled

The local reservoir will contribute to the probability of a local water shortage if it's contaminated due to a natural disaster. You obtained information on the likelihood of various natural disasters occurring in your area. Potential threats in your area include earthquake, fire, and flooding. The probabilities of occurrence for each are provided in the Table 3-1.

But a natural disaster does not always result in contamination of the reservoir. Consequently, the natural disaster must occur **and** contaminates

must exist for the reservoir to be contaminated. To define these inputs, you're first going to connect an AND gate to the Local Reservoir Contamination binary node.

In previous sections you dropped nodes in whitespace and then employed both the command ribbon and a keyboard shortcut to connect it to a node above. DPL provides one additional means for creating connections between a new node and an existing node when Auto Arrange is enabled. You'll try it now.

- ⇨ Select Fault Tree | Gate | AND.
- ⇨ Place it directly on top of the Local Reservoir Contamination binary node.

DPL automatically centers the AND gate below the Local Reservoir Contamination binary node and makes a connection between them. Now you'll create a binary node that represents a natural disaster occurring and connect the binary event to the AND gate using this same method.

- ⇨ Select Fault Tree | Node | Binary Event and place it directly on top of the AND gate you just created.
- ⇨ Provide a long name of "Natural Disaster" and accept the default short name of "ND". Leave the annotation field blank.
- ⇨ Click OK to close the dialog.

You are not going to enter a probability of occurrence for this binary event, instead you'll break this event down further into the three natural disaster events in Table 3-1.

Short name	Probability of Occurrence
Quake	1.8e-7
Fire	3.7e-4
Flood	2.5e-4

Table 3-1. Probability of Natural Disasters Occurring

- ⇨ Drop an OR gate onto the Natural Disaster binary node.
- ⇨ Create a binary node for each of the natural disasters and connect them to the OR gate by dropping them on the gate.

⇨ Provide a short name and probability of occurrence using the information in Table 3-1.

Your fault tree should look like Figure 3-10.

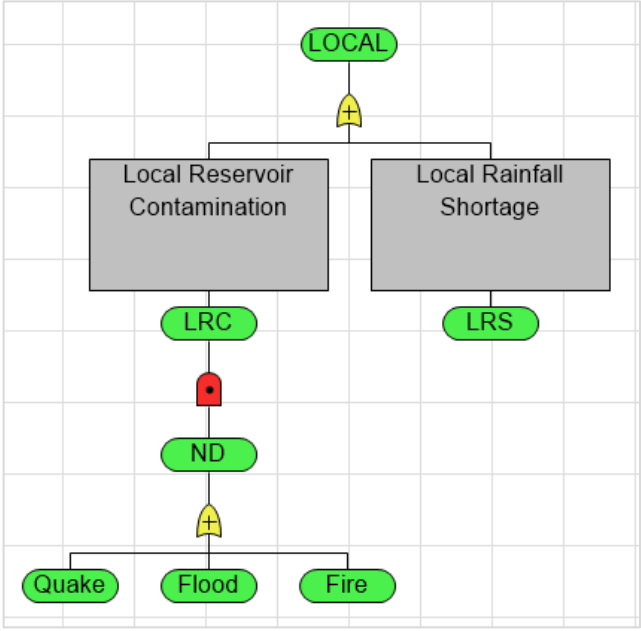

Figure 3-10. LOCAL Subtree with Natural Disaster Nodes Defined

You need to define one more event to complete the fault tree and calculate the probability of a local water shortage. You need a binary event that represents how likely it is that contaminates exist that could contaminate the reservoir given its location and the topography of the area.

⇨ Select Fault Tree | Node | Binary Event and place it on top of the AND gate below the Local Reservoir Contamination binary node.

⇨ Provide a long name of "Local Water Contamination", accept the default short name, and enter a probability on the Data tab of "0.33".

⇨ Click OK to close the dialog. See Figure 3-11.

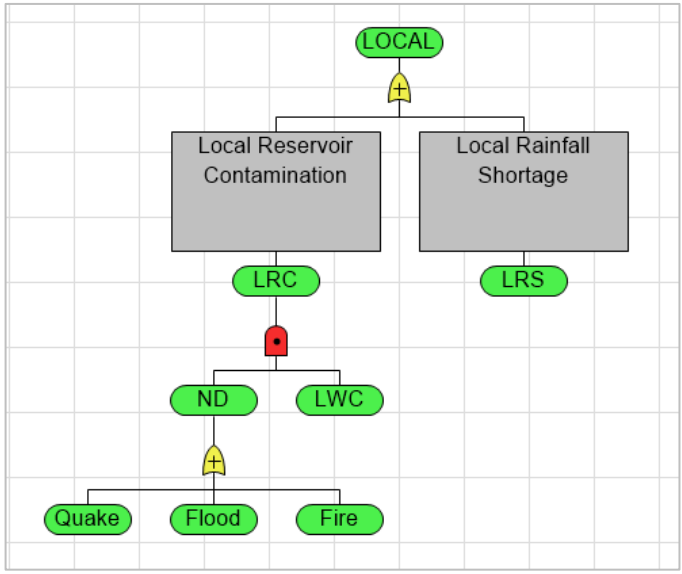

Figure 3-11. Complete Local Water Shortage Fault Tree

3.3 Calculating Probabilities

The Local Water Shortage fault tree is now complete as all the basic events, e.g., events that don't have any predecessors, have been defined and supplied with probabilities. You can now calculate the probability of occurrence for the top event or any event in the fault tree.

When you run this analysis, DPL will first compile the fault tree to check it for errors. If it finds any, a message will be shown and the node causing the error will be selected. If there are no errors, the calculation is completed and a dialog is displayed with the value. You'll run this analysis now.

⇨ Hit ESC or click white space to make sure nothing is selected within the Fault Tree.

⇨ Select Fault Tree | Analysis | Calc Prob.

If you don't have any nodes selected in the fault tree, DPL will calculate the probability of occurrence for the top event in the tree. Once the analysis is complete the Calculation Complete dialog will appear and display the probability as shown in Figure 3-12. The number of decimals may differ. You'll see how to change those below.

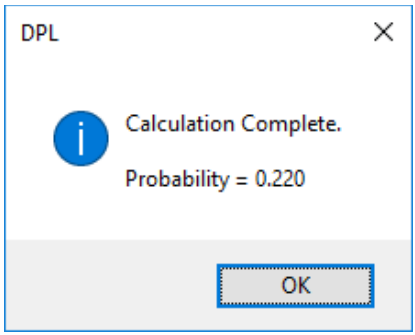

Figure 3-12. Calculation Complete Dialog for Top Event (LOCAL)

This output is the probability that the top event is TRUE. Let's say you'd also like to know how likely it is that the local reservoir is contaminated. To calculate a probability of occurrence for an event other than the top node, you must select it and then run the analysis.

⇨ Select the Local Reservoir Contamination node.

⇨ Select Fault Tree | Analysis | Calc Prob.

The probability of occurrence is displayed as 0.000, but it isn't actually zero, it's just so small that it's being rounded to zero. Consequently, you'll need to increase the number decimal places displayed for outputs. To do so:

⇨ Click OK to close the dialog.

⇨ Select File | Options | Outputs.

⇨ Under the General number formatting heading, change number of decimal places to "4" as shown in Figure 3-13.

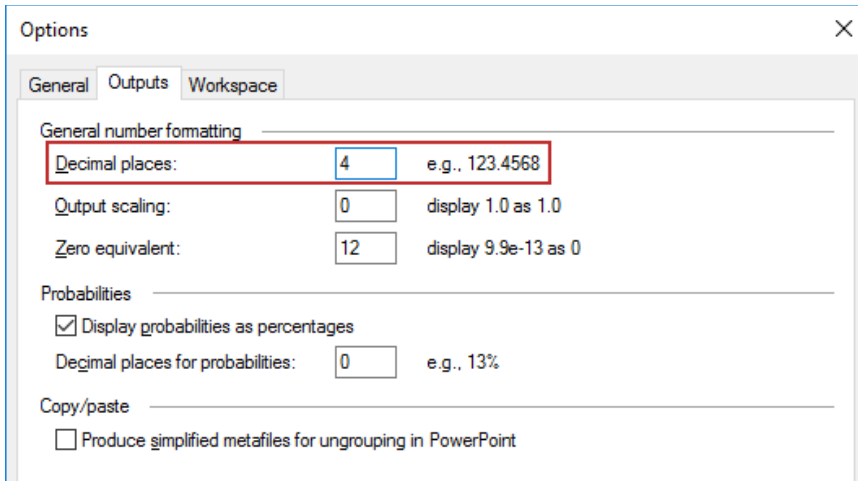

Figure 3-13. Increasing Number of Decimal Places for Outputs in File | Options | Outputs

⇨ Calculate probability for the Local Reservoir Contamination event again.

Now you can see that the probability of occurrence is not zero but 0.0002 as shown in Figure 3-14.

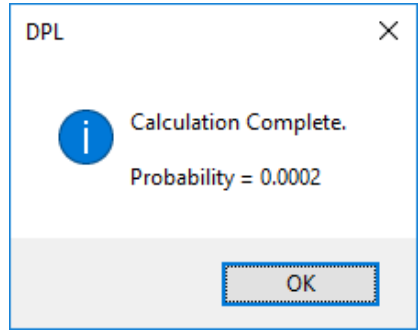

Figure 3-14. Calculation Complete Dialog for Local Reservoir Contamination with 4 Decimal Places

3.4 Events Fed by Value Nodes

You've assessed the likelihood of a localized water shortage but haven't taken into consideration the fact that there are two outside water sources

(referred to in this tutorial as "Source 1" and "Source 2") that could potentially augment the local water supply to avert a water shortage.

So, a water shortage will only occur if it involves the local water supply **and** all imported water sources. You will add a new, named AND gate to the top of the fault tree.

- ⇨ Select Fault Tree | Gate | AND and place it near the top of the fault tree above LOCAL.
- ⇨ Edit the gate and provide a long name of Water Shortage, accept the default short name, and copy the long name to the annotation.
- ⇨ Connect the LOCAL binary node to the new AND gate. Your fault tree should look like Figure 3-15.

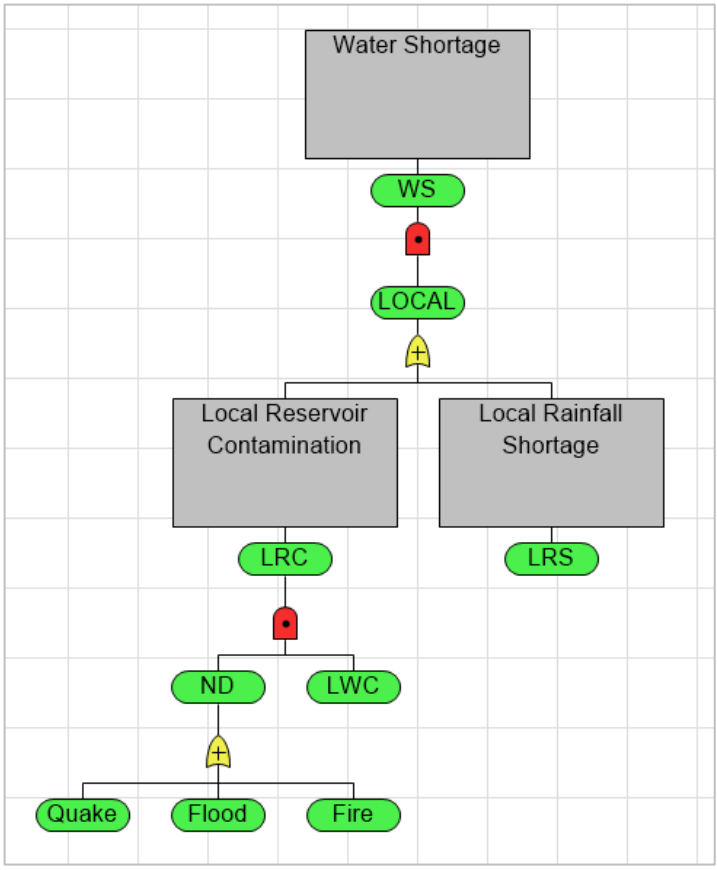

Figure 3-15. Fault Tree with Water Shortage as the Top Event

You'll build out the subtree that breaks down a water shortage/disruption at Source 1.

- ⇨ Drop a new OR gate onto the AND gate named Water Shortage.
- ⇨ Edit the newly created OR gate and supply a long name of "Imported Water Shortage 1", accept the default short name, and copy the long name to the annotation field.

An incoming water shortage from Source 1 will occur if there is shortage at the source **or** if there is a problem with the aqueduct that brings the water in from Source 1. Furthermore, like the local subtree, there will be a shortage at source 1 if there is a rainfall shortage in the area **or** if the reservoir is contaminated. The aqueduct at source 1 will fail if a natural disaster occurs there **or** if the aqueduct suffers from age damage. Engineers estimate the probability of age damage occurring to be 0.18. Consequently, you will add and name three new OR gates to represent this fault.

- ⇨ Drop a new OR gate onto the OR gate named IWS1.
- ⇨ Edit the newly created gate and supply a long name of "Aqueduct Fault 1" and accept the default short name. Leave the annotation field blank.
- ⇨ Drop another new OR gate onto the OR gate named IWS1.
- ⇨ Edit the newly created gate and supply a long name of "Source 1 Shortage" and accept the default short name. Leave the annotation field blank.

Your fault tree should look like Figure 3-16.

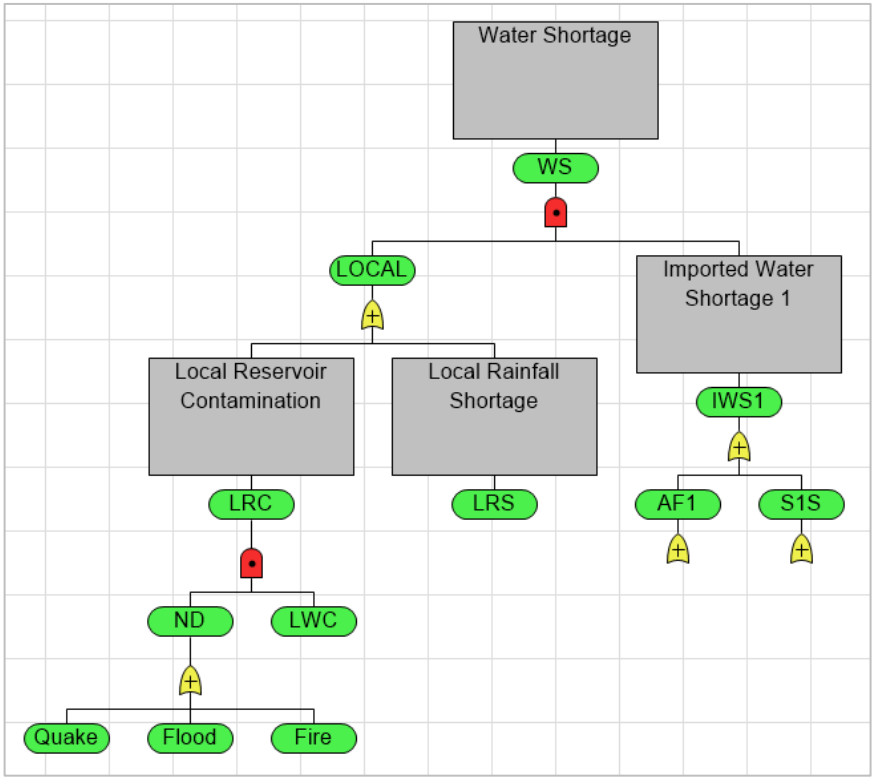

Figure 3-16. Fault Tree with Source 1 Water Shortage Subtree Partially Defined

You'll break down the aqueduct failure into its component faults first. Its predecessors are a natural disaster **or** age damage. Note that the natural disaster at Source 1 should be treated as independent from the natural disaster event that is an input to the local water shortage. In other words, it is possible to have a natural disaster occur at Source 1 but not in your area.

⇨ Drop a new Binary event onto the OR gate named AF1.

⇨ Provide a long name of "Natural Disaster 1", accept the default short name, and copy the long name to the annotation.

While they are independent events you assume that, due to their proximity, natural disasters in all locations (Local, Source 1, Source 2) have an equal probability of occurrence. Therefore, you will use a value node to capture the probability of the local Natural Disaster binary event and then will reference it in this and other parts of the fault tree.

⇨ Delete the connection between the ND binary node and the OR gate beneath it within the local subtree.

⇨ Select Fault Tree | Node | Value and place it whitespace near this same spot.

The Fault Tree Node Definition dialog opens for the Value node. You'll see that the dialog is very similar to those for binary nodes.

⇨ Supply only a short name of "NDprob".

3.4.1 Rules for Value Nodes and Connections

Note that a value node is not an event itself and, therefore, cannot be used as an input to a gate. Only events can feed into gates. Value nodes can contain numbers and expressions. Expressions can use an unlimited number of variable names but all value nodes that are referenced must be connected to the node.

If the value is between zero and one, it can be used as the probability value of a binary event, which is what you're doing now. Value nodes can also be used to provide true and/or false cost data for a binary event. For more on true and false costs, see Chapter 6.

⇨ Select the Data tab for the value node to review it.

Notice that there is only one field for entering data for the value node, unlike a binary which has fields to enter a probability and true and false costs.

It should be further noted that, like gates, there are no limits on the number of predecessors a value node can have.

⇨ Leave the Data tab blank. Click OK to close the dialog.

⇨ Connect the OR gate above Quake, Flood, and Fire to the NDprob value node.

⇨ Connect the NDprob value node to the ND binary event.

See Figure 3-17.

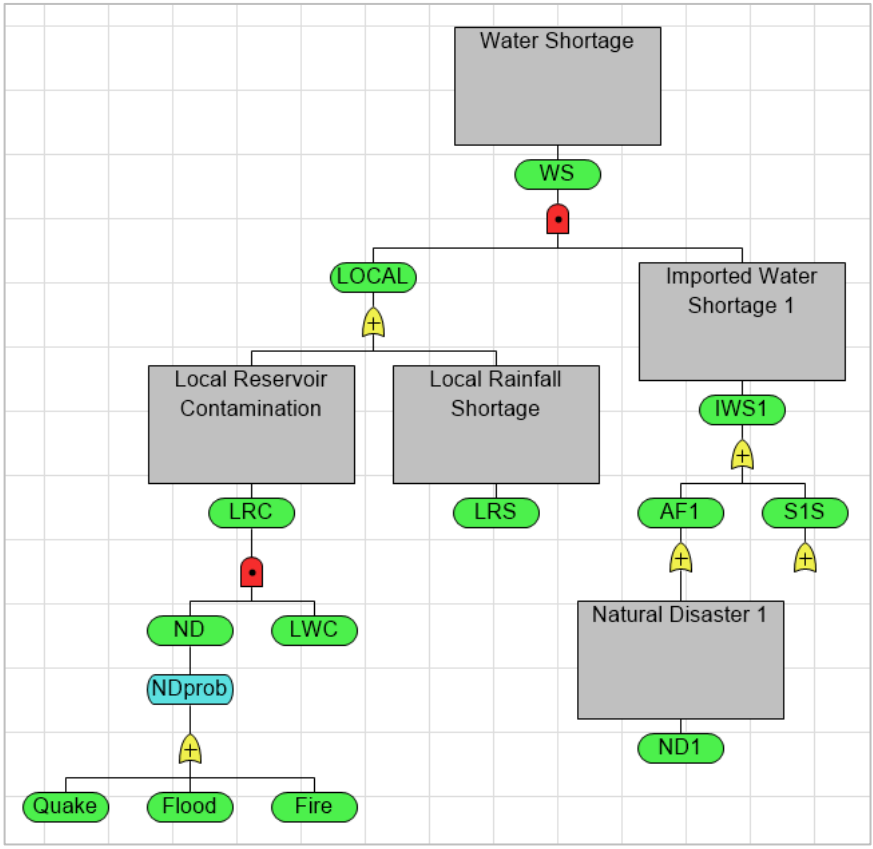

Figure 3-17. Fault Tree with Probability Value Node NDprob Added

The calculated probability of the OR gate now provides the probability value for NDprob. NDprob initializes the value of the local natural disaster event (ND) which feeds the Local Reservoir Contaminated gate. As mentioned previously, only events can be predecessors to gates which is why we need the ND event as well.

