

# **FAULT TREE CREATION AND ANALYSIS TOOL**

**USER MANUAL**

<b>1. INTRODUCTION</b>	<b>4</b>
1.1. WHAT IS A FAULT TREE?	4
1.2 WHAT DOES “FAULTCAT” DO?	5
<b>2. INSTALLATION</b>	<b>6</b>
2.1 HOW TO INSTALL THE APPLICATION	6
2.1.1 <i>Windows</i>	6
2.1.2 <i>Linux</i>	6
2.2 HOW TO SET UP DATABASE SUPPORT	7
2.2 HOW TO SETUP XML SUPPORT	7
<b>3. LAYOUT</b>	<b>8</b>
<b>4. CONTROLS</b>	<b>9</b>
4.1 MENU	9
4.1.1 <i>File</i>	9
4.1.2 <i>Options</i>	9
4.1.3 <i>About</i>	9
4.2 GENERAL TOOLBAR	9
4.3 FAULT TREE TOOLBAR	10
<b>5. HOW TO USE THE FAULTCAT APPLICATION</b>	<b>12</b>
5.1 DRAWING A FAULT TREE	12
5.1.2 <i>Wires.</i>	12
5.1.3 <i>Setting title, value and information.</i>	12
5.1.4 <i>Deleting</i>	12
5.2 SAVING A FAULT TREE TO A XML FILE	13
5.3 OPENING A XML FILE	13
5.4 USING THE DATABASE	14
5.4.1 <i>Database preferences</i>	14
5.4.2 <i>Create a new project</i>	14
5.4.3 <i>Edit the title and author</i>	14
5.4.4 <i>Saving to Database</i>	14
5.4.5 <i>Loading from the Database</i>	14
5.4.6 <i>Deleting a project</i>	14
5.5 CREATE A NEW FAULT TREE	14
5.6 ZOOMING IN AND OUT	15
5.7 ARRANGE THE FAULT TREE	15
5.8 GET A REPORT WITH ALL THE OBJECTS IN THE FAULT TREE	15
5.9 SEE ALL THE CALCULATIONS DONE IN THE FAULT TREE	15
5.10 EXIT THE PROGRAM	15
<b>6. IMPLEMENTED RULES</b>	<b>15</b>
6.1 HARD RULES	16
6.2 SOFT RULES	16
<b>7. APPENDIX: SAMPLE FAULT TREES</b>	<b>17</b>
7.1 EXAMPLE 1 -	17

7.2 EXAMPLE 2 -	21
<b>8. GLOSSARY</b>	<b>25</b>
<b>9. BIBLIOGRAPHY</b>	<b>26</b>

# 1. Introduction

Welcome to the user-manual of the “Fault Tree Creation and Analysis Tool”, hereafter called “FaultCAT”.

The “FaultCAT” is an easy to use program, that will let you build and manage most kinds of fault trees quickly, and then let you do a series of calculations to find the information that you’re seeking. Not only will this manual show you how to properly operate the “FaultCAT” application, but it will also help you to understand what fault trees are and what rules governs them.

The “FaultCAT” application lets you have full control in placing and managing the fault trees as you see fit. You can add the information you want to the components, and the fault trees will always show you the probabilities for each component.

And to make sure that others can easily receive a copy of your fault trees for viewing or further development, the “FaultCAT” lets you save your fault trees in either a database or as a handy XML file.

## 1.1. What is a Fault Tree?

Simply put, a fault tree is a graphical representation of an analytical technique in which we are trying to find all credible ways of how an undesired event can occur.

These trees are often used to see what different parts of a system can contribute to make a component fail, for example how an electrical component can short-circuit in an electrical system. In computer security, these trees are often called attack trees, showing how easy (or hard) it can be to attack/break into something using various approaches.

The top event of the fault tree is the undesired event that can happen, whilst the different paths below it are events that have a probability of inducing this undesired event and gates that serve to permit or inhibit the passage of fault logic up the tree.

It is important to understand that a fault tree is not a model of all possible system failures or all possible causes for system failure. A fault tree is tailored to its undesired event, and thus it only includes events that can lead to this top event.

Below is a summary of the different events that fault tree can be composed of. Note, however, that these are not all the events that exist, but only those commonly used and which are included in the “FaultCAT” application.

Basic Event: This event looks like a circle and this is the most used event in a fault tree. This object is used to describe an event that requires no further development and that can lead to the top-level fault, and the probability for it.

Undeveloped Event: This event looks like a diamond. This event follows the same rules as the Basic event. It is used to describe an event that is not further developed either because it is of insufficient consequence or because information is unavailable.