3.4.2 Probability Value Reference Node

As mentioned above NDprob is the probability of a natural disaster which is derived from the binary events below it. You can now reference this value node for the probability of the Source 1 natural disaster event.

⇨ While pressing the Shift key click on the NDprob value node and then the ND1 binary node.

DPL automatically makes a reference node to the value node NDprob and names it "*NDprob". See Figure 3-18.

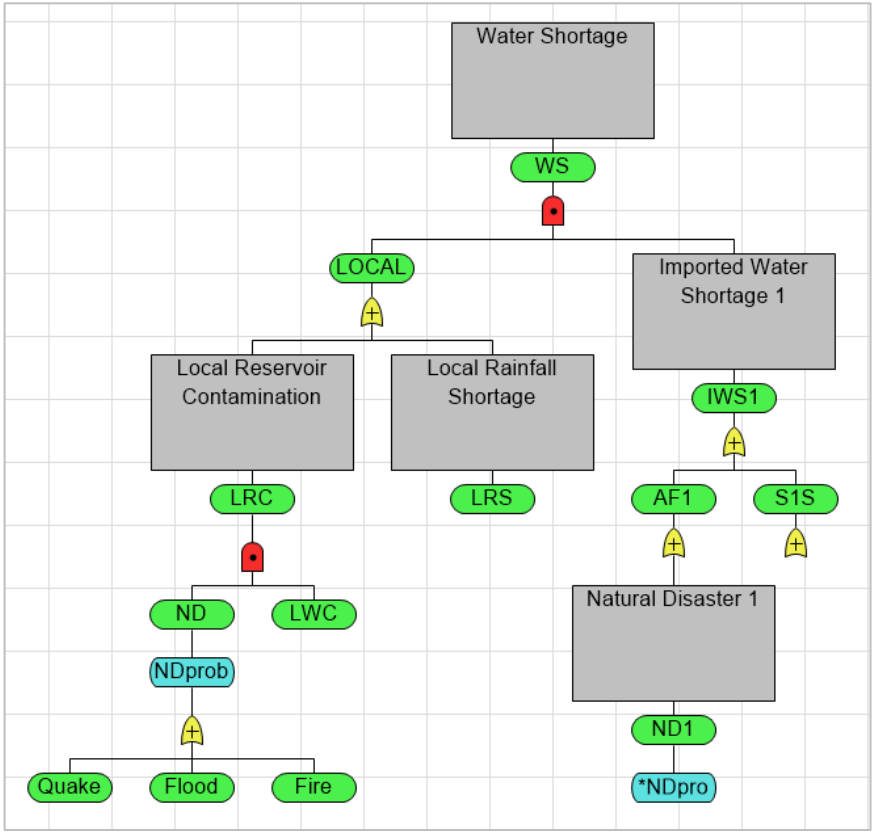

Figure 3-18. Fault Tree with Probability Value Node Reference

The reference node is a value node which means that the same probability of occurrence is being used for both ND and ND1. However, the events ND and ND1 are separate events and may or may not occur independently of each other. They are both initialized with NDProb and hence each occur with the same likelihood. See Section 3.4.3 for information on reference nodes that refer to an event. Using a reference node avoids needing to enter the same data in multiple places and in this instance avoids repeating the structure of the events and gate needed to derive the probability.

You will now complete the logic for aqueduct fault at source 1.

⇨ Drop a new Binary event onto the OR gate named AF1.

- ⇨ Provide a long name of "Age Damage 1", accept the default short name, and copy the long name to the annotation field.
- ⇨ Click OK to close the dialog.
- ⇨ Create a new Value node and drop it onto the Age Damage 1 binary node.
- ⇨ Delete the default long name and provide only a short name of "ADprob".
- ⇨ On the Data tab enter a probability of "0.18". Click OK to close the dialog.

Your fault tree should look like Figure 3-19.

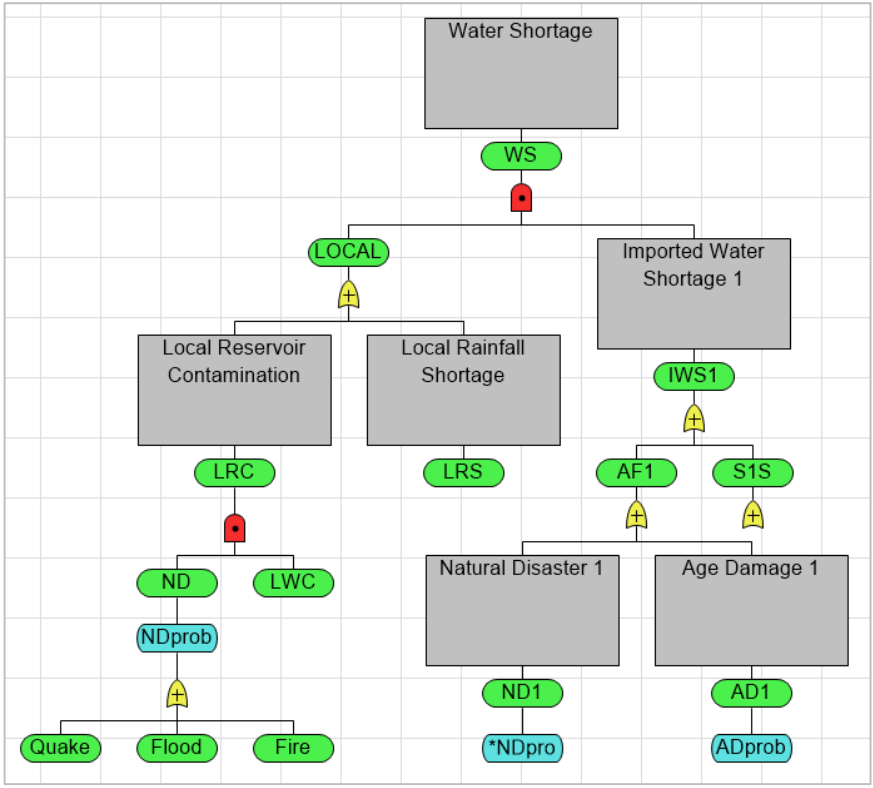

Figure 3-19. Fault Tree with ADprob Probability Value Node Defined

You'll continue building out the fault tree by breaking down the Source 1 Shortage (S1S) event. Like the local side, a shortage will occur at Source 1 if

there is a lack of rainfall **or** if the reservoir is contaminated due to a natural disaster. You'll define the rainfall shortage first.

- ⇨ Drop a new Binary event onto the OR gate named S1S.
- ⇨ Provide a long name of "Rainfall Shortage 1", accept the default short name, and copy the long name to the annotation field.
- ⇨ This is a basic event. On the Data tab enter a probability of occurrence of "0.15". Click OK to close the dialog.
- ⇨ Drop a new AND gate onto the OR gate named S1S.
- ⇨ Edit the newly created AND gate and supply a long name of "Reservoir Contamination 1", accept the default short name, and copy the long name to the annotation field.

The fault tree should now look like Figure 3-20.

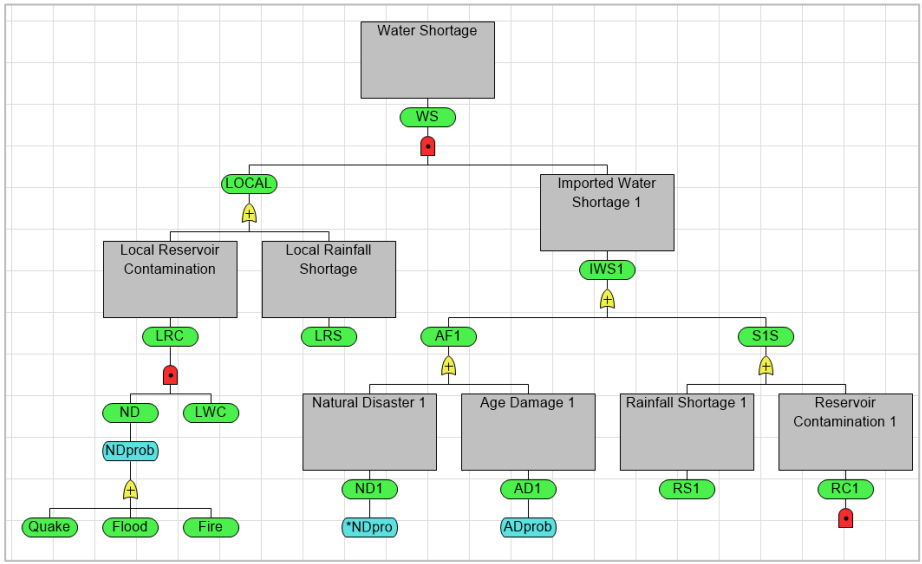

Figure 3-20. Fault Tree with Source 1 Water Shortage Partially Defined

3.4.3 Binary Reference Nodes

You'll experience a contamination if a natural disaster occurs at Source 1 **and** the reservoir is contaminated. The probability of contamination depends upon the topography and location of the reservoir at each location. Therefore, you'll define independent contamination events with a different probability of occurrence for each location.

Recall that you've already added an event to the fault tree that represents a natural disaster occurring at Source 1. That same event contributes to the probability that the reservoir at Source 1 is contaminated. So rather than creating another, independent natural disaster event you're going to create a second instance of the ND1 event via a reference node.

⇨ While pressing the Shift key click on the ND1 binary node and then click on the AND gate RC1.

DPL creates a duplicate ND1 event and connects it to the AND gate named RC1. The two nodes ND1 and *ND1 in this fault tree represent the same event.

⇨ Hover your mouse cursor over the *ND1 node.

A dotted line is drawn between the reference node and its source. If you were to hover the cursor over ND1, a line would be displayed to each of its references, which in this case is just *ND1. Keep in mind that a node can have multiple references to it, as you'll see later.

In this section, you created an event reference node which results in the two nodes being treated as the same event when DPL performs computations. In section 3.4.2, you created a value reference node, NDprob, for the probability of a natural disaster. The two events the value nodes are connected to (ND and ND1) are independent and are calculated as such. They simply share the same probability of occurrence. For a further discussion of this, see Chapter 4.

⇨ Drop a new Binary event onto the AND gate named RC1.

⇨ Provide a long name of "Water Contamination 1" and accept the default short name.

⇨ Enter a probability of "0.23" on the Data tab. Click OK to close the dialog.

Your fault tree should look like Figure 3-21.

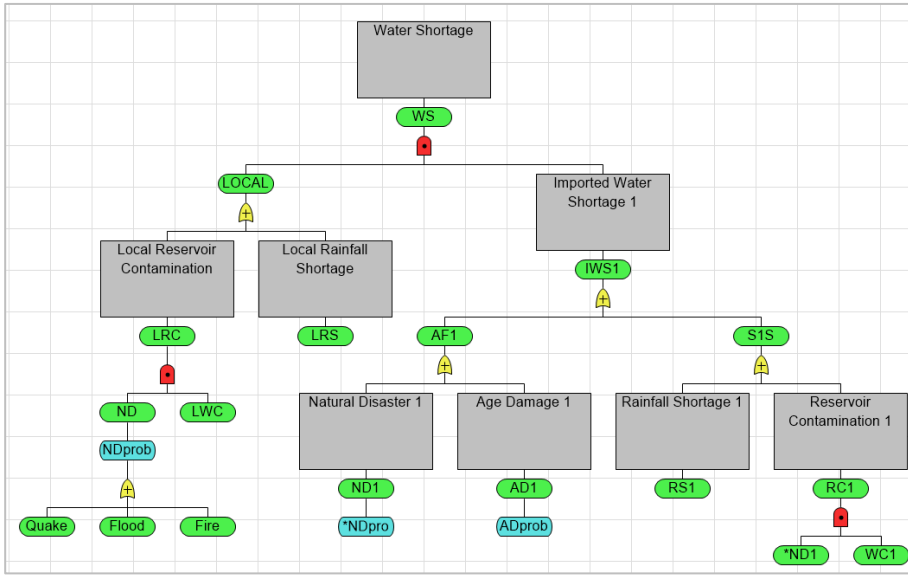

Figure 3-21. Fault Tree with Completed Local and Source 1 Subtrees

The fault tree is now complete as it stands. All the basic events have been defined and supplied with probabilities and all the nodes are connected in some fashion to the top node. Consequently, you can again calculate the probability of occurrence for any event in the fault tree.

⇨ Calculate the probability of occurrence for Water Shortage and Imported Water Shortage 1 (Figure 3-22).

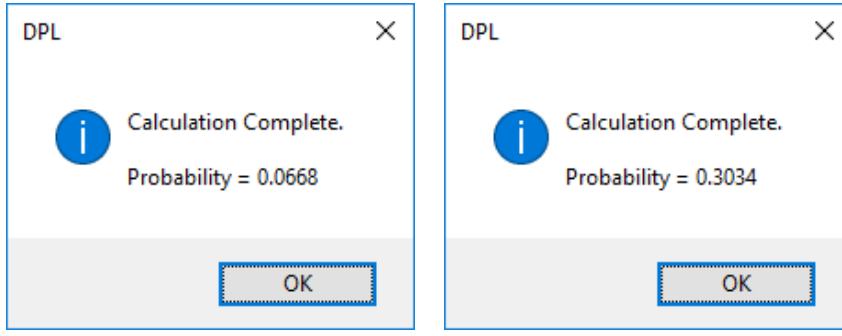

Figure 3-22. Calculation Complete Dialog for Water Shortage (left) and Imported Water Shortage (right)

3.5 Manipulating Fault Trees

3.5.1 Copying and Pasting Subtrees

You'll experience an imported water shortage if both Source 1 **and** Source 2 are disrupted by shortages, contamination, or a fault in the aqueduct. To reflect this, you'll add a named AND gate that's a predecessor to Water Shortage representing an imported water shortage.

- ⇨ Delete the connection between the IWS1 node and Water Shortage.
- ⇨ Drop an AND gate onto the AND gate named Water Shortage.
- ⇨ Edit the newly created AND gate and provide a short name only of "IMPRT". Click OK to close the dialog.
- ⇨ Connect the IWS1 node to the AND gate named IMPRT. Your fault tree should look like Figure 3-23.

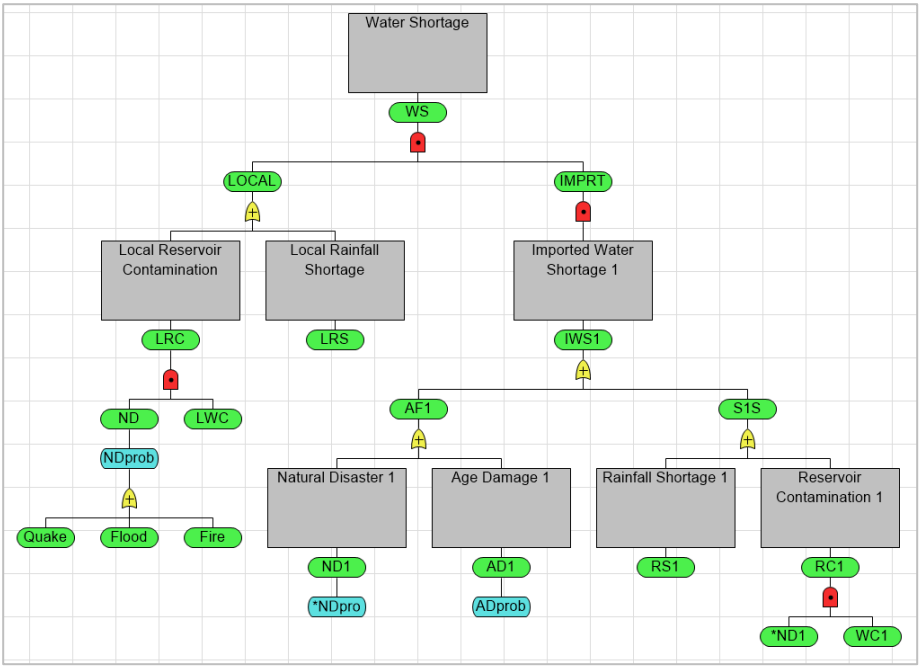

Figure 3-23. Fault Tree with AND Gate Named IMPRT Added

Instead of building out a nearly identical subtree one step at a time for Source 2, you're instead going to make a duplicate of the Imported Water

Shortage 1 subtree via copy/paste. You'll then simply need to rename the copied nodes within the subtree and connect it to the AND gate named IMPRT.

- ⇨ Press Shift and Ctrl keys and click on the IWS1 node. This will select the node and the entire subtree beneath it.
- ⇨ Press Ctrl+C to copy the subtree to the clipboard.
- ⇨ Press Ctrl+V to paste the subtree.

The copied subtree is placed slightly above and to the right of the original and is selected. The entire subtree structure and data has been copied. Only the node names are different in the copied subtree. The long names have been appended with "_copy" and the short with "_c". You'll now connect this subtree to the IMPRT gate.

- ⇨ Press ESC or click white space to deselect.
- ⇨ While pressing the Shift key select the IWS1_c node and then click on the AND gate named IMPRT.

You'll now remove the copy suffix and update all instances of the wording "Source 1" to "Source 2" in the copied subtree. Instruction will be provided for a single node, and then you can follow the same steps for the rest.

- ⇨ Double-click on the IWS1_c node to edit it.
- ⇨ Delete the short name as you want this to update based on the new long name you provide.
- ⇨ Change the long name to read "Imported Water Shortage 2", accept the default short name, and copy the long name to the annotation.
- ⇨ Rename the rest of the nodes in the copied subtree in the same manner. Delete the annotations for the rest of the nodes, as these aren't necessary since the subtree logic matches that of Source 1.

The aqueducts at source 1 and source 2 are of the same type and age. While the two events (AD1 And AD2) are independent (they may each occur or not) their probability of occurrence is the same. Therefore, a reference to a probability value node is used to initialize the AD2 event. You will do this now.

- ⇨ Delete the ADprob_c value node beneath the AD2 node and create a reference from the ADprob value in the Source 1 subtree to the AD2 binary event. See Figure 3-24.