External Event: This event looks like a house. This event follows the same rules as the Basic event. It is used to describe an event that is expected to have an impact on the system/area that you're analysing, but which is outside this.

Intermediate Event: This event looks like a rectangle. This is a fault event that occurs because antecedent causes acting through logic gates. These are used to describe the "new" fault after a logic gate, and are also the top-level events in a fault tree.

OR Gate: The OR Gate is used to show what event happens and the probability for it, if one or more of the events below it occur. There may be any number of inputs to an OR Gate.

AND Gate: The AND Gate is used to show what event happens and the probability for it, if all of the events below it occur. There may be any number of inputs to an AND Gate.

For more information regarding the events that a fault tree can be composed of, see the Controls section.

## 1.2 What does "FaultCAT" do?

- Lets you create any kind of fault tree you want from a list of finished components.
- Lets you place your components and structure the trees how YOU want them to be.
- Lets you write specific notes for each component so that you more easily can remember what each component is.
- Automatically calculates and shows the probability for each component in the tree.
- Lets you easily save your created trees to a database or as a XML file.
- Lets you see how each value/ probability is calculated.
- Shows the biggest probability path(s) in a tree.

## 2. Installation

### 2.1 How to install the application

Installation of the “FaultCAT” application is very simple. All the files and images comes packed in a JAR file which can be executed without having to unpack or install anything.

The only prerequisite is that you have the Java Runtime Environment (JRE) installed on your machine.

If you do not have the JRE installed you can download the latest version at Sun’s website:

<http://www.java.sun.com> .

Once you’ve placed the “Faultcat.jar” file in a directory of your choosing, you can execute the file.

#### 2.1.1 Windows

If you’re using Windows there are two ways you can execute the JAR file. If you’re using a shell window, you simply type

```
java -jar "Faultcat.jar"
```

in the directory where the JAR file is located. The application should start up and let you start designing fault trees.

Another way to execute the program if you’re using the regular windows environment is to create a shortcut in the directory where the JAR file is, or in another place if you prefer – the way to do create the shortcut is still the same.

Once the shortcut has been created, you edit it and enter in the “Target” the path for where the java executable is, and then the same as was written in the shell window.

Finally you write in the “Start In” field the path for where the JAR file is located.

Example, lets say you have installed the JAR file in the directory C:\Program Files\FaultCat\, and the Java executable is located in C:\Program Files\Java\2sdk1.4.0\_01\jre\bin\, then to create a working shortcut I enter the following in the “Target” and “Start In” fields respectively

```
C:\Program Files\Java\2sdk1.4.0_01\jre\bin\java.exe -jar Faultcat.jar
```

```
C:\Program Files\FaultCat\
```

Once that has been done, everything should be set to go and you can start creating those fault trees.

#### 2.1.2 Linux

If you’re using Linux, then the best option is to start a shell window and go to the path where the JAR file is located. Then you simply type

```
java -jar Faultcat.jar
```

If the Java executable is not in your \$PATH variable, you need to write the full directory for where the executable is installed to get it to work. After that the program should be running without any problems.

## 2.2 How to set up database support

The “FaultCAT” application does not come with a database, but it does provide support for MySQL databases. This will let you create projects that you can save to the database or open a project and place the contents onto the “FaultCAT” drawing area.

Assuming that you already have a MySQL database up and running, then all you need to do is to start the application and click on the database icon. If no prior database configuration exists, it will open a configuration menu.



**The database configuration screen.**

There you can enter your username and password (if applicable) that you have on the database, and also the path for the location of the database. If everything has been entered correctly a new window will be created where you can create new projects to enter to the database, or to retrieve already saved projects from the database.