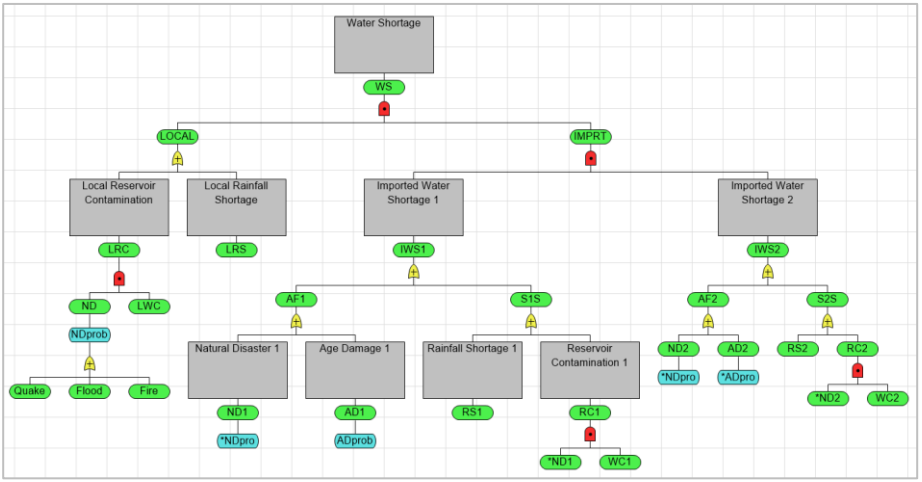

Figure 3-24. Fault Tree with Source 2 Subtree Names Updated

The probability of a rainfall shortage and contamination of the reservoir given a natural disaster are different at Source 2 so you'll also need to update these probabilities.

⇨ Update the probability for the RS2 node to be "0.22"

⇨ Update the probability for the WC2 node to be "0.19".

The full fault tree is now complete. You can again calculate the probability of occurrence for any event in the fault tree.

⇨ Calculate the updated probability of a water shortage.

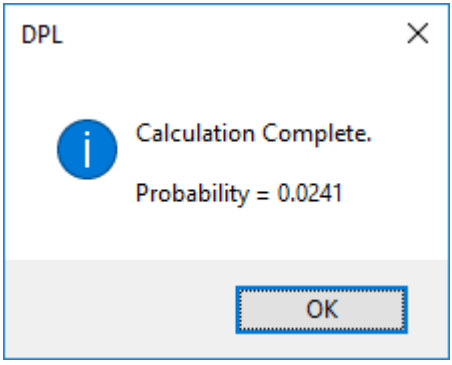

Figure 3-25. Calculation Complete Dialog of Water Shortage for Fully Complete Fault Tree

3.5.2 Display Options for Fault Trees

You may have noticed that it is becoming more difficult to easily view the entire fault tree on the screen. There are a couple of ways to remedy this. First, you'll modify the size of the annotation boxes in the fault tree display.

⇨ Within the Fault Tree | Display group, change the Width value from "15" to "10" and the Lines value from "5" to "3" as shown in Figure 3-26.

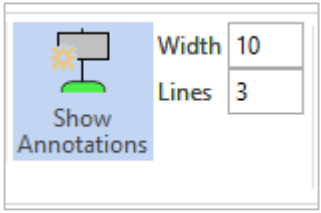

Figure 3-26. Updated Annotation Width and Lines Sizing Boxes

Another way to make a fault tree more compact is to collapse certain sections. Since the Source 2 subtree nearly matches that of Source 1, you might want to collapse the Source 2 subtree to make it easier to view the fault tree on the screen. You'll first select the node directly atop the subtree you want to collapse.

⇨ Select the Imported Water Shortage 2 node.

Notice within the Fault Tree | Display group that with this node (or any node) selected the Collapse Node toggle button is active on the command ribbon.

⇨ Click the Fault Tree | Display | Collapse Node toggle button to collapse the subtree beneath the selected node.

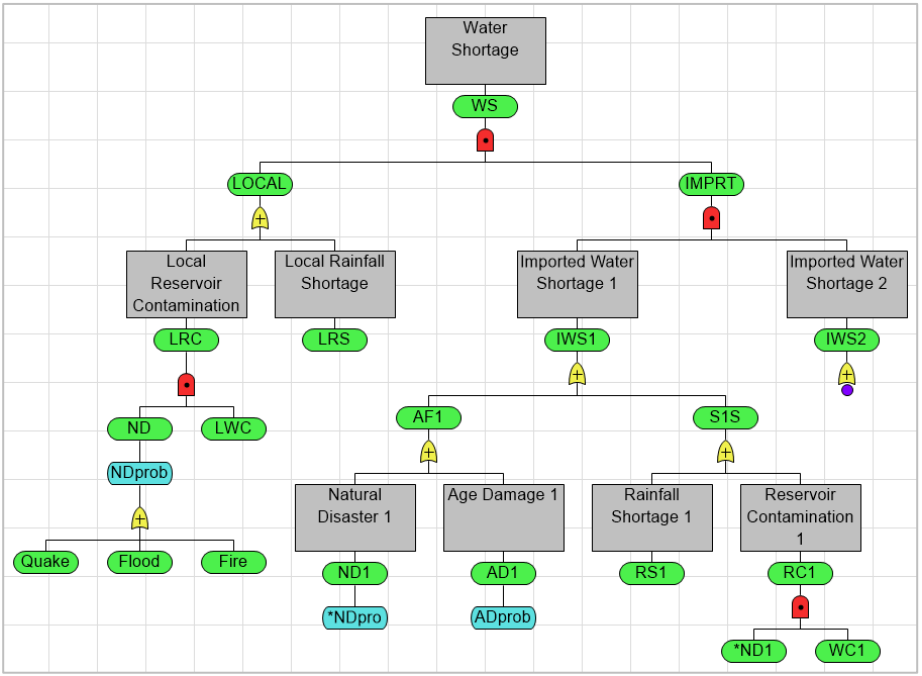

Figure 3-27. Fault Tree with Imported Water Source 2 Shortage Node Collapsed

All the predecessors beneath the IWS2 node are hidden from view and a small purple circle is drawn beneath the node to indicate the presence of a collapsed subtree as shown in Figure 3-27. Note that collapsing the subtree does not affect the behavior of the fault tree; it is for viewing purposes only.

If you prefer short-cuts, press and hold the Ctrl key and double-click on the node you'd like collapsed.

To expand the subtree, you would follow the same steps as above: select the collapsed node and toggle the Fault Tree | Display | Collapse Node off or Ctrl+double-click the collapsed node.

⇨ Save the Workspace file with the fault tree for use in subsequent chapter tutorials.

4. Circuit Diagrams

A circuit diagram is an alternative way of viewing a fault tree. A circuit diagram uses the graphical notation of an electrical circuit to indicate the possible ways in which the top event in a fault tree can be true or false.

4.1 Interpreting Circuit Diagrams

The circuit diagram consists of all basic events in the fault tree, connected by branches. AND and OR gates are not displayed as nodes in a circuit diagram but instead define the diagram's structure. Value nodes are not displayed in the diagram, only events and gates.

As its name suggests a circuit diagram is analogous to an electrical circuit. A light bulb is attached to the left end of the diagram. The bulb is initially yellow, indicating that the circuit is closed and the light bulb is on, as shown in Figure 4-1. If the circuit is open or broken, the bulb turns gray (off). A broken circuit (i.e., a circuit in which there is a fault) is equivalent to the top level event occurring. In the water shortage fault tree you built in Chapter 3, the top level event occurring means a water shortage occurs.

As mentioned above, the top level event in the fault tree is represented by a larger circle at the left end of the diagram which is either yellow or gray. Basic events are represented in the circuit diagram as slightly smaller circles. AND gates are represented by vertical connections between events. E.g., the basic events GF and GOS in Figure 4-1 are connected by an AND gate. OR gates are represented by horizontal connections in the diagram. E.g., GOS and GOF are connected by an OR gate.

If a basic event is in a FALSE state (does not occur), its circle will be green and current is able to flow through. In Figure 4-1 all basic events are in a FALSE state. If the basic event is in a TRUE state, it's circle is red and current is not allowed to flow through. You can use the circuit diagram to determine which assignments of true and false to the basic events break the circuit and cause the bulb to go gray, i.e, the top event to be true. Probability values associated with basic events are ignored here.

A robust system with many redundancies will have a taller circuit diagram, whereas a frail system will have a wider circuit diagram with many points of failure.

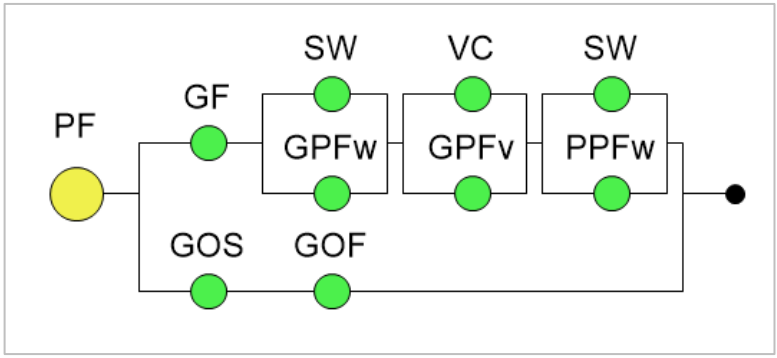

Figure 4-1. An Example of a Circuit Diagram with a Closed Circuit

Below are two simple circuit diagram examples to show how gates define the structure.

The circuit diagram in Figure 4-2 shows that if either A **or** B is in its true state, the circuit will be broken. The connection structure on the right represents an OR gate.

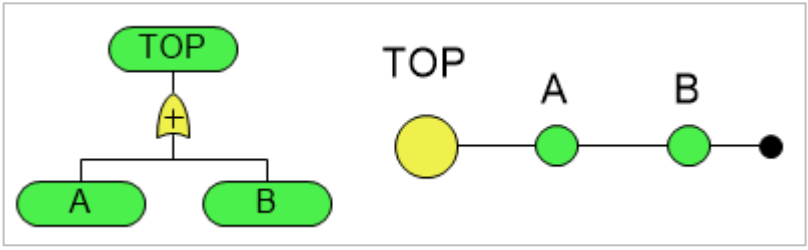

Figure 4-2. A Circuit Diagram Representing an OR Gate

The circuit diagram in Figure 4-3 shows that both A and B must be in their true state for the circuit to be broken. The connection structure on the right represents an AND gate.

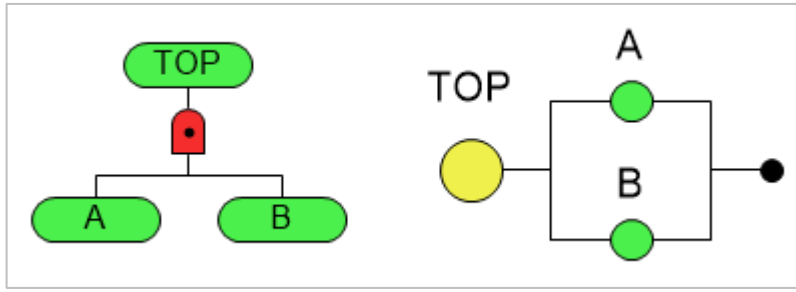

Figure 4-3. A Simple Circuit Diagram Representing an AND Gate

If reference nodes to an event appear in a fault tree, then the event will also appear multiple times in a circuit diagram. However, it is the same event. If you set the event to be true in one place, it will become true everywhere it appears. The reference node *ND1 in the water shortage fault tree from Chapter 3 is an example of this.

If a NOT gate is included in a fault tree, the event's name has a tilde symbol next to it and is initially colored red to indicate that the event is in a TRUE state. The basic event MD is connected with a NOT gate in the circuit diagram shown in the Figure 4-4. NOT gates will be discussed in more detail later in this guide.

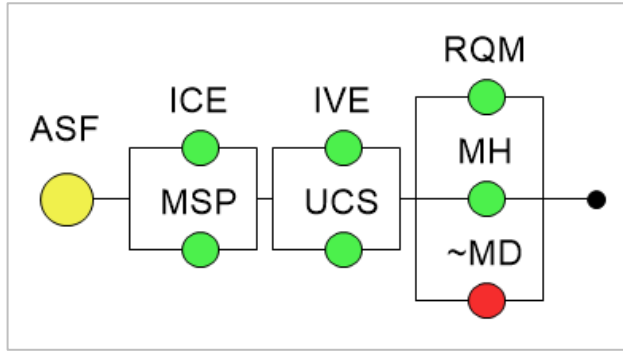

Figure 4-4. An Example of a Circuit Diagram that Includes a NOT gate

Keep in mind that a circuit diagram represents the fault tree in a single time period. If a fault tree uses time series, DPL will prompt you to choose a time period before you can view the circuit diagram. Time series will be discussed in Chapter 8.

4.2 Create and Manipulate a Circuit Diagram

You're now going to view a circuit diagram for the completed fault tree built in Chapter 3. If it is not already open, open the file you saved at the end of the Chapter 3 tutorial.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ If you have a working file from Chapter 3:
  Navigate to the folder where you saved your file and open it.
- ⇨ If you don't:
  Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.

  Select Water_Shortage_Done.da and open it.
- ⇨ To switch from the Fault Tree view to the Circuit Diagram view click the Fault Tree | Display | Circuit Diagram button or press the Tab key to toggle between views.

DPL creates a circuit diagram representing the fault tree as shown in Figure 4-5.

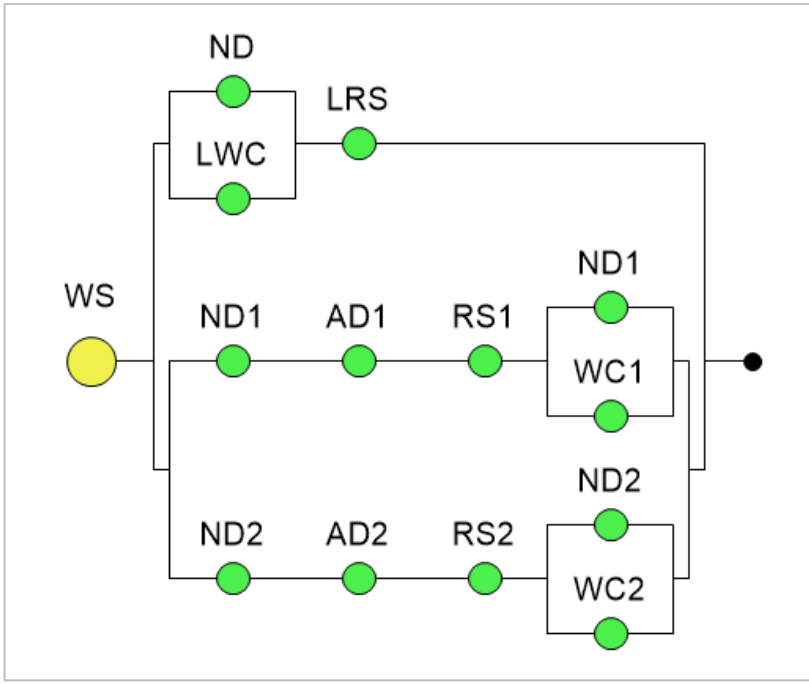

Figure 4-5. Circuit Diagram View for Water Shortage Fault Tree

On the left end of the diagram is a yellow "lightbulb". All other nodes correspond to basic events in the fault tree which are named using their short names by default. You can opt to view the long names for nodes instead by toggling the Fault Tree | Display | Long Names command.

All nodes are initally green which means they are all in their false states. You will toggle the state of some of the nodes to TRUE to see what breaks the circuit (causes a fault).

⇨ Click on the LRS node to toggle its state.

The node turns red, indicating it's in a TRUE state. Note that the bulb on the left is still yellow – a local rainfall shortage occurring is not enough to cause a water shortage.

⇨ Click on one of the ND1 nodes to toggle its state.

Notice that both instances of the event ND1 turned red indicating that they are both in a TRUE state. In the fault tree, recall that you created a reference node (*ND1) to the node ND1. These two nodes in the fault tree represent the same event, as such, if ND1 is true in one place it must be true everywhere. Note that in the circuit diagram the events do not have

different names as they do in the fault tree. The *ND1 nomenclature in the fault tree is meant solely to give you an indication that the node is a reference. Again, setting ND1 to true is not enough to cause a system fault (gray bulb).

⇨ Click on the AD2 node to toggle its state.

The bulb on the left has turned gray, indicating that the diagram represents a broken circuit (a system fault). From this, you can see that the events Local Rainfall Shortage (LRS), Natural Disaster 1 (ND1), and Age Damage 2 (AD2) being in the TRUE state are sufficient to cause the top event (Water Shortage) to be in its TRUE state.

⇨ Click on LRS, ND1, and AD2 again to toggle them back to their FALSE state.

⇨ Try out some other combinations of events to see what causes the light bulb to go out.

⇨ When you are done, press the Tab key or select Fault Tree | Display | Circuit Diagram to return to the fault tree view.

Note the notion of an event being in one state or another only applies to the circuit diagram. When you revert to the normal fault tree view, event states are not saved and events are true or false with a certain probability.

5. Cut Sets and Partial Derivatives

This chapter covers two additional analyses that can be performed on Fault Trees: Minimal Cut Sets and Partial Derivatives. Both of these are intended to help you understand the sources of risk and the relative importance of the various events in a fault tree.

5.1 Cut Sets

In a properly designed fault tree, the top event will be true for some combination of basic events. That is, there should be some set of basic events such that if each event in the set occurs, the top event will occur. Such a combination of basic events is called a *cut set* . Elements of a cut set may be basic events or the negation of basic events if NOT gates are present. A *minimal cut set* is a cut set such that if any basic event is removed the remaining events will not be a cut set and the top event will not be true.

The probability of a minimal cut set is the probability of all the events in the cut set being TRUE (for regular connections; FALSE for NOT gates) which is the product of the probabilities of the basic events. This is equivalent to the probability of the event that is formed from connecting all the elements of the cut set to an AND gate.

The minimal cut sets, along with their associated probabilities and costs, help identify the most likely or least costly ways for the top event to occur.

5.1.1 Generating and Viewing Minimal Cut Sets

You will generate minimal cut sets for the completed Fault Tree built in Chapter 3. If it is not already open, open the file you saved at the end of the Chapter 3 tutorial.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ If you have a working file from Chapter 3:
  Navigate to the folder where you saved your file and open it.
- ⇨ If you don't:
  Navigate to the Examples folder installed with DPL Fault Tree, usually

C:\Program Files\Syncopation\DPL9FaultTree\Examples.

Select Water_Shortage_Done.da and open it.

⇨ Select Fault Tree | Analysis | Minimal Cut Sets. The Minimal Cut Sets dialog will open as shown in Figure 5-1.

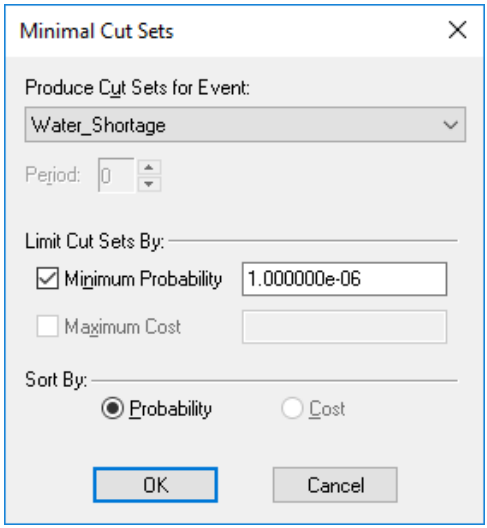

Figure 5-1. Minimal Cut Sets Dialog

At the top of the dialog is a drop-down list from which you can select the event for which you'd like to produce minimal cut sets. This is usually the top event of the fault tree, in this case Water Shortage, which is selected by default. If you choose something other than the top event, DPL will analyze the subtree below the event specified to find the minimal cut sets that result in the selected event being true.

A large fault tree can have a large number of distinct minimal cut sets. In fact, you may have so many minimal cut sets that it is most informative to only look at those that have some reasonable chance of occurring. Consequently, the Cut Sets dialog allows you to limit the number of cut sets displayed by a minimum probability or maximum cost (discussed later).

By default, DPL has set the Minimum Probability to "1.000000e-06" as shown in Figure 5-1. This indicates that you will only be shown those cut sets whose probabiltiy of occurrence is at least 1.0e-6. You can enter any number between zero and one in this box.

The Maximum Cost checkbox is grayed out as there are no costs included in the fault tree.

Lastly, you can sort the cut sets generated by descending probability or ascending cost. Again, you can only sort by probability in this instance.

⇨ Accept the default settings and click OK to have DPL generate minimal cut sets.

The minimal cut sets are calculated and since the cut sets were sorted in decreasing order by probability, the highest probability cut set is displayed first within the Cut Set Viewer as shown in Figure 5-2.

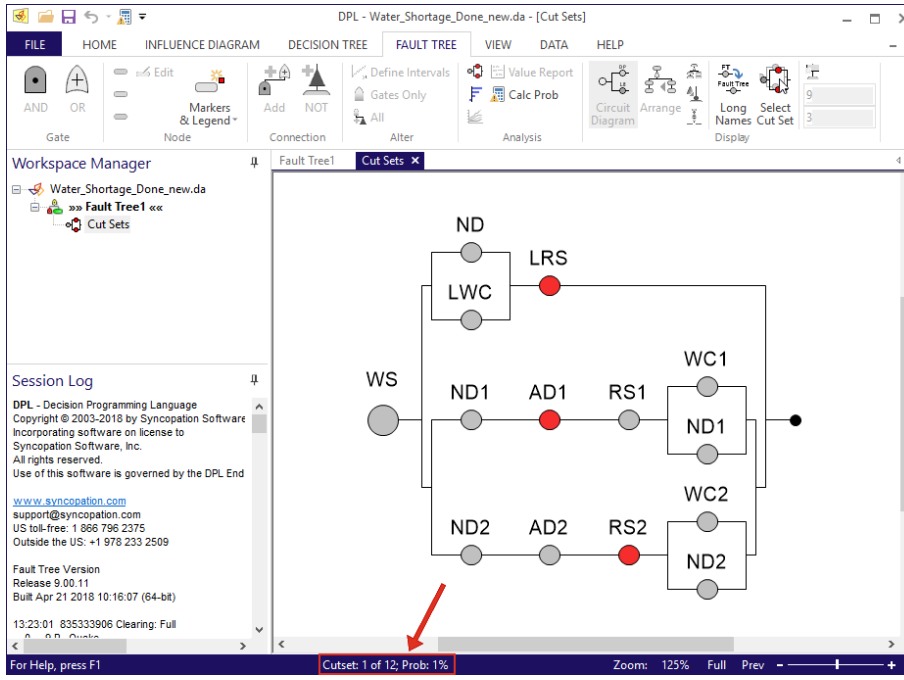

Figure 5-2. Circuit Diagram displaying the first Minimal Cut Set within the Cut Set Viewer

First note that the Cut Set Viewer uses the same notation and symbols as a Circuit Diagram: the top level (or other event selected in the Cut Sets dialog) is represented by a larger circle on the left, basic events are represented by smaller circles, AND gates result in vertical connections and OR gates result in horizontal connections. For the remainder of this section, top event refers to the event for which cut sets were calculated not necessarily the top event of the entire fault tree.

The probability (and cost, if available) of the currently displayed cut set is shown in the status bar at the bottom of the screen as shown in Figure 5-2. The status bar tells you that you're currently viewing the first cut set of 12. The probability of its occurrence is listed as 1%. The percentage is likely much lower but is being rounded to a whole number. The number of decimal places displayed for percentages can be updated within File | Options | Outputs. The cut set probabilities are listed more precisely elsewhere.

Notice that the bulb on the left is gray indicating a broken circuit. This makes sense because a cut set is a set of events that cause the occurrence of the top event. In the Cut Set Viewer, the light bulb is always gray and events in their true state are red. Furthermore, events in their false state in the cut set viewer are gray, whereas in the circuit diagram view of the fault tree they are green, which helps to distinguish the two windows.

You can see from this diagram that the events LRS (Local Rainfall Shortage), AD1 (Age Damage 1), and RS2 (Rainfall Shortage 2) being true cause the top event, Water Shortage, to be true. The short names for events are displayed in the diagram. As mentioned previously, you can toggle between viewing the short names and long names for events within a circuit diagram by selecting Fault Tree | Display | Long Names.

In the Cut Set Viewer window, you can scan through the minimal cut sets by using the arrow keys or via a selection dialog. You'll do the former first.

⇨ With the Cut Set Viewer active, use the up and down arrow keys to cycle through the 12 minimal cut sets and then return to the first.

Now you'll view the minimal cut sets as listed within the Select Cut Set dialog.

⇨ To see the list of cut sets select Fault Tree | Display | Select Cut Set or double-click in whitespace within the Cut Set Viewer window. The Select Cut Set dialog appears as shown in Figure 5-3.

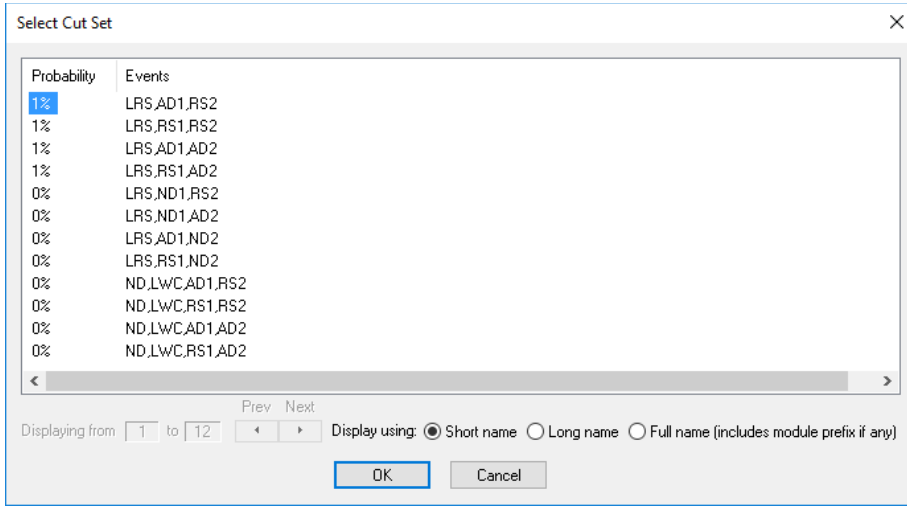

Figure 5-3. Select Cut Set Dialog

Again, the dialog lists the minimal cut sets from mostly likely to least likely to occur as specified within the Cut Sets dialog. The first column lists the cut set's probability of occurrence, then cost is listed (if applicable), and lastly the events that make up the cut set. Notice that the *Display using* radio button is currently set to Short name. You can view the long name or full name for events.

Note that the Fault Tree only has 12 cut sets so all are initially displayed within the dialog. If there were more cut sets than could be displayed at once, you can use the Next and Prev buttons to increment/decrement the group of cut sets being displayed.

⇨ From this list, double click on any cut set to view it in Cut Set Viewer.

Cut set information is also written to the Session log in a similar manner to the Select Cut Set dialog. See Figure 5-4.

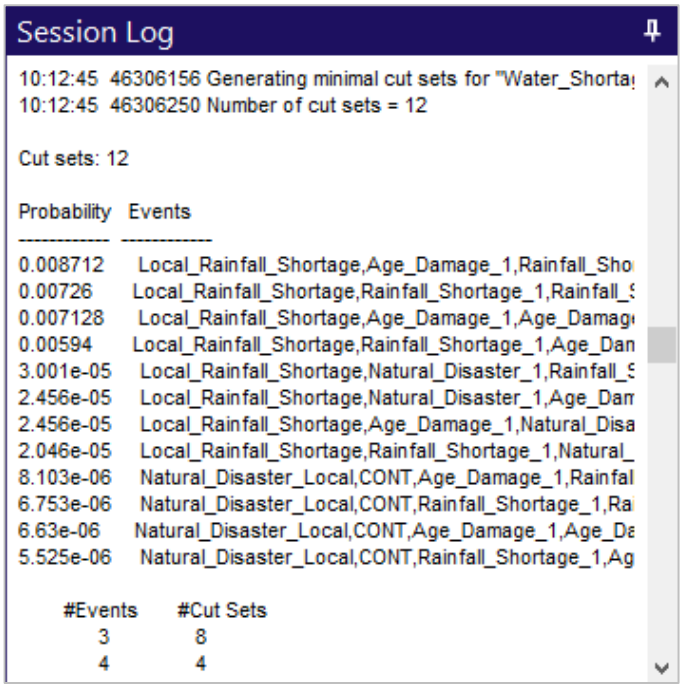

Figure 5-4. Minimal Cut Set Information listed in the Session Log

You can export the cut set data as a CSV for further analyses by selecting Data | Export | Data while the Cut Set Viewer is active. You'll be prompted as to which name type should be used in the export: short, long, or full (Figure 5-5).

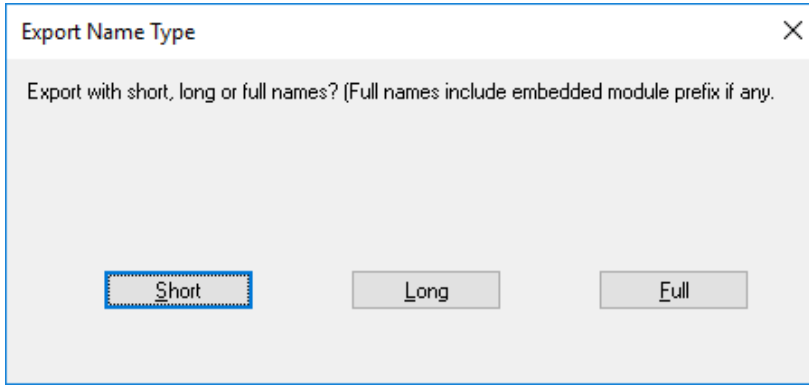

Figure 5-5. Export Name Type Prompt when Exporting Cut Set Data

5.1.2 How Costs are used in Minimal Cut Sets

In addition to a probability value, it is possible to specify a true state cost and a false state cost for a basic event. The true state cost of an event is the cost of forcing the event to be true. As you may expect, the false state cost is the cost of forcing the event to be false.

These cost values are only used by DPL Fault Tree when generating minimal cut sets. The cost of a minimal cut set is the cumulative cost of all elements of the cut set occurring (or not occurring if NOT gates are present). If applicable, DPL displays the cost of each cut set along with its probability of occurrence. This may help you determine the least expensive ways in which someone could deliberately cause harm to the system.

If cost data is specified for all basic events in a fault tree, DPL Fault Tree allows you to limit cut set generation by maximum cost as well as minimum probability restrictions. You may also opt to sort the cut sets by cost from lowest to highest.

In this section you looked at the minimal cut sets for the Water Shortage fault tree built in Chapter 3, which does not include cost data. Later is this guide minimal cut sets will be generated for a fault tree that includes cost data.

5.2 Partial Derivatives

Partial Derivatives are a form of sensitivity analysis intended to provide information about the relative importance of the basic events in a fault tree. A fault tree is essentially a large, deeply nested probability expression. As such, there is an explicit mathematical formula for the probability of the top event in terms of all the basic events. DPL calculates the partial derivatives of this formula with respect to each basic event. The top level event is always a linear function of each basic event, so the result is a constant associated with each basic event.

A large value for an event's partial derivative means that changing that event's probability would have a large impact on the probability of the top event.

Minimal cut sets offer a method of examining key failure scenarios or combinations of events. Partial derivatives help to identify individual events that increase the likelihood of the top level event being true.

5.2.1 Generating Partial Derivatives

You will calculate partial derivatives of all basic events for the top event of a fault tree.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.
- ⇨ Select Water_Shortage_3_Sources.da and open it.

This is a slightly more complex version of the Water Shortage fault tree built in Chapter 3.

- ⇨ Select Fault Tree | Analysis | Partial Derivatives.

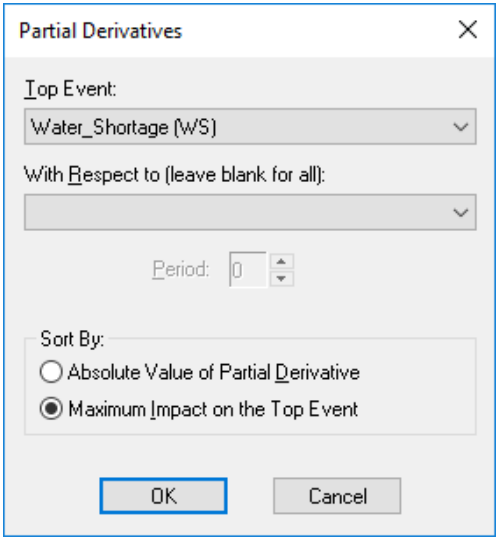

Figure 5-6. Partial Derivatives Dialog

DPL will display the Partial Derivatives dialog as shown in Figure 5-6. The dialog is divided into four sections:

Top Event

Select the top event for which you wish to calculate partial derivatives. The default is the top event for the entire fault tree, the most common choice. In the remainder of this section we'll refer to the event selected here as the top event.

With Respect to (leave blank for all)

If you are interested in the partial derivative of the top event with respect to a particular event, select it from the drop-down list. Leave this blank for the partial derivatives with respect to all basic events.

Period:

If your fault tree has multiple time periods defined, select the desired time period. Time series will be discussed in Chapter 8.

Sort By

Select how you would like the output sorted. You may sort by either the absolute value of the partial derivative or the maximum impact on the top event. We'll explain the maximum impact below.

⇨ Accept the defaults and click OK to calculate partial derivatives.

The results of the run are displayed in a bar chart as show in Figure 5-7.

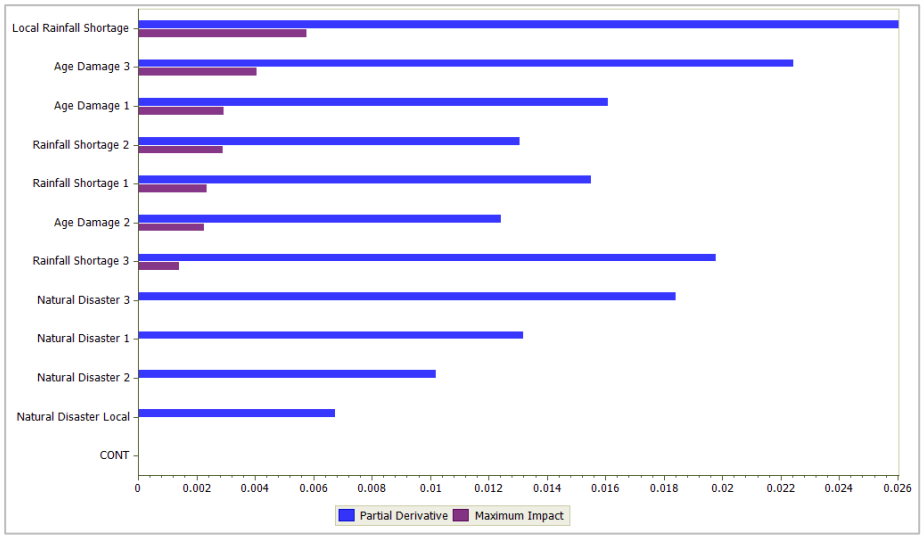

Figure 5-7. Partial Derivatives Chart for Water Shortage

DPL displays the partial derivatives in a horizontal bar chart sorted in descending order of the maximum impact on the top event, as specified in the set up dialog. The partial derivative of each event is represented by a blue bar and the maximum impact on the top event is represented by a purple bar. In this chart, all the partial derivatives are positive which will

always be the case in a fault tree which has no NOT gates. Partial derivatives are negative for events connected with or below a NOT gate. The partial derivative information is also be printed to the Session Log as shown in Figure 5-8.

Session Log

Basic Events: 18 Derived Events: 16

Partial derivatives of "Water_Shortage" with respect to all basic binary events (* = expe

Partial Deriv	Probability	Maximum Impact	Event
0.026	0.22	0.00573	Local_Rainfall_Shortage
0.0224	0.18	0.00403	Age_Damage_3
0.0161	0.18	0.00289	Age_Damage_1
0.013	0.22	0.00286	Rainfall_Shortage_2
0.0155	0.15	0.00232	Rainfall_Shortage_1
0.0124	0.18	0.00223	Age_Damage_2
0.0198	0.07	0.00138	Rainfall_Shortage_3
0.0184	0.00062	1.14e-05	Natural_Disaster_3
0.0132	0.00062	8.17e-06	Natural_Disaster_1
0.0102	0.00062	6.3e-06	Natural_Disaster_2
0.0067	0.00062	4.16e-06	Natural_Disaster_Local
1.26e-05	0.33	4.16e-06	CONT

Figure 5-8. Session Log displaying Partial Derivative Information

5.2.2 Interpreting Partial Derivatives

The partial derivatives tell you how much a change in the probability of each basic event affects the probability of the top event. This value depends on the structure of the fault tree and the probabilities of other events, but not the probability of the basic event itself.

The maximum impact of each basic event shows how much the probability of the top event can be reduced by setting the probability of that basic event to zero. This is calculated as the partial derivative of that event multiplied by its probability.

If a particular basic event is in every cut set, the maximum impact of that event will be the probability of the top event. I.e., the maximum value of the maximum impact is the probability of the top event.

In this example, Water Shortage is the top event. The probability of its occurrence is 0.0057 (this result is not indicated in the partial derivatives output).

Let's take a closer look at the partial derivative information within the Session Log (Figure 5-8). The first column shows the partial derivative value for the event. So the partial derivative of Water Shortage with respect to Local Rainfall Shortage is 0.026. This means that increasing the probability value of Local Rainfall Shortage by 0.1 (say, from 0.22 to 0.32) increases the probability of Water Shortage by 0.1x0.026, or 0.0026 (from 0.0057 to 0.0083). You could confirm this by changing the probablity data for LRS and calculating the probably of occurrence for the top event and comparing that result to that of the original fault tree.

The second column indicates the probability of each basic event. In this example, these values are simply the probabilities entered in the fault tree. In a decision model with an embedded fault tree, some uncertainty can be introduced into these values, and the expected values of the probabilities would be shown here. See Chapter 9 for information on embedding fault trees in a decision model.

The third column, titled Maximum Impact shows the greatest amount the probability of Water Shortage can be changed by altering the probability of the basic event in the row to be zero. The maximum impact answers the question "how much would the risk of a Water Shortage be reduced if I could prevent this event (i.e., set its probability to 0)?".

Maximum Impact and Relative Risk

Which events are the largest sources of risk? In most cases the events with the largest maximum impact values are the ones you should be most concerned about. Basic events with large partial derivatives but low maximum impacts are usually rare events with a short path to the top event.

Consider the fault tree in Figure 5-9. A worker can be late to work due to a bridge collapse, or the combination of a flat tire and the lack of a good spare.

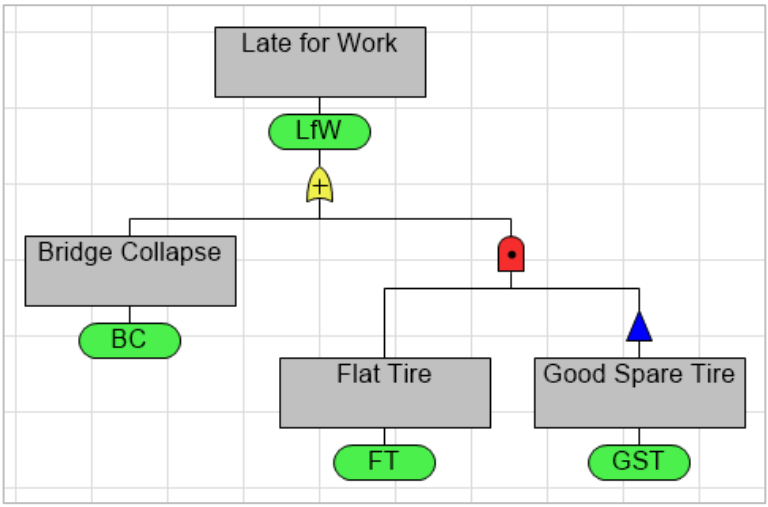

Figure 5-9. Fault Tree for the Risk of being Late for Work.