## 2.2 How to setup XML support

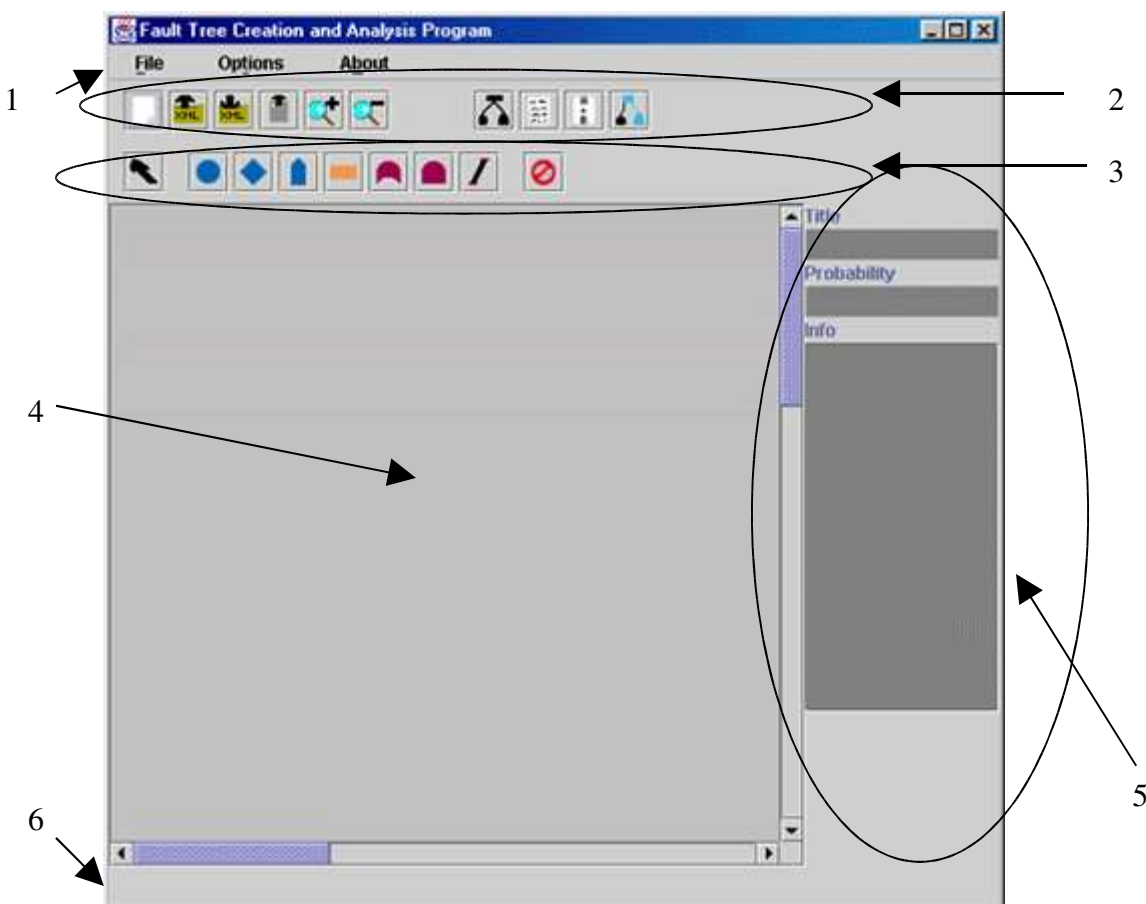
Setting up XML support involves no work for you. The “FaultCAT” application takes care of everything since it comes supplied with the ability to save and load XML files.

All that is required of you is to select which file to open, or where to save a new XML file.

### 3. Layout

Below (diagram 1) is a picture of the “FaultCAT” GUI. This is the only screen that is needed to create and manipulate fault trees. A brief explanation of the different areas of the application is included here.

- The menu bar. The menu includes options like saving and loading fault trees, options that will affect the application in various ways, and information about the developers.
- The general toolbar. This toolbar shows the most used file operations, and also include buttons for zooming in and out of the drawing area, structuring the FaultCAT tree, as well some common reports.
- The fault tree toolbar. This toolbar shows the buttons for creating fault tree components that will be placed on the drawing area.
- The drawing area. This is the area where the fault tree is created. All objects are placed here.
- The event detail area. This area lets you enter details on the different fault tree events that you have placed on the screen and selected. Some events have more details than others, and those details that can't be manipulated are grayed out.
- The information line. This line shows some additional information about the different parts of the application, based on where you hold your mouse cursor



## 4. Controls

This section describes in detail what the menu bar and the toolbars do.

### 4.1 Menu

The menu bar includes the most used file manipulations and options that will change how you build the fault trees. The menu bar has the following menus.

#### 4.1.1 File

- New - This option clears the drawing area and lets you start making a new fault tree. If there already are components on the drawing area, you will be asked whether you want to save it or not.
- Open XML file – Lets you load a previously saved XML file.
- Open Database – Lets you open a project previously saved to the Database
- Save XML file – Lets you save your FaultCAT tree to an XML file
- Save to Database - Lets you Save your FaultCAT project to the database
- Exit – Quits the program. If you've got components on the drawing area, you will be asked whether you want to save or not.

#### 4.1.2 Options

Hold Component – If this checkbox is checked, then the fault tree toolbar will not revert back to the pointer after you've placed a component on the drawing area. Instead it will keep your last selection.

Gate-Inter – If this checkbox is checked, then every time you place an OR Gate or an AND Gate on the drawing area, an Intermediate Event with a wire to the gate will be placed on the screen too.

OR-Gate children  $\leq 1$  total. OR-gate shows in pink if the sum of its children is higher than 1, and returns 0.

#### 4.1.3 About

- The Crew – That's us -the developers.

### 4.2 General Toolbar

This toolbar lets the user do various operations that are not tied to fault tree creation. These operations include saving and zooming, and clearing the drawing area to let you start anew.



New: This clears the drawing area and lets you start making a new fault tree. If there are components on the drawing area, you will be asked whether you want to save or not.



Load: This lets you load a previously saved fault tree, saved as an XML file.

Also see Section 5.3



Save: This lets you save a fault tree as an XML file.

Also see section 5.2



Load from database: Lets you Load a previously saved Fault Tree from your database. Also see section 5.4

Zoom In: This lets you zoom in on the drawing area so that you can see parts of a large fault-tree in better detail.

Hot-key: Alt + Z

Zoom out: This lets you zoom out from the drawing area so that you more easily can see the whole of a large fault tree.

Hot-key: Alt + X



Arrange Tree: Pressing this button will arrange the the Fault Tree you are working on in a nice tree structure

HotKey: Alt + C

Report: This creates a new window that displays every component, its title, info and the probability that are contained in each component. See also section 5.9

Hot-key: Alt + I

Show Calculation: This creates a new window that displays the calculations made by each AND Gate and OR Gate. See Also Section 5.10

Hot-Key: Alt + O



Show Path: This button shows the path(s) with the highest probability. The path is coloured light green. AND Gates will make the path split up to all the fault tree components below, whilst the OR Gate will choose the path with the highest probability. If an OR Gate has several paths with the same probability, it will choose all of those.

Hot-key: Alt + P

### 4.3 Fault tree Toolbar

This is the toolbar that lets you create and delete the different fault tree events. The buttons are not only separated, but are logically colour-coded as well, to help you quickly find the events that you need. All these buttons will stay selected after pressed until you click on the drawing area, after which the Pointer button will be selected.

Note: See the Menu section for how to avoid this.



Pointer: This is the default selection. The pointer is used to manipulate already created events, such as selecting them or moving them on the screen.

Hot-key: Alt + 1



Delete: This is button with the red sign on it. When pressed, the next event you click on will be deleted from the drawing area. All wires attached to this event are also deleted.

Hot-key: I

Primary Events: The first 3 buttons that are colour coded blue are termed Primary Events. These objects are more or less similar in how they behave. It is their graphical representation that will tell the user what the difference is.

Basic one Event: Creates a circle on the drawing area where the user clicks. This can have one wire attached to it that is connected to a higher-level event attached to it.  
Hot-key: Alt + Q

Undeveloped Event: Creates a diamond on the drawing area where the user clicks. This can have one wire attached to it that is connected to a higher-level event attached to it.  
Hot-key: Alt + W

External event: Creates a “house” on the drawing area where the user clicks. This can have one wire attached to it that is connected to a higher-level event attached to it.  
Hot-key: Alt + E


Gates: The two buttons that are colour-coded red are termed Gates. These are the components that do the calculations from the lower-level events that are attached to them.

AND Gate: Creates a “dome” on the drawing area where the user clicks. This can have one wire to a higher-level event and an infinite number of wires to lower-level events.  
Hot-key: Alt + S

OR Gate: Creates a “dome with domed floor” on the drawing area where the user clicks. This can have one wire to a higher-level event and an infinite number of wires to lower-level events.  
Hot-key: Alt + D

Other: We’ve listed the other fault tree events here. This is the orange colour-coded Intermediate Event, which is quite important in the creation of fault trees, and the green colour-coded Transfer In object.

Intermediate Event: Creates a rectangle on the drawing area where the user clicks. This can have one wire attached to it that is connected to a higher-level event, and one wire that is attached to a lower-level event.  
Hot-key: Alt + A

Wire:  Creates a line between two events. When pressed the next two valid events that you click on will receive a wire between them. Please refer to the Implemented Rules chapter to see which components are valid.  
Hot-key: Alt + 2

## 5. How to use the FaultCAT application

This section is a simple guide on how to use the program. For a more comprehensive guide on how to design a fault tree please refer to section 7 and the bibliography in section 9

### 5.1 Drawing a Fault Tree

This will be a simple guide on how to draw a fault tree. In section 3 and 4 we have explained the layout and the different menu bars. This section is to explain how to use everything.

#### 5.1.1 To draw a Fault Tree.

To draw a fault tree you use the icons on the Fault Tree Toolbar.  
(See section 4.3) To draw an AND gate, you click on the AND gate icon, and click on the drawing area. This makes an and gate marked red appear where you clicked. It is the same for all the events and gates. If you click anywhere on the drawingarea the and gate should turn white. If you click on it again it should turn red again, it is now selected.  
To drag it around you can click on it with left mouse button, hold the button and drag it where it suits you. To release it, release the left mouse button. This applies for all the gates and events.