Given the structure of this fault tree the partial derivative of Bridge Collapse will always be 1, regardless of the probabilities specified, because this event has a direct connection to the top event. However, bridge collapses are very rare, so as a practical matter the Bridge Collapse event is probably not the greatest risk for worker tardiness. With notional probabilities (Bridge Collapse 1e-7, Flat Tire 1e-3, Good Spare Tire 0.8) the partial derivatives and maximum impacts for this model are as shown in Figure 5-10 below.

Session Log

Basic Events: 3 Derived Events: 1

Partial derivatives of "Late_for_Work" with respect to all basic binary events (* = expec

Partial Deriv	Probability	Maximum Impact	Event
0.2	0.001	0.0002	Flat_Tire
-0.001	0.8	0.0002	Good_Spare_Tire
1	1e-07	1e-07	Bridge_Collapse

Figure 5-10. Partial Derivative Information for Late for Work Fault Tree

6. True/False Costs and Fault Tree Inversion

A fault tree provides the probability that the top event in the tree is true. You may also be interested in knowing the least costly way in which the top event can be true. You can do this by providing True costs for basic events and running Minimal Cut Sets. You will look at this feature now.

6.1 True Costs

To do this, you will consider an example of a fault tree representing the probability of a secure facility being compromised.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.

  Select Secure_Facility.da and open it.

There are three ways to gain access to the secure facility: bribery, coercion or infiltration. Access will be gained if any of these happen, hence the top level event in the tree (ASF) is an OR gate. For each of the three modes of access, two or three conditions must be met so the event for each mode of access (GAbB, GAbC, GAbI) is an AND gate. See Figure 6-1.

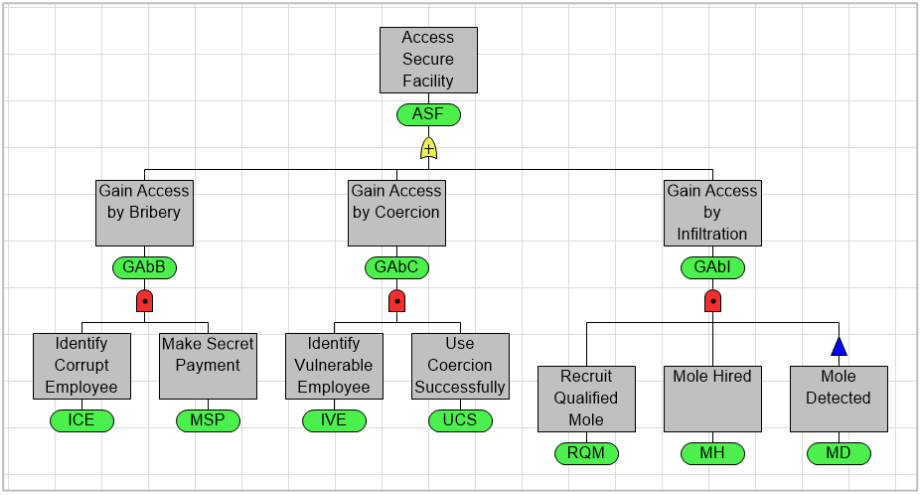

Figure 6-1. Secure Facility Fault Tree

You will examine the minimal cut sets of the fault tree as it currently stands.

⇨ Select Fault Tree | Analysis | Minimal Cut Sets.

⇨ Accept the defaults in the Minimal Cut Sets dialog and click OK.

The cut set viewer comes up with the most likely cut set (ICE and MSP true) being displayed.

⇨ Double-click the cut set view to bring up the Cut Set dialog. See Figure 6-2.

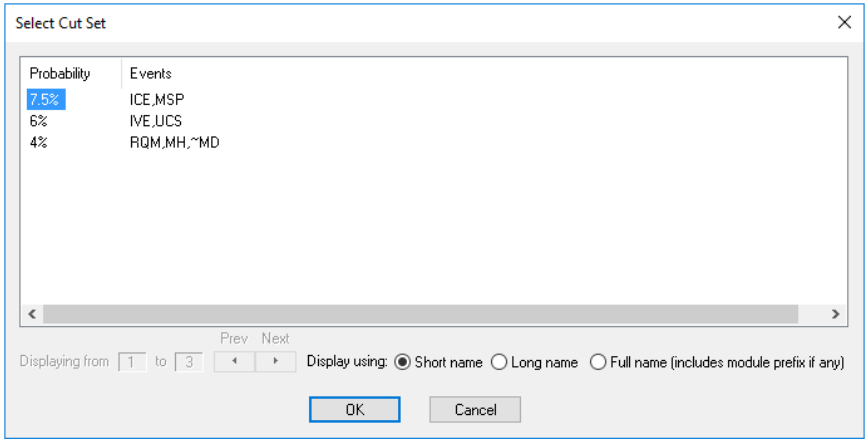

Figure 6-2. Select Cut Set Dialog for Secure Facility

There are three cut sets with probabilities of 7.5%, 6% and 4%. You might like to know which of these is the least costly way for an attacker to compromise the secure facility. You will now introduce true costs to accomplish this. A true cost is the cost to force an event to be true.

⇨ Click OK to close the Select Cut Set dialog and activate the Attacker Perspective fault tree.

⇨ Double-click ICE to edit the node.

⇨ Enter 50 in the True Cost column on the Data tab.

⇨ Click OK.

⇨ Repeat the above process providing True Costs according to Table 6-1 below.

Short Name	True Cost
MSP	500
IVE	200
UCS	10
RQM	150
MH	0
MD	0

Table 6-1. True Costs for Attacker Perspective

⇨ Select Fault Tree | Analysis | Minimal Cut Sets.

Note that you can now limit cut sets by maximum cost and sort by cost in the Minimal Cut Sets dialog.

⇨ Select Cost to sort by and click OK.

Note that the cut set with RQM, MH and ~MD is now initially selected in the Cut Set Viewer. It is the least costly method for an attacker to try to gain access to the secure facility.

6.2 Inversion

Often the top event in a fault tree is some type of failure or an undesirable event. While the probability of the top event not occurring is simply one minus the probability it occurs, you may wish to analyze and/or view the fault tree for the situation where the top event does not occur. This can be accomplished quickly using inversion. You will do this now.

⇨ Activate the Attacker Perspective fault tree.

⇨ Right-click on its item in the Workspace Manager and select Duplicate.

⇨ Select the item for the new fault tree and press F2 to rename it.

⇨ Rename the new fault tree: Defender Perspective. See Figure 6-3.

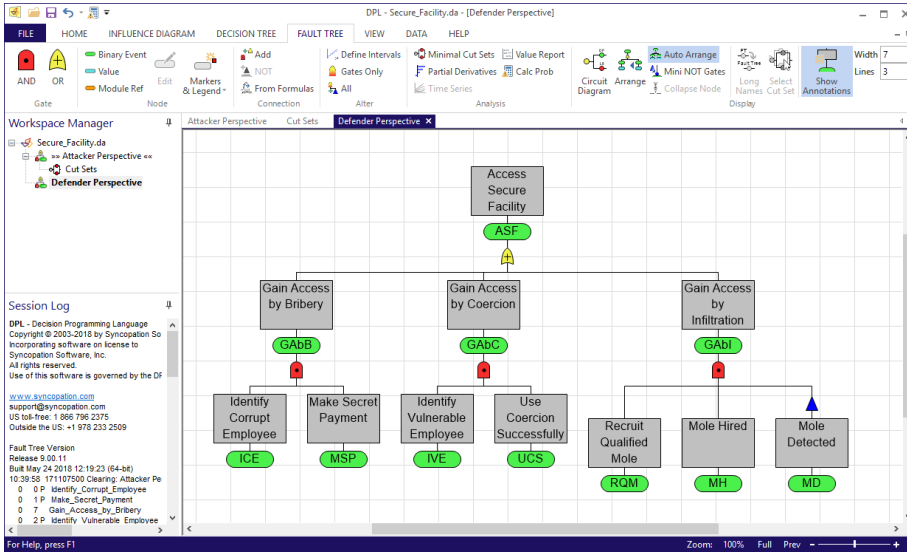

Figure 6-3. Renamed Fault Tree

⇨ Select Fault Tree | Alter | All to invert the fault tree.

Figure 6-4 shows the inverted fault tree with gates already re-named. Note that the top level OR gate has become an AND gate and the lower level OR gates have each become AND gates. Further, the connections of the basic events at the bottom of the tree to the OR gates have changed: regular connections have become NOT connections and NOT connections have become regular.

The first thing you need to update in the inverted fault tree is the true cost for the Mode Detected node. As indicated in Table 6-1, the attacker would not spend anything to detect a mole that they've implanted in the facility. But the defender would spend money to detect a mole implanted by the attacker. You'll incorporate this true cost now.

- ⇨ Double-click MD to edit it.
- ⇨ Enter 115 in the True Cost column of the Data tab.
- ⇨ Click OK.

The naming in the tree is a bit misleading at this point. You will fix that now.

- ⇨ Double-click ASF to edit it.
- ⇨ Delete the Short Name.
- ⇨ Enter "Secure Facility Safe" for the long name. The short name updates automatically. Keep the default.
- ⇨ Click Use long name to transfer the long name to the annotation.
- ⇨ Repeat the above process to rename the remaining three gates according to Table 6-2 below. Don't forget to delete the old short name before changing the long name so it will update automatically.

Old Short Name	New Long Name
GAbB	Access by Bribery Prevented
GAbC	Access by Coercion Prevented
GAbI	Access by Infiltration Prevented

Table 6-2. Rename Gates for Defender Perspective

The Defender Perspective fault tree should now look like Figure 6-4. With the renamed gates, the fault tree can be read more clearly: the secure facility is safe if access by bribery is prevented and access by coercion is prevented and access by filtration is prevented.

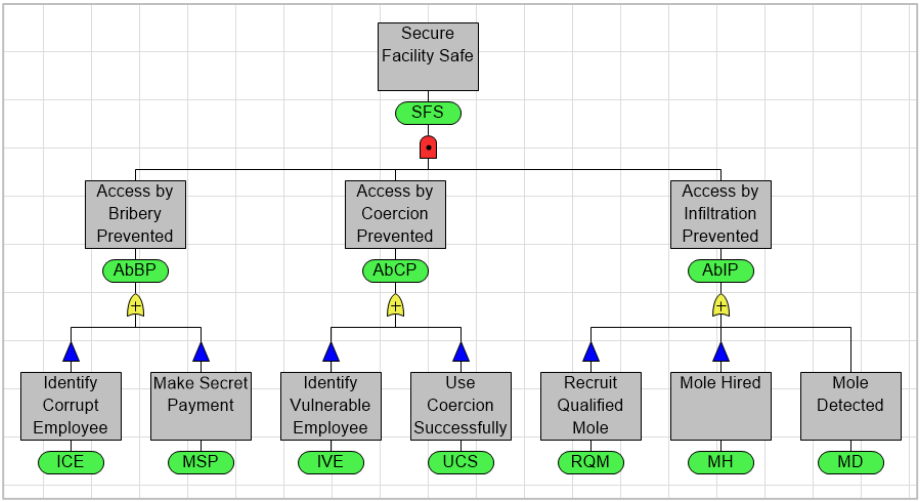

Figure 6-4. Defender Perspective Fault Tree with Renamed Gates

⇨ Calculate the probability of the top event in the Attacker Perspective fault tree by activating it and selecting Fault Tree | Analysis | Calc Prob.

The probability that the attacker accesses the secure facility is 0.17.

⇨ Now switch back to the Defender Perspective fault tree and calculate the probability of the top event in it.

The probability that the defender defends the secure facility is 0.83 which as stated earlier is equal to one minus the probability of the top event being true prior to inversion.

6.3 False Costs

You may wish to know the least costly way to defend the secure facility. To do this, you will introduce false costs. A false cost is the cost to force an event to be false. Note that in the Defender Perspective fault tree most of the basic events are connected via NOT connections. This means that in a

number of cut sets the basic events need to be false for the top event to be true.

⇨ Double-click ICE to edit the node.

⇨ Enter 100 in the False Cost column on the Data tab.

⇨ Click OK.

⇨ Repeat the above process providing False Costs according to Table 6-3 below.

Short Name	False Cost
MSP	0
IVE	150
UCS	0
RQM	0
MH	225
MD	0

Table 6-3. False Costs for Defender Perspective

Now, you will examine the minimal cut sets of the Defender Perspective fault tree.

⇨ Select Fault Tree | Analysis | Minimal Cut Sets.

⇨ Make sure sort by cost is selected and click OK.

The cut set viewer comes up with the least costly cut set displayed. In this case, the least costly cut set is ~MSP, ~UCS, ~RQM. This cut set represents a failure on the attacker to succeed (they fail to make secret payment, fail to coerce successfully and fail to recruit a mole). As a defender, you would not rely on this cut set so it is not of much interest.

⇨ Arrow down to select the next most costly cut set. See Figure 6-5.

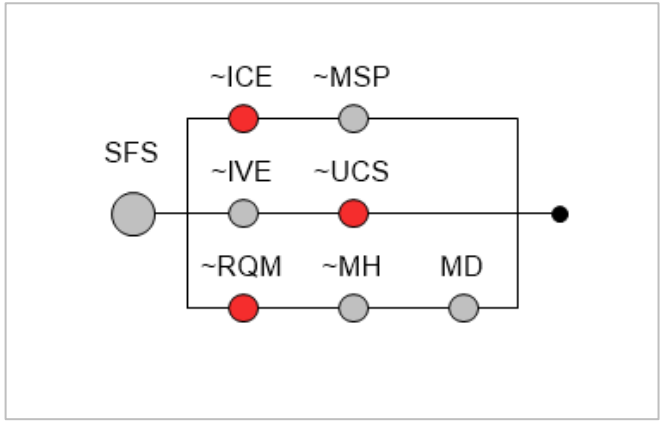

Figure 6-5. ~ICE, ~UCS, ~RQM Cut Set

This cut set represents a situation in which the defender has put in a screening program to weed out corrupt employees. Based on the false costs, it is the least costly action that the defender can take to prevent the facility from being compromised.

6.4 Probability and True/False Cost Predecessors

A binary event may have up to three value node predecessors. The first predecessor provides a probability, the second provides a true cost, and the third provides a false cost for the binary event. If probability data is already specified directly in the binary node's data, then the first value node predecessor provides true costs, etc. The arcs for predecessors that provide true and false costs are marked with a T or F. In Figure 6-6, V1 is a probability predecessor to B1, V2 is a true cost predecessor and V3 a false cost predecessor.

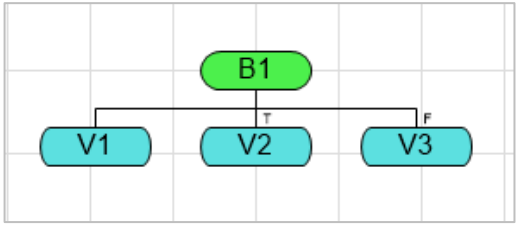

Figure 6-6. Probability, True Cost, False Cost Predecessors

7. Using Fault Tree Modules

While there is no fixed limit on the size of a DPL fault tree model, it is often more convenient to split a complex model into several parts. DPL supports this with features for creating and embedding fault tree modules. A module is simply a fault tree that is referenced by one or more other fault trees. A module reference is a node that represents a fault tree module in the higher-level fault tree. Fault tree modules can be embedded in other fault trees, or in influence diagrams to create a decision model with fault tree calculated probabilities (Chapter 9).

There are two use cases for modules embedded in fault trees. The first is where the module represents a submodel of the overall fault tree with specific, nonvarying probabilities. The second is where the module acts as a subroutine or "user defined gate type", with variable inputs. Modules of the latter type often appear more than once in same fault tree, with different events or probabilities connected to their inputs.

DPL fault trees may not be recursive, and DPL will prevent you from creating a module reference in such a way that a fault tree cycle would exist.

7.1 Embedding a Module as a Submodel in a Fault Tree

In this section you will create a fault tree module to enhance a model analyzing the risk of a power failure at a critical industrial site.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.
- ⇨ Select Power_Failure.da and open it.

A power failure at the facility can occur either because of a general problem with the utility system or because of local issue.

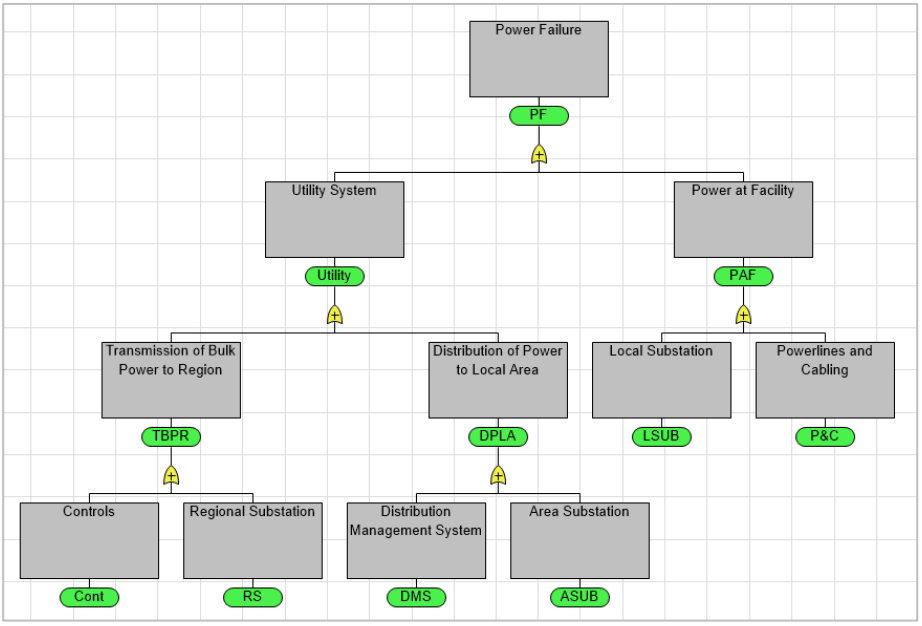

Figure 7-1. Power Failure Fault Tree

One of the events that could affect the reliability of the power system is a failure of transmission controls, the node in the bottom left corner of Figure 7-1. In the first draft of the fault tree this risk was modelled as a single event, but since then further analysis has taken place and we now have a more detailed model of a failure of controls.

⇨ In the Workspace Manager, double-click the fault tree UCntrl.

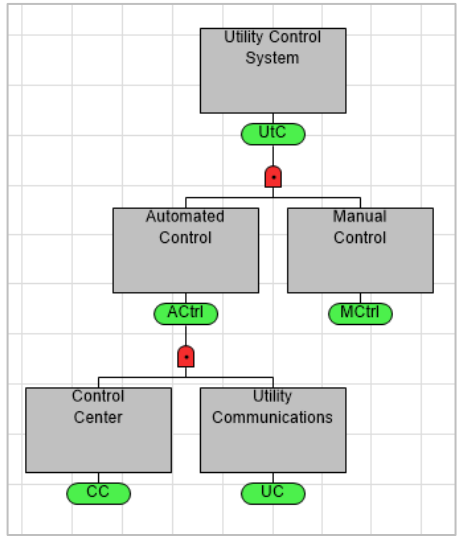

Figure 7-2. Utility Control System Fault Tree

You will use this fault tree as a module in the overall Power Failure model. To do so you'll first activate the Power Failure model and will duplicate it.