#### 5.1.2 Wires.

To connect two elements, click on the intermediate event icon, and then click on the drawingarea. This makes an intermediate event. Make sure the intermediate event is placed above the and gate, (drawn in section 5.1.1) To connect them, click the wire icon, then click on the intermediate event and finish with clicking on the and gate. This draws a wire between them. You can only draw wires between certain events. See section 6.1 for more details.

#### 5.1.3 Setting title, value and information.

It is possible to give all events a title, and write some information about the event. This is done in Fault Tree Details area on the right of the screen. (See section 3.) Primary events can also be given a value. If no event is selected, the text fields in the details area will be grey.

To set a title, select an event, click on the title field, write the title you wish, then click on the drawingarea. The title will now appear next to the event. To write information about an event, you do the same as for title. You can review the information anytime you select the event.

To set a value, select a primary event, a basic-, undeveloped- or external event.

The value field is now white, click on the value field, enter a value below 1, separated by a dot, not a comma. When the value has been entered, click the drawingarea. The value will appear in the middle of the event.

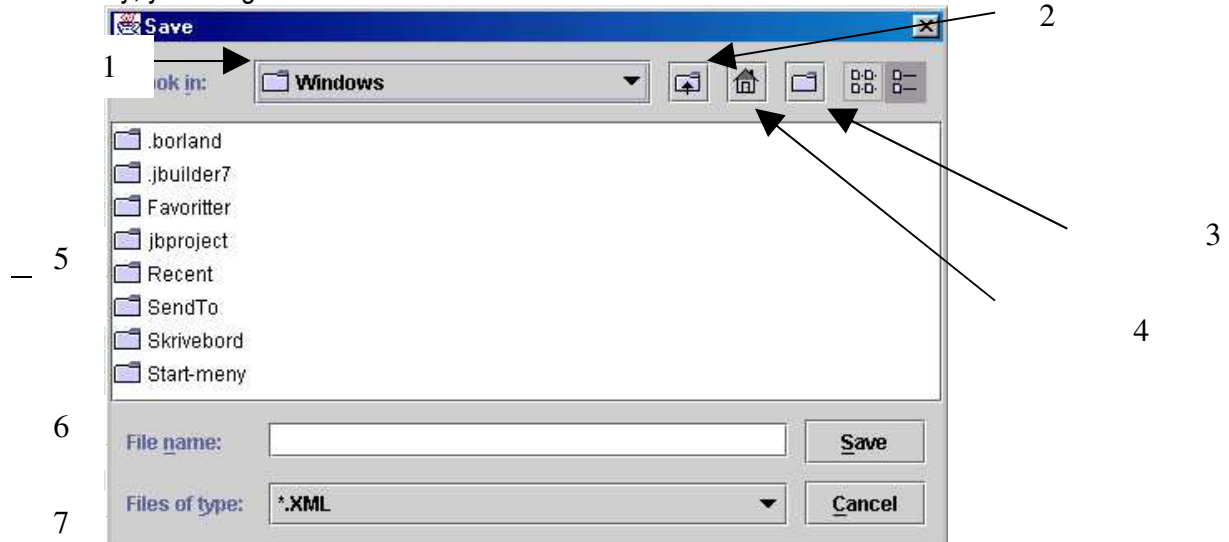
#### 5.1.4 Deleting

To delete anything drawn on the drawing area, simply click on the icon that looks like a stop sign, and then click the object you want to delete.

## 5.2 Saving a Fault Tree to a XML file

To save a Fault Tree to an XML file, there are two ways to go. The first is in the browser menu. Select file, and save XML file(See section 4.1.1.) The other way is to simply click the icon, save to XML in the general toolbar.( See section 4.2)

Either way, you will get this screen:



- Here you can select another drive, if you XML is located somewhere else than your home directory
- Press this one to move up a level
- This button takes you to your home directory
- To create a new folder, press this button. The folder will appear in the main window under the name New Folder. You must manually rename it.
- This is the main window.
- Here you can enter the filename of your file, or press a file, and the name will appear in this text field.
- To view all files press the drop-down menu shown here, and select All files. By default you can only view files with the XML extension.

To save a file you enter the filename in the text field, and press save. If you need to exit press cancel.

## 5.3 Opening a XML File

The open a XML file option is the same as the save option. There are two ways, from the browser menu, file Open XML file(Section 4.1.1), and general toolbar(Section 4.2) the save to XML icon. You will get the same screen. The difference is the opening button. To open a file, you can either locate the file and press the left mouse button twice on the file of your choosing. You can press the filename once, it will then appear in the text field. And press the open button, or you can type in the filename directly, and press enter or the open button.

## 5.4 Using the Database

You can use a database to save and load Fault Trees. If you want to do this you press the Use Database on the fault Tree button on the general toolbar or you can select it from the menu bar **File** **Use DB**.

### 5.4.1 Database preferences

If it is the first time you use this option, you will be shown this window



Here you have to give the username, password and path for your mySQL database.

When you have done this you will be shown this window

**<bilde her>**

### 5.4.2 Create a new project

To create a new project, press the new button. A new row will be visible in the main window.

### 5.4.3 Edit the title and author

You can edit the title and author of any project in the database. Just click twice on the cell you wish to edit - a marker should now be visible in the selected cell. Enter text of your choosing, and press enter.

### 5.4.4 Saving to Database

To save to the database, select a row in the main window. It should be highlighted with blue colour. And press the save button.

### 5.4.5 Loading from the Database

To load from the database, select a row, and press the load button.

### 5.4.6 Deleting a project

To delete a project you do the same as in the two paragraphs over. Select the row you wish to delete, and press the delete button

To exit the database window without doing anything press cancel.

## 5.5 Create a new Fault Tree

If you need to start over, with an empty drawing area, press the new icon on the general toolbar, or select **file** **New**, on the browser menu. You will then get a small pop up screen, asking you if you want to save your current tree, the YES button. It will be saved to a XML file. Or to just empty the screen, press the NO button, or to cancel, press the CANCEL button.

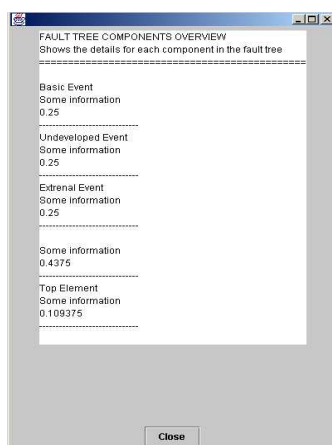
## 5.6 Zooming in and out

To Zoom in, press the Zoom inn button, to zoom out you press the zoom out button. You find both on the general toolbar.

## 5.7 Arrange the Fault Tree

If you feel your Fault Tree is chaotic, you can press the arrange fault tree button on the general toolbar (section 4.2). This will organize your tree in a nice tree structure. It will make sure that all fault tree components in the tree will be not overlap and keep a respectable distance from each other.

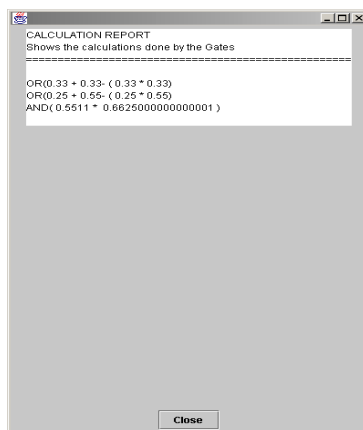
## 5.8 Get a report with all the objects in the Fault Tree



You can get a report that shows all the objects that have a probability and info field on the drawing area. Click on the report icon on the general toolbar, and it will show you a window which should look like the figure shown beside.. The components that will be shown are the Basic Events, External Events, Undeveloped Events and Intermediate Events. Gates are not shown since they do not have their own probability or information field.

When done viewing the report, press the close button on the bottom of the screen.

## 5.9 See all the calculations done in the Fault Tree



It is also possible to view all the calculations done by the gates in a Fault Tree. To do this press the calculations button on the general toolbar. The window you get will be like shown to the left. It shows each Gate and how it calculates the probability from each of its children).

When done press the close button on the bottom of the screen.

## 5.10 Exit the program

To exit the program, you select file **Quit** in the browser menu. Before you quit the program will ask you if you want to save your work. It will be saved by default to an XML file

## 6. Implemented Rules

This section describes the rules that the “FaultCAT” application uses to create fault trees that give meaning. Thus, it shows which events can be combined together and why. The first section shows the

rules that the program uses and which you have to conform to. The second section shows the rules you *should* strive to act on, to make better fault trees.

## 6.1 Hard Rules

No gate-to-gate connection: You are not allowed to put two or more gates together without any other type of event between. This is a signal of a fault tree that has not been developed enough, and there should always be an Intermediate event in between to describe what happens.

## 6.2 Soft Rules

Define inputs: Make sure that all events to a gate are fully explored before moving on. This will help ensure that each gate has the correct number of child components at all times, since its easy to forget something when your fault trees grow big.

Width before Height: You should make sure that you finish each level in a fault tree before you move down to the next components.