- ⇨ In the Workspace Manager, right-click on the Power Failure fault tree and select Duplicate.
- ⇨ Rename the new fault tree to "Power Failure Using Module".
- ⇨ Double-click the new fault tree to activate its window.

You will need to delete the constant probability for a Controls failure.

- ⇨ Double-click on the node named Cont edit it.
- ⇨ On the Data tab, select the probability and press the Delete key.
- ⇨ Click OK.

The Cont event in the Power Failure Using Module fault tree will depend on a module reference to the Utility Control System (UCntrl) fault tree.

- ⇨ Click Fault Tree | Node | Module Ref.
- ⇨ In the Select Module dialog, leave the Source at "(Local)" and select the module UCntrl as shown in Figure 7-3.

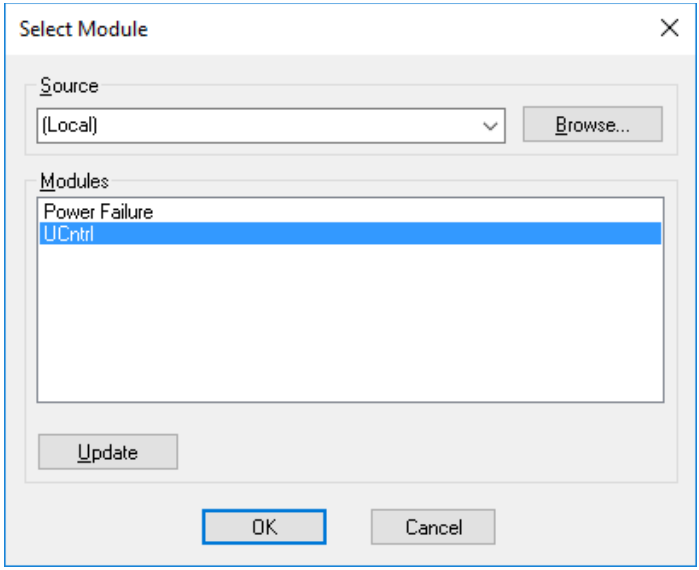

Figure 7-3. Select Module Dialog

You can use a module that is stored in another DPL workspace by changing the source. This allows for simultaneous editing by different team members and can also facilitate libraries of commonly used modules. In our case, the module is local -- i.e., within the workspace.

The Cont event will depend on a module reference to the Utility Control System fault tree.

⇨ Click OK to close the dialog.

There will be a semi-transparent module reference node beneath your cursor ready to be placed.

⇨ Place the module reference node directly on to the Cont event as shown in Figure 7-4.

The UCntrl module reference node is now a predecessor (e.g., an input) to the Controls binary node. Now you'll see what the probability of a power failure is with the module reference node included in the fault tree.

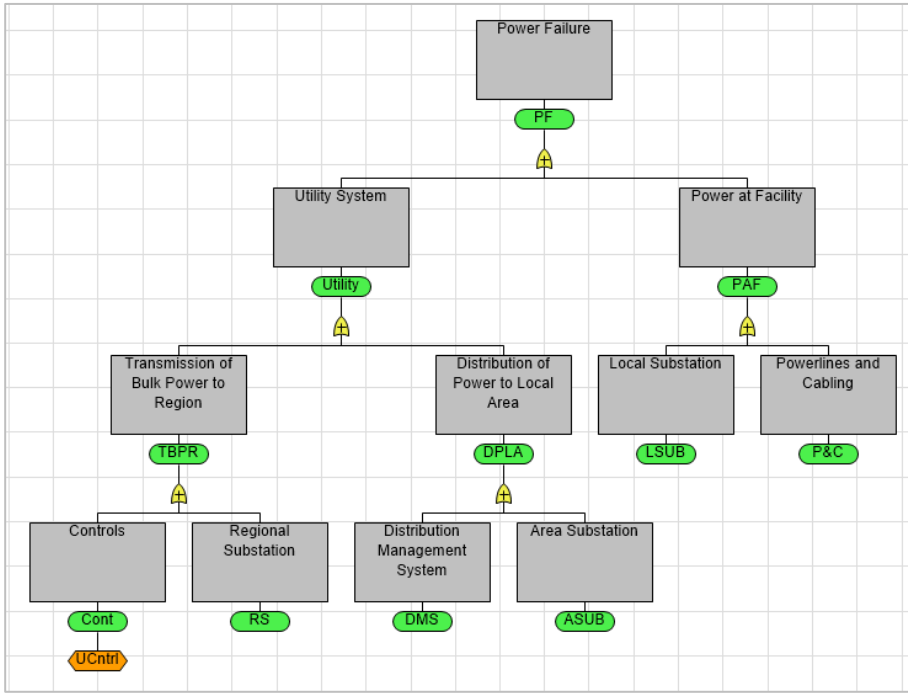

Figure 7-4. Power Failure Fault Tree with Module Reference

⇨ With nothing selected in the fault tree, select Fault Tree | Analysis | Calc Prob to calculate the probability of the top event occurring.

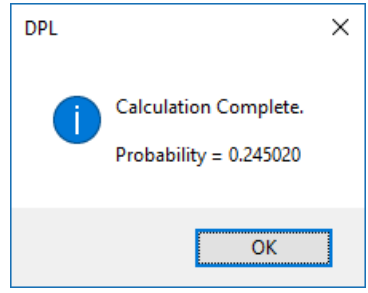

Figure 7-5. Calculation Complete Message

If you see a different number of decimal places, you can change the setting in File | Options | Outputs.

7.2 Embedding a Module with Variable Inputs

A fault tree module is often used more than once in a larger fault tree. The structure of the module may apply to several subsystems within the higher-level model, but with different probabilities or event connections.

In the previous section you used the UCntrl module once in the Power Failure fault tree, for a controls failure associated with the transmission system. In this section you'll modify the module and the main fault tree so it can be used for both transmission and distribution.

You'll first need to modify the UCntrl module so that it takes a variable probability for Manual Control.

- ⇨ In the Workspace Manager, double-click on the UCntrl fault tree to activate it.
- ⇨ Double-click on the Manual Control node. The Fault Tree Node Definition dialog box will be displayed.
- ⇨ Press the Delete key to delete the probability ("0.01").

Since the Manual Control node doesn't have a probability, that value will be an input to the module. Each use of the module will need to supply an appropriate probability.

- ⇨ In the Workspace Manager, double-click on Power Failure Using Module to activate it.
- ⇨ Create a new binary node and drop it on the Distribution of Power to Local Area OR gate.
- ⇨ Name the new event Controls Dist, accept the default short name, and click the button to use its long name as its annotation.
- ⇨ Click Fault Tree | Node | Module Ref and select UCntrl in the dialog.
- ⇨ Place the new module reference node on to the Controls Dist binary node.

Next, you will create two value nodes, for the probabilities of manual control failure in the transmission and distribution subsystems.

- ⇨ Create a new probability value node and place it beneath the UCntrl module reference node on the left-hand side under transmission.
- ⇨ Name the value node "Manual Control Trans" with data "0.01". (Recall that this was the constant value previously in the module.)

⇨ Click OK to close the dialog.

DPL will display the Select Module Input dialog (Figure 7-6). Whenever you connect a value or event to a module reference, DPL will use this dialog to ask you to which input the node should be connected. In this case our module only has one input, the probability of a manual control failure.

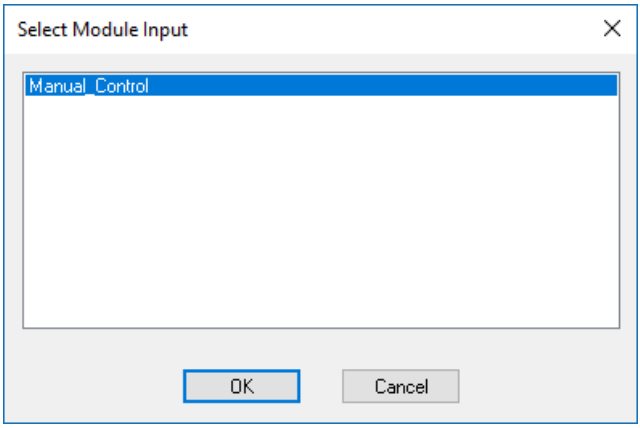

Figure 7-6. Select Module Input Dialog

⇨ With Manual_Control selected click OK to close the dialog.

⇨ Create another probability value node and drop it onto the distribution module reference node.

⇨ Name the node "Manual Control Dist" and enter a probability of 0.15 within the data tab.

The distribution system is much more dispersed, so a failure of manual controls is deemed more likely.

⇨ Click OK to close the dialog.

⇨ Again, select Manual_Control within the Select Module Input dialog.

Your fault tree should look like Figure 7-7.

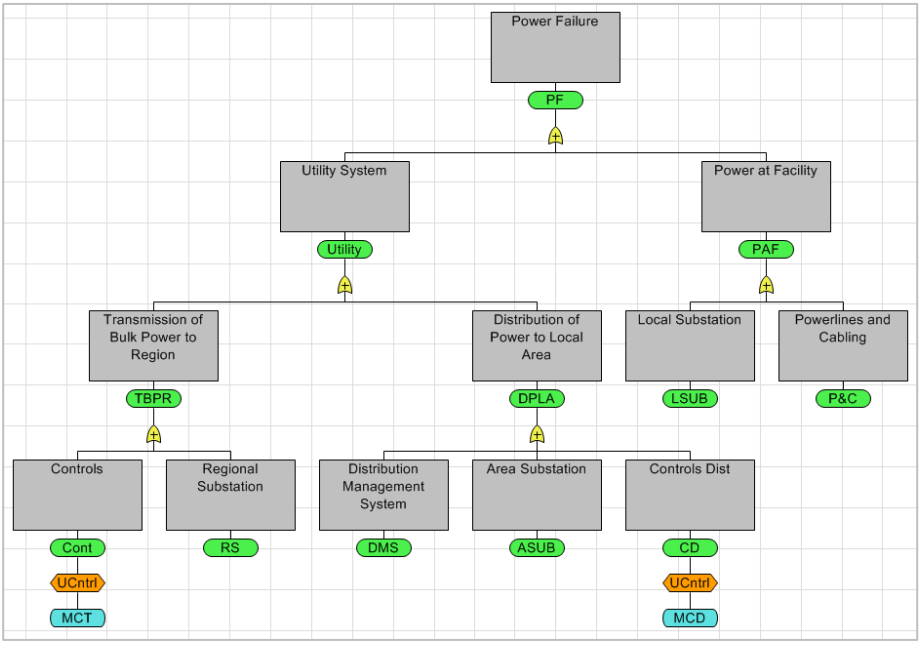

Figure 7-7. Power Failure Fault Tree with Two Module References

Now you'll see what the probability of a power failure is with two module reference nodes that have variable inputs included in the fault tree.

⇨ With nothing selected in the fault tree, select Fault Tree | Analysis | Calc Prob to calculate the probability of the top event occurring.

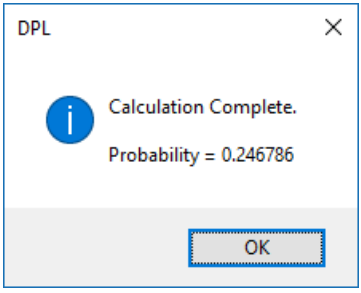

Figure 7-8. Calculation Complete Message

8. Time Series Fault Trees

To this point in this manual, you have worked with binary events and values that are scalars, i.e., the input and output of each is a single number. However, the probability and cost data of a basic event (or value) can be defined as vectors. Often an event or value defined as such represents how the data may change over time. For this reason, vector events or values are referred to as time series events/values. Time series events are defined by intervals. As mentioned, these intervals, which usually represent blocks of time. However, the notion of time is not inherent in the definition of a time series event/value. The intervals could represent spatial differences for an event or value, e.g., such as sections of a pipeline. Time series events/value can be used to represent anything that varies in a single dimension.

Setting up a time series fault tree in DPL is a two-stage process in which one or more series intervals are defined and then one or more time series nodes are created.

8.1 Defining Time Series Intervals

As discussed, a series is a way of defining vector-valued events/values instead of single-valued events/values. While series are most often used to represent data that change over time, they can also be used for other applications. For example, the elements of a series might represent sections of a pipeline, separate sites vulnerable to attack, or business units of a corporation. Within the tutorial in this chapter you will define a series interval that represents time. More specifically, you will introduce a time series to the Water Shortage fault tree completed in Chapter 3.

⇨ If necessary, open DPL Fault Tree.

⇨ Select File | Open.

⇨ If you have a working file from Chapter 3:
Navigate to the folder where you saved your file and open it.

⇨ If you don't:
Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.

Select Water_Shortage_Done.da and open it.

Recall that the two aqueducts that bring water from Sources 1 and 2 will fail if a natural disaster occurs at the source or if the aqueduct suffers from age damage. Currently the probability that the aqueducts fails due to age damage is 0.18 and is contained within a scalar value node.

Engineers now believe that due to general wear and tear over time the likelihood of each aqueduct failing due to age damage increases by 1% annually over the each of the next 10 years. To capture this gradual deterioration, you will set up a time series interval of 10 years and then will change the ADprob value node into a series node.

Define the time series interval for the fault tree as follows.

⇨ Select Fault Tree | Alter | Define Intervals.

⇨ Click the Add button to add an interval.

⇨ Enter a 1 in the "From" column and a 10 in the "To" column. The dialog should look like Figure 8-1

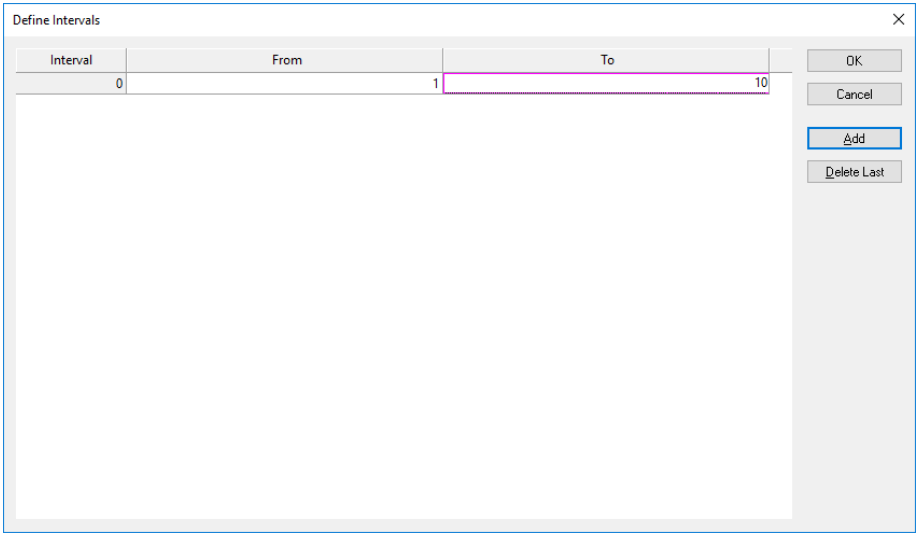

Figure 8-1. Define Intervals Dialog with 10 Year Time Interval Defined

Note: interval boundaries must be contiguous; hence, the To value is only necessary for the last interval. If the first interval goes from 0 and the

second goes from 5, then the first interval goes to 4. The From value of the first interval can be a non-negative integer. All From values must be increasing over the intervals. The To value of the last interval must be greater than or equal to its From value.

8.2 Creating Time Series Nodes

Now you'll change the ADprob value node, which supplies the probability data for the AD1 binary event, to be a series value node. Recall that there is a copy of the ADprob node in the Source 2 subtree that references this value node. The data of the referenced node (*Adprob) will be updated automatically to a series node as well.

⇨ Double-click the ADprob node within the Source 1 subtree to edit it.

⇨ Select the General tab. Now that intervals have been defined the Time series value radio button is active, select it as shown in Figure 8-2.

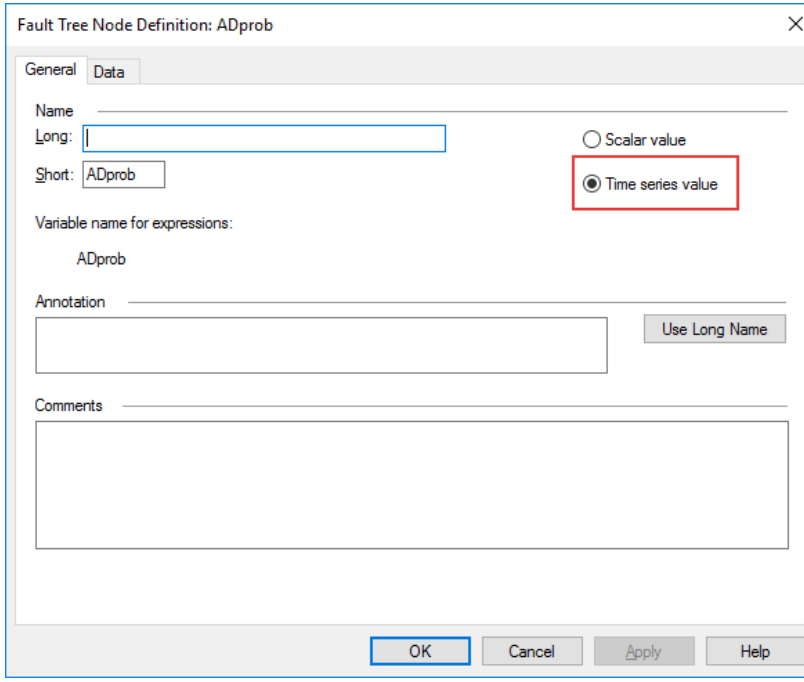

Figure 8-2. Time Series Value Selected within the General Tab of Node Definition Dialog for ADprob

⇨ Switch to the data tab. Notice that the From and To boxes now display the series interval defined for the fault tree as shown in Figure 8-3.

Now you'll enter a formula that captures the increase in age damage. To do so, you'll use a relative subscript. A relative subscript is a special identifier ($) that represents the current subscript of the series being defined.

⇨ Edit the Value column to be a formula "0.18+0.01*[$-1]" as shown in Figure 8-3.

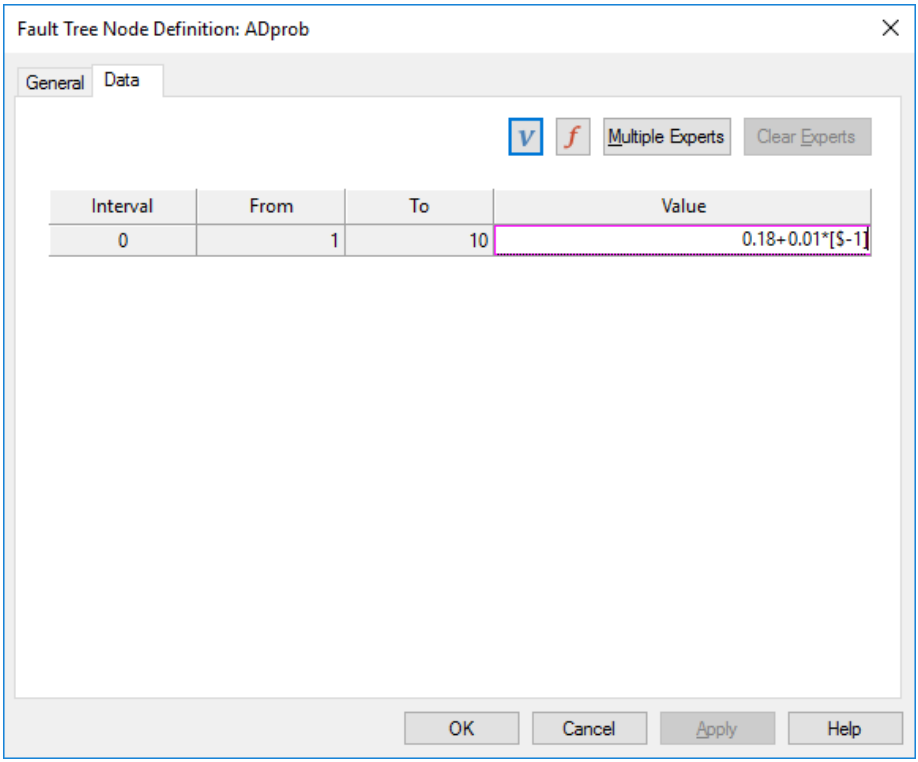

Figure 8-3. Data Tab of the Node Definition Dialog for Series Value Node ADprob

It is possible to define the time-dependent values explicitly with constants or as a function of the time period. You've done the latter. The relative subscript [$] takes on the value of the index, which is the time period the variable is in at the time. In this case, the series numbering starts at 1. Hence, in the first time period nothing is added to the age damage probability since the 0.01 increment is multiplied by [$-1], which equals zero in time period 1.

⇨ Click OK to close the node definition dialog.

DPL now displays a vector symbol (rightward pointing arrow) near the top, left corner of both the series value node and the series binary node it feeds to indicate they are series nodes. If you expand out the Source 2 subtree you'll find that the reference value node *ADprob and the binary node AD2 have been updated to be series nodes as well.

A time series node cannot be a predecessor to a scalar node. Once you change a scalar node to a time series node, DPL changes all successors of that node into time series. A scalar can be a predecessor to a time series node. In this situation, the scalar is a predecessor of the first interval of the time series node. If no data is specified for subsequent intervals of the successor, then the scalar is used as the predecessor for all intervals of the series successor.

If a gate has any series predecessors, then it is a series as well. Gates with only scalar predecessors are scalars. Since you cannot explicitly edit the data of a gate, the series indicator is not displayed on gates even if they are series.

You'll now calculate probabilities for the fault tree.

⇨ Make sure nothing is selected within the fault tree so probabilities are calculated for the top event.

⇨ Select Fault Tree | Analysis | Calc Prob.

Now that we have defined time series, the Select Period dialog will appear as shown in Figure 8-4.

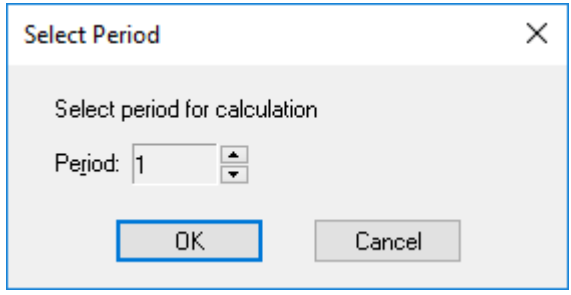

Figure 8-4. Select Period Dialog

DPL needs to know for which time period you'd like to generate results. We'll leave it at period 1 for now to make sure the result matches what we got before introducing time series.

⇨ Click OK.

The Calculation Complete dialog indicates that the probability is 0.0241 which is what you got before introducing time series. Now you'll calculate probabilities for another time period.

⇨ Make sure nothing is selected again and click Calc Probs.

⇨ Set the Period to 5 using the Period spin box arrows as shown in Figure 8-5.

⇨ Click OK.

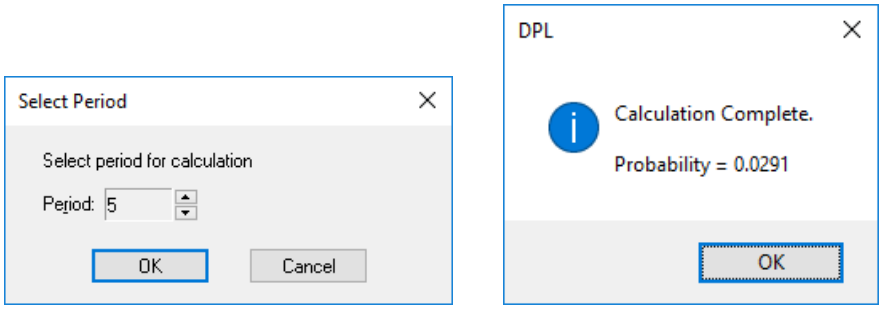

Figure 8-5. Select Period and Calculation Complete Dialog for Period 5

The likelihood of a water shortage occurring has increased from year 1 to year 5. You may visualize how the likelihood of the top event occurring changes over the 10-year time period within a single chart which is covered in the next section.

8.3 Time Series Percentiles

Now that time series intervals and nodes have been defined, there is a new output chart available: a Time Series Percentiles chart. This chart will display how the probability of the top event changes over the time interval defined.

⇨ Select Fault Tree | Analysis | Time Series.

The Time Series Percentiles set up dialog will appear. You'll accept the default setting.

⇨ Press OK to generate the chart. The Time Series Percentile chart will appear as shown in Figure 8-6.

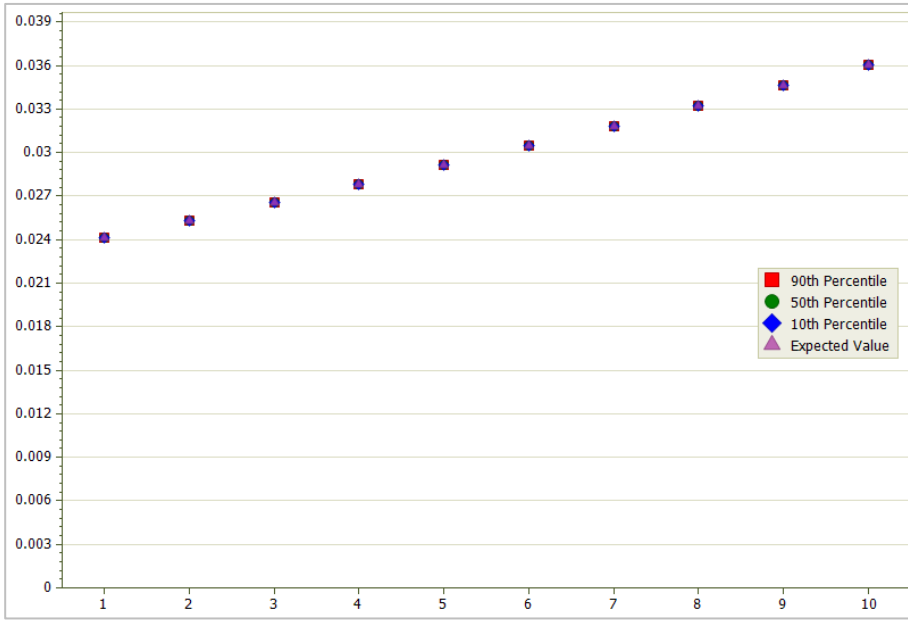

Figure 8-6. Time Series Percentiles Chart for Water Shortage Fault Tree

If you're familiar with the Time Series percentiles chart from a probabilistic decision analysis model, then this chart may seem a bit uninteresting in comparison. The fault tree is deterministic, so there isn't any difference between the values of the 10th, 50th, and 90th percentiles. It simply shows how the probability of the top event changes over the given time period. In this case the probability of a water shortage increases from about 0.024 to 0.036 over the 10 year period, given the increase in likelihood of the aqueducts failing due to age damage.

If the fault tree is incorporated into a decision model with uncertainty, the percentiles will often be different. However, if the fault tree is not embedded in a decision model or is embedded in a decision model with no uncertainty, then the percentiles in each time period are the same and are equal to the calculated value of the node.

If you were to switch to a circuit diagram view, you would need to specify a period via the Select Period dialog. If you were to generate Minimal Cut Sets or Partial Derivatives for the fault tree, you would be able to specify the time period for which you'd like to view those within their respective set up dialogs as shown in Figure 8-7.

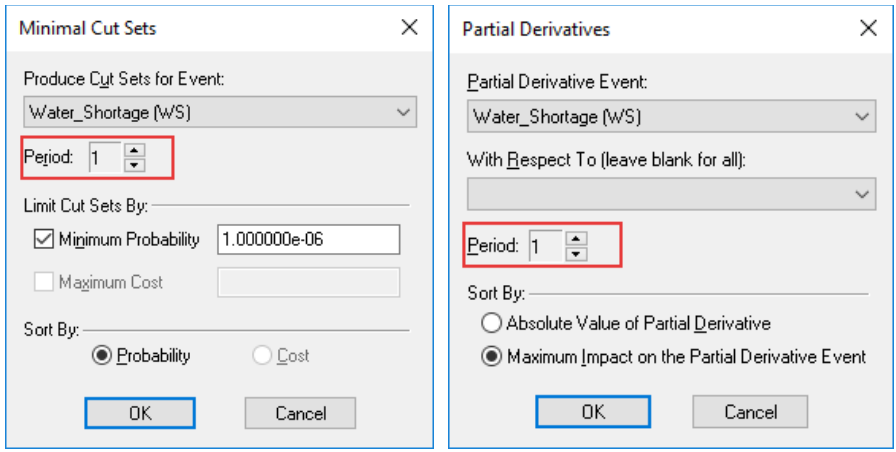

Figure 8-7. Minimal Cut Sets and Partial Derivatives Set Up Dialogs with Period Spin Box

9. Using Fault Tree Modules in Decision Models

The motivation for a fault tree analysis is normally some kind of decision. If we know how likely a system failure is, we have a better idea of how much to spend to prepare for the consequences. If we know the modes of failure (i.e., the cutsets), we can assess how much certain risk mitigation measures will help. Sometimes these decisions are simple enough that they can be considered implicitly. When they are not so simple, the natural next step is to embed the fault tree in a decision model and use influence diagrams and decision trees to model them explicitly. The best decision policy can then be determined by the analysis of a single, unified model.

Conversely, a decision analysis model is often found to be very sensitive to one or a small number of key probabilities. When those probabilities relate to a combination of events in a complex system, a fault tree may produce a more reliable probability estimate than a top down expert assessment.

To support both these situations, DPL fault trees can be easily embedded in DPL decision models.

9.1 Single Period Fault Tree Modules

In this section we will embed a fault tree module in a decision model and use its probability calculation as the probability for a chance node.

- ⇨ If necessary, open DPL Fault Tree.
- ⇨ Select File | Open.
- ⇨ Navigate to the Examples folder installed with DPL Fault Tree, usually C:\Program Files\Syncopation\DPL9FaultTree\Examples.
- ⇨ Select Secure_Facility_Decision_Model.da and open it.

The workspace contains two fault trees and two decision models.

- ⇨ In the Workspace Manager, double-click the Facility Access fault tree to activate it.

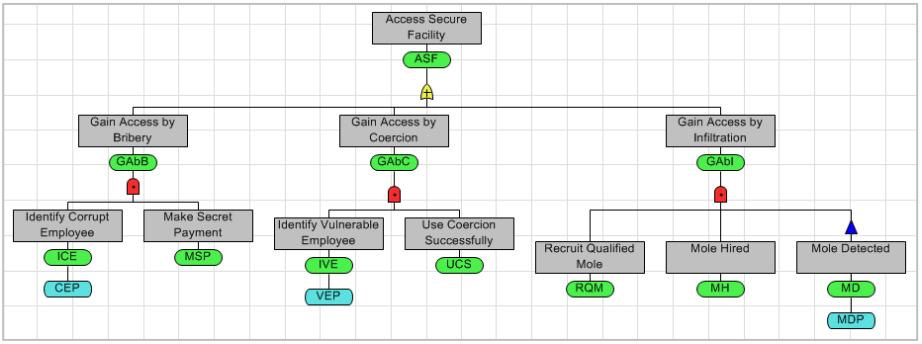

Figure 9-1. Access Secure Facility Fault Tree

This fault tree calculates the probability of an adversary gaining access to a secure facility by one of three methods: bribery, coercion or infiltration. It is similar to the fault tree analyzed in Chapter 6. However, this version is intended to be used as a module in another model. The three probability value nodes (short names CEP, VEP and MDP) don't have any data and thus serve as inputs to this fault tree when it's used as a module. If we tried to analyze this model standalone it would produce an error since it doesn't contain complete data.

We will now embed this fault tree in a decision model.

⇨ In the Workspace Manager, double-click the Defensive Actions model to activate it.

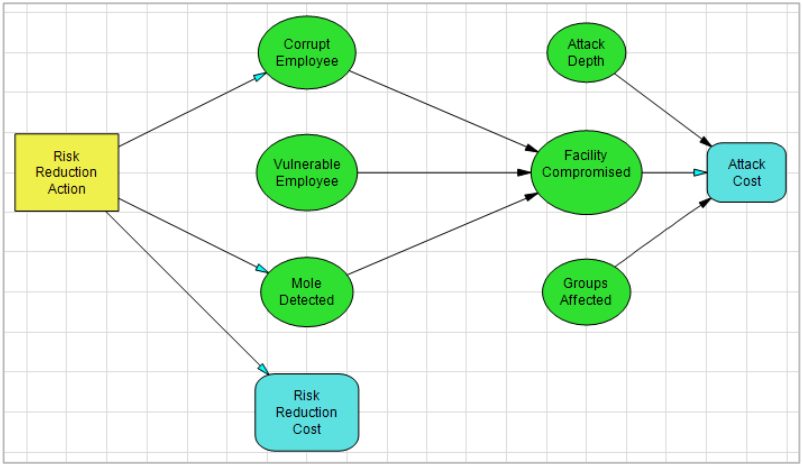

Figure 9-2. Defensive Actions Decision Model without Module

This decision model shown in Figure 9-2 considers two possible actions for reducing the risk of the facility being compromised, as well as an alternative for continuing the status quo. The probability for Facility Compromised can be better estimated using the fault tree in Figure 9-1.

⇨ Delete the three influence arcs into the Facility Compromised chance node.

⇨ Click Influence Diagram | Node | Add | Fault Tree Module. DPL displays the Select Module dialog (Figure 9-3).

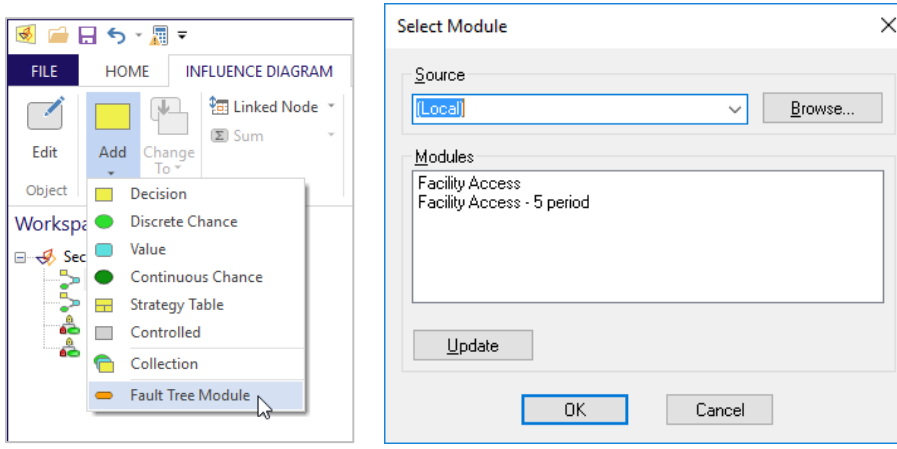

Figure 9-3. Influence Diagram | Node | Add | Fault Tree Module Command and the Select Module Dialog

⇨ Select Facility Access and click OK.

There will be a semi-transparent fault tree module node underneath your cursor ready to be placed.

⇨ Place the module node in the blank space within the middle of the influence diagram.

⇨ In the Node Definition dialog, name the module "Attack Successful" and click OK to close the dialog.

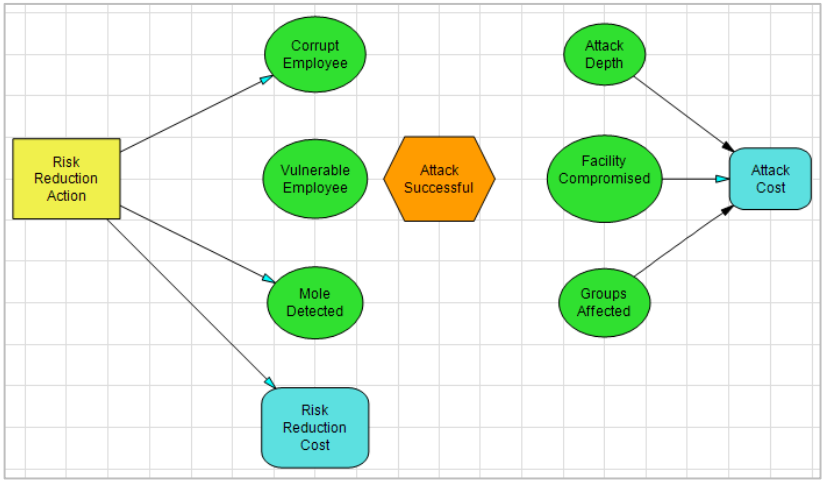

Figure 9-4. Defensive Actions Decision Model with Module

Next, we need to connect the module's three input values to nodes in the influence diagram. We do this by adding influence arcs.

⇨ Draw an influence arc from Corrupt Employee to Attack Successful. DPL displays the Select Module Input dialog (Figure 9-5).

⇨ Select Corrupt Employee Prob and click OK.

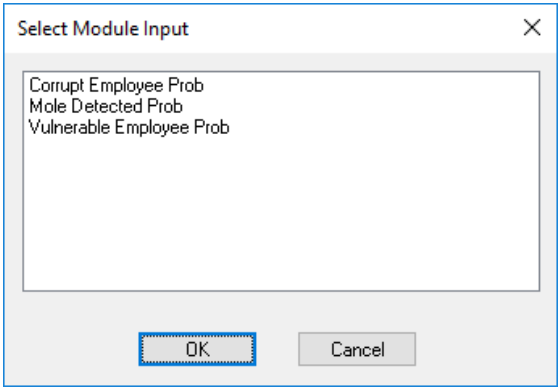

Figure 9-5. Select Module Input Dialog

⇨ Repeat this process for Vulnerable Employee (module input: Vulnerable Employee Prob) and Mode Detected (Mole Detected Prob).

⇨ Draw an arc from Attack Successful to Facility Compromised. Your fault tree should look like Figure 9-6.