In the table showing all saved project. If it's marked as reserved, someone else is working on that project. Do not save or load from here.

## 7. Appendix: Sample Fault Trees

This appendix shows a few sample fault trees to help you better understand how they're built based on a given assignment. The first example is more of a step-by-step in how to use the program, and lets you see each stage as the fault tree is built. The second example shows only the process of making a fault tree. Suffice to say, it's beyond the scope of this manual to explain the whole process behind making fault trees, and the reader is encouraged to read the books in the bibliography section.

### 7.1 Example 1 -

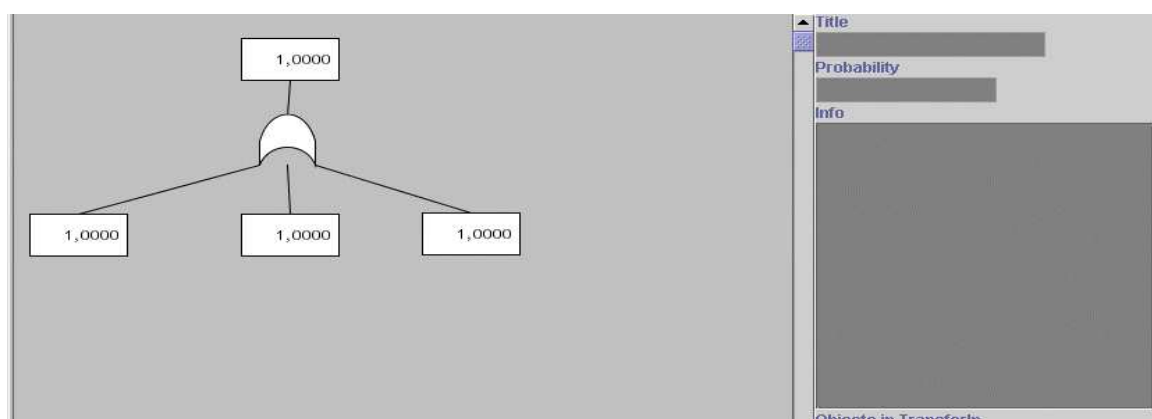
*You're responsible for your company's web server. You have to make sure that the server is up and running at all times. It has to be safeguarded against both mechanical and software failures as well as wrongful human interaction (wilfully or not). You decide that an analysis of the web server and the components that interact with it is in order.*

(Note: Proper design would have you set outer limits on the area of analysis and how detailed you should go).

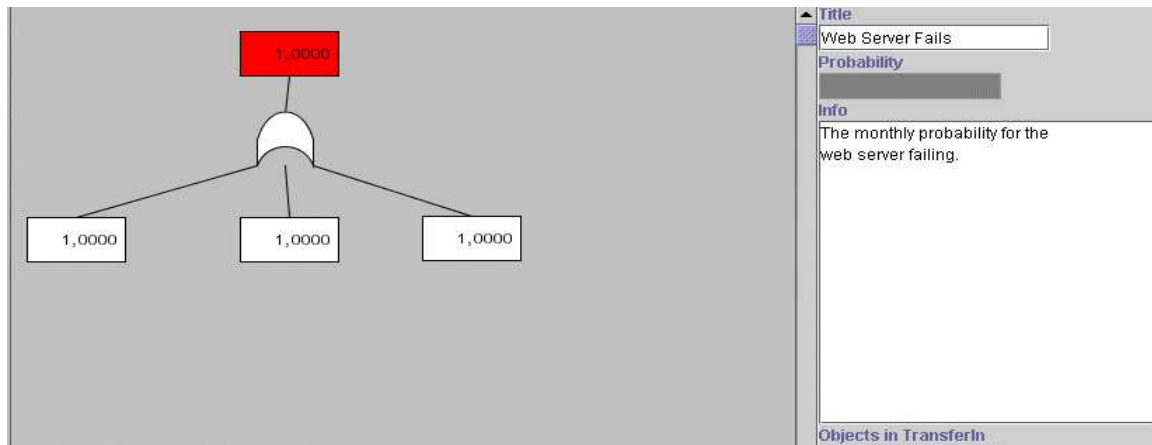
We set the top level event as "Web server fails/ per month", and then find the first events that can result in this. We see that if the server itself fails (mechanical failure), then there is little else to do. This also applies to software failure. The last thing we can think of that can apply is the ISP that we're connected might experience problems that will affect us.

We see that just one of these faults is enough to make our web server fail, and that these faults can be further developed. So we put an OR gate in between the top level event and the intermediate events.