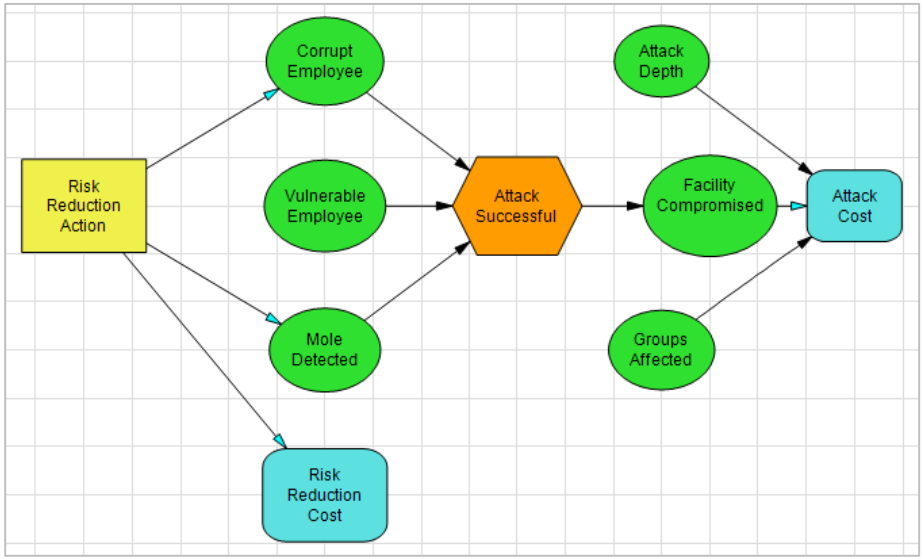

Figure 9-6. Defensive Actions Decision Model with Module and Connections

The connections made in the previous steps mean that the inputs to the fault tree depend on the outcomes of three chance nodes which are its predecessors in the influence diagram. The fault tree will be recalculated as necessary for each combination of event states of those nodes. While in a standalone fault tree the inputs (basic events and probability values) are deterministic, in this case the inputs to the fault tree are uncertain.

You still need to connect the output of the fault tree.

- ⇨ Double-click Facility Compromised node to edit it.
- ⇨ On the data tab, click the variable button to the right of the probability input box and select Attack_Successful from the Select Variable dialog.
- ⇨ Click OK to close the dialog.

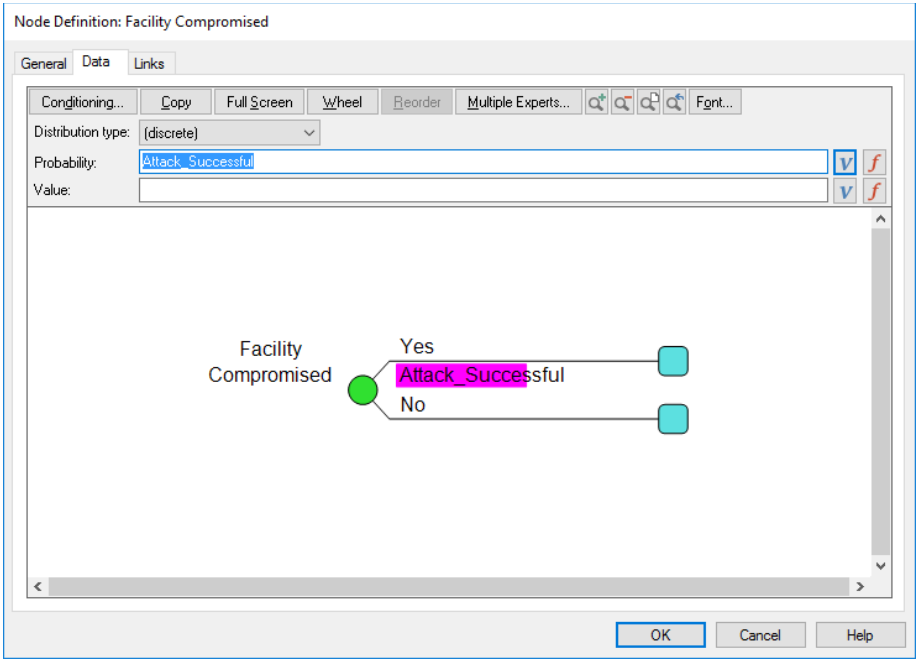

Figure 9-7. Node Definition Dialog for Facility Compromised

The decision model with the embedded fault tree module is now ready to run.

⇨ In the Home tab of the ribbon, make sure Risk Profile and Init Dec Alts are checked. Uncheck Policy Tree.

⇨ Click Home | Run | Decision Analysis to generate the outputs.

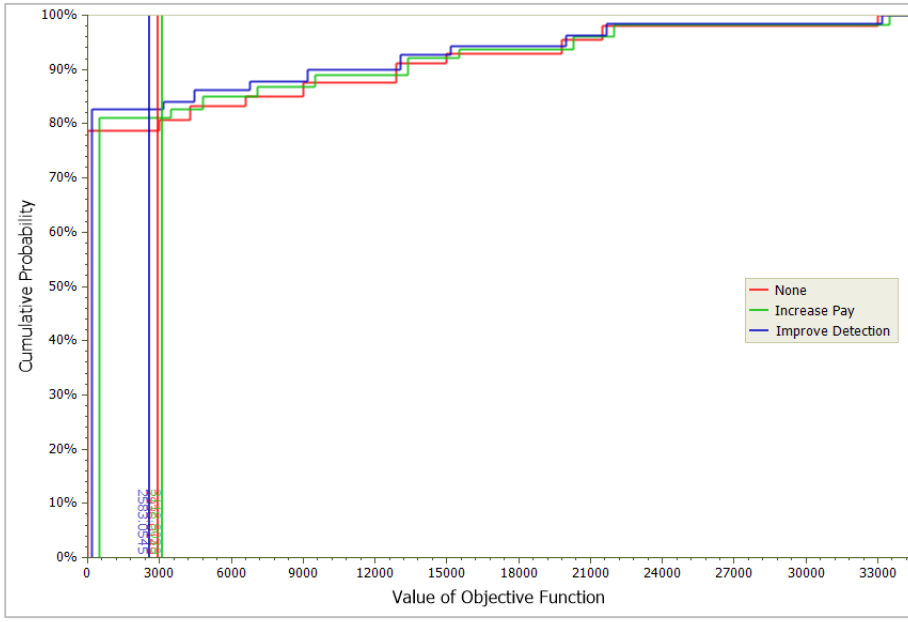

Figure 9-8. Initial Decision Alternatives Risk Profile Showing Alternative Defensive Actions

The Improve Detection alternative (shown in blue) is the least costly way to defend against an attack.

9.2 Time Series Fault Tree Modules

In the previous section we embedded a scalar fault tree module in a single period, single attribute decision model. In many practical settings the risks of interest are uncertain and evolve over time. In this section we'll embed a vector valued fault tree in a multi-period decision model.

⇨ In the Workspace Manager, double-click the Facility Access - 5 period fault tree to activate it.

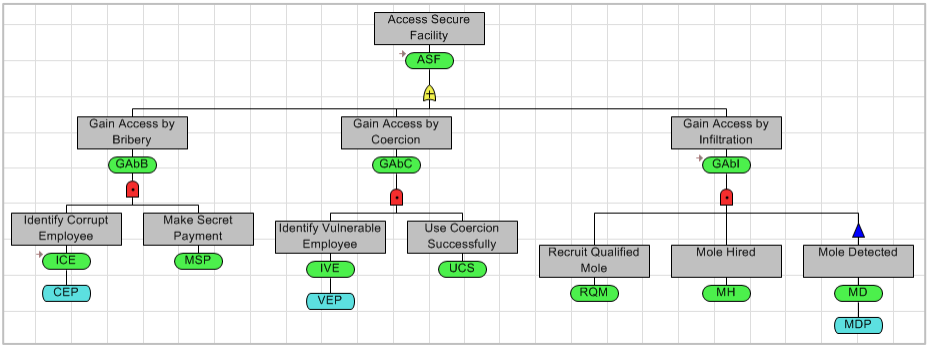

Figure 9-9. Access Secure Facility Fault Tree with 5 Time Periods

Note that some of the binary events have the gray arrow indicating that they are vector valued.

⇨ Click Fault Tree | Alter | Define Intervals.

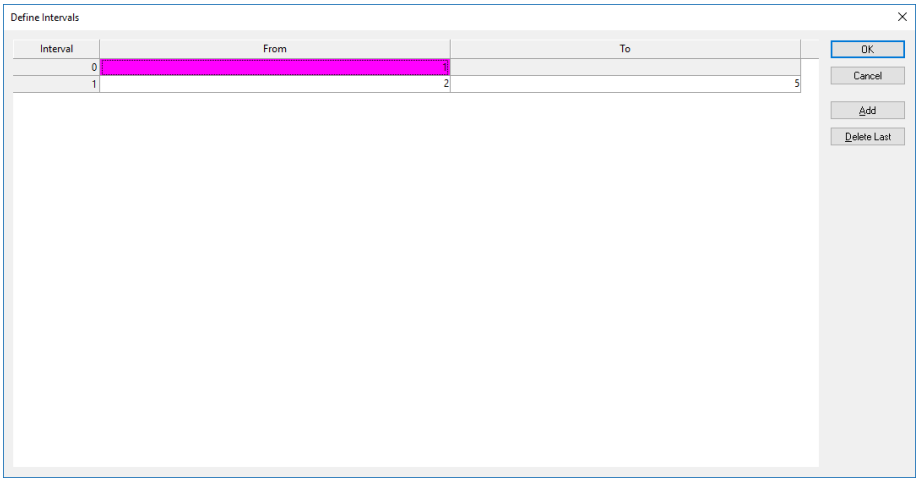

Figure 9-10. Define Intervals Dialog

As seen in Figure 9-10, this fault tree has two intervals which together cover five periods, numbered 1 to 5. The output of this fault tree is a series with five values, each one a probability of the top level event in the applicable period.

⇨ Click OK to close the dialog.

⇨ In the Workspace Manager, double-click the Defensive Actions - Time Series decision model to activate it.

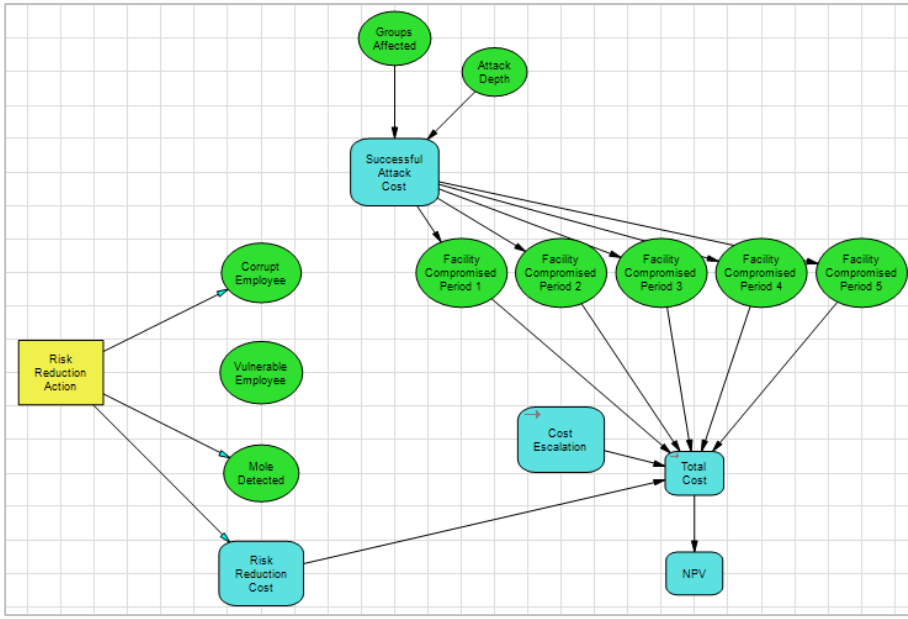

Figure 9-11. Defensive Actions Time Series Model without Module

This model has a chance node for the risk of the facility being compromised in each of 5 time periods. You will use the 5 period version of the fault tree module to calculate the relevant probabilities.

- ⇨ Click Influence Diagram | Node | Add | Fault Tree Module.
- ⇨ In the Select Module dialog, choose "Facility Access - 5 period".
- ⇨ Place the module node in the blank space within the center of the influence diagram.
- ⇨ Name the module "Attack Successful".
- ⇨ Draw influence arcs from Corrupt Employee, Vulnerable Employee and Mole Detected to Attack Successful, in each case selecting the appropriate module input as in the previous section (Figure 9-5).
- ⇨ Draw influence arcs from Attack Successful to each of the five Facility Compromised Period N nodes.

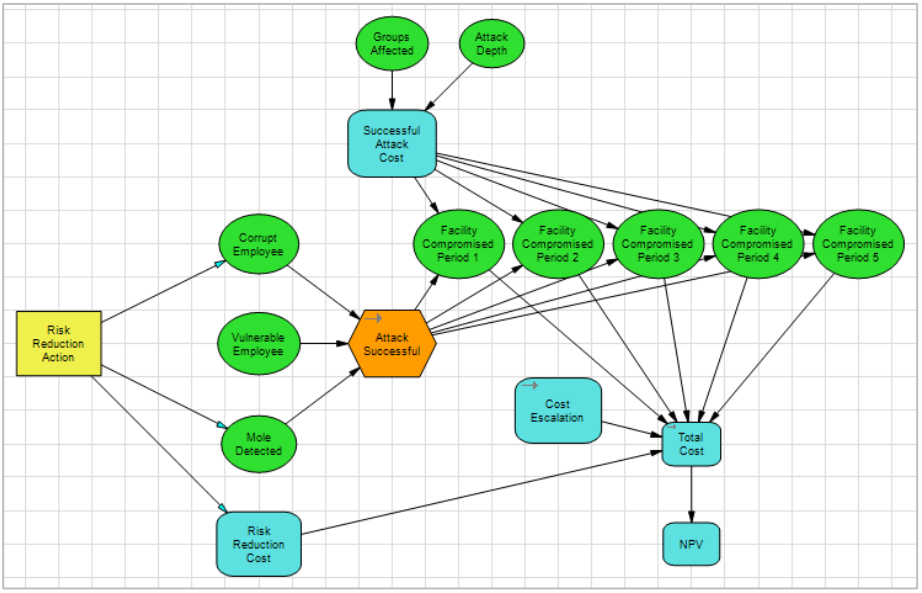

Figure 9-12. Defensive Actions Time Series Model with Module

The Attack Successful module node is a series with 5 elements. We need to replace the constant probabilities in the Facility Compromised chance nodes with values from that series.

- ⇨ Double-click the Facility Compromised Period 1 chance node.
- ⇨ In the Node Definition dialog, press the variable button to the right of the probability input box and select Attack_Successful.
- ⇨ Add a subscript of "[1]" to Attack_Successful.
- ⇨ Click OK to close the dialog.

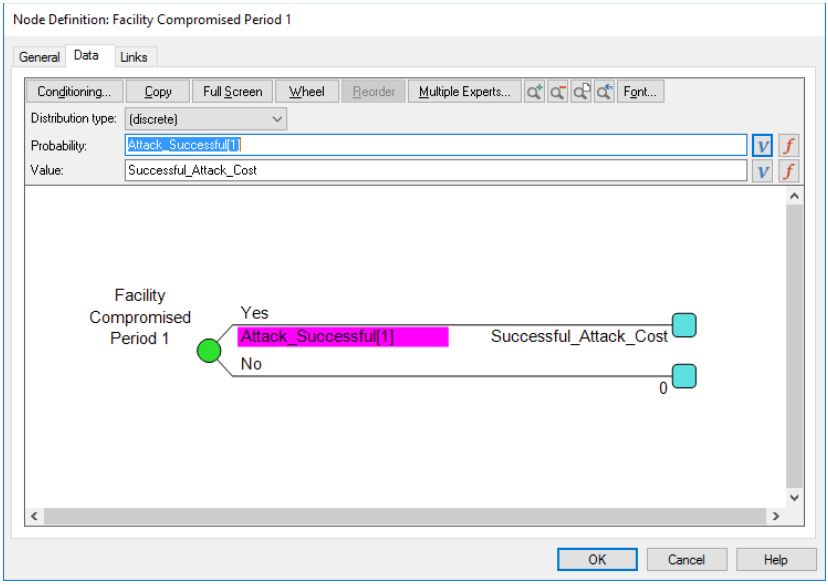

Figure 9-13. Probability with Subscript

⇨ Repeat the process for the other four Facility Compromised nodes, using subscripts [2] through [5].

Now the model is complete and ready for analysis. Since the model contains multiple time periods, you will generate Time Series Percentiles chart.

⇨ Click Home | Sensitivity | Time Series. DPL displays the Time Series Percentiles dialog (Figure 9-14).

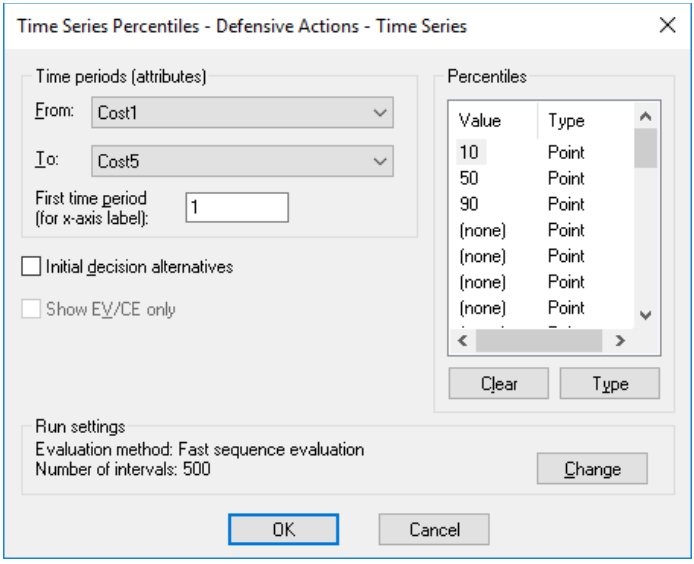

Figure 9-14. Time Series Percentiles Setup Dialog

⇨ Click OK to accept the defaults.

DPL runs an analysis and displays a Time Series Percentiles chart as shown in Figure 9-15.

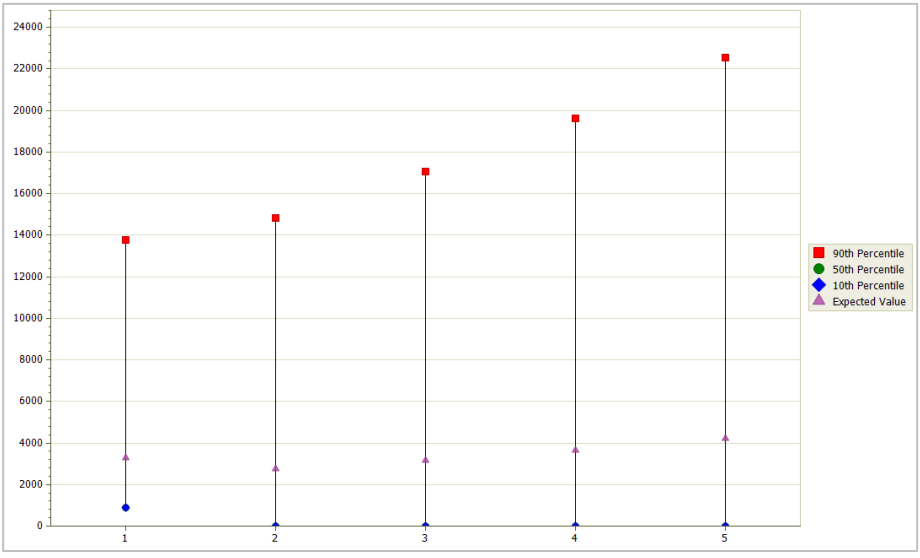

Figure 9-15. Time Series Percentiles Chart for Secure Facility Decision Model

For each of the five time periods, this chart shows the range of possible values for total cost in that period. Contrast this chart with Figure 8-6 in the prior chapter, where the model was a deterministic fault tree and thus there was only a single possible value for each time period.

Index