We click on the Intermediate Event button, and place four of these events on the drawing area (If you select Hold Component, you can place all four events at the same time without having to click on the Intermediate Event button each time). We then click on the OR Gate button and place the gate on the drawing area. Then we select the Wire, and attach the wires between the gate and the Intermediate Events. Finally we select each of the Intermediate Events on the drawing area by clicking on them with the mouse pointer. The Title and Info areas next to the drawing area, should now be editable. Write in the caption for each event, and any additional information you think might be necessary.

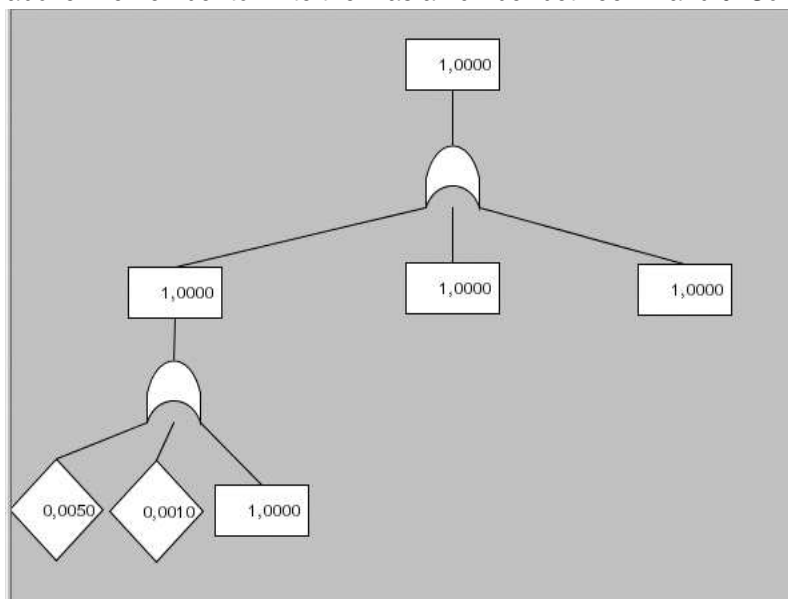


**The first fault tree components have been placed out on the screen.**



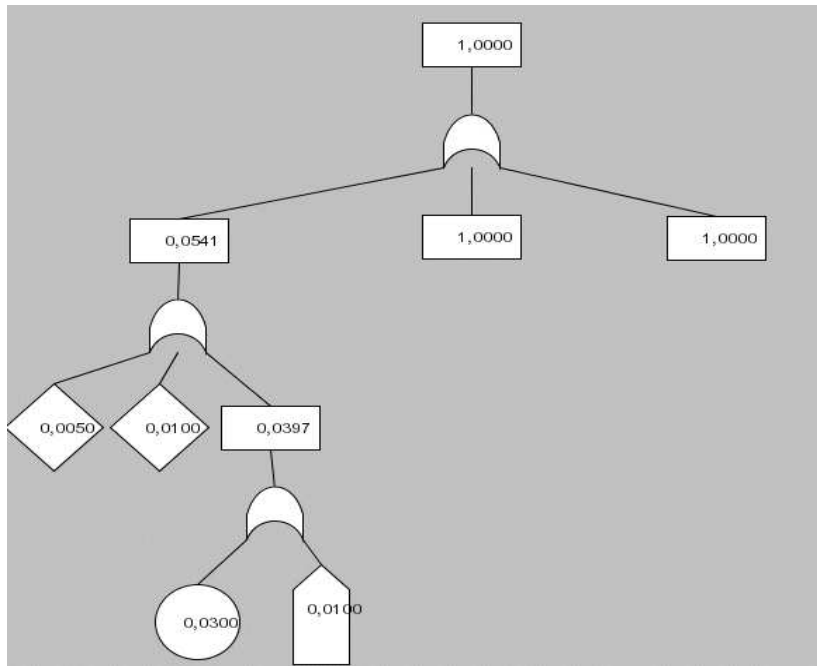
**The Title and Info area can now be manipulated since a fault tree component has been selected.**

Next we focus on the mechanical failure. Reasons that the server might fail could be because of a component overheating, or short-circuiting or even a power loss. There might be several reasons for a mechanical component overheating or short-circuiting, but we do not want to go further into detail. So we place these as Undeveloped Events. We estimate the chance for a short-circuit is around 0.5% per month, and the overheating as 1% per month. We see that we can further develop the power loss and put that as Intermediate Event. Since just one of these failures is enough to make the web server fail, we put an OR Gate in between. After the events and gates have been placed, connect the wires between them, and enter the title and any additional information needed. The Undeveloped Events also take a probability input (set to 1 as default), and these should be changed to the given probabilities above. Remember to write them as a number between 1 and 0. So 1% is really written as 0.01.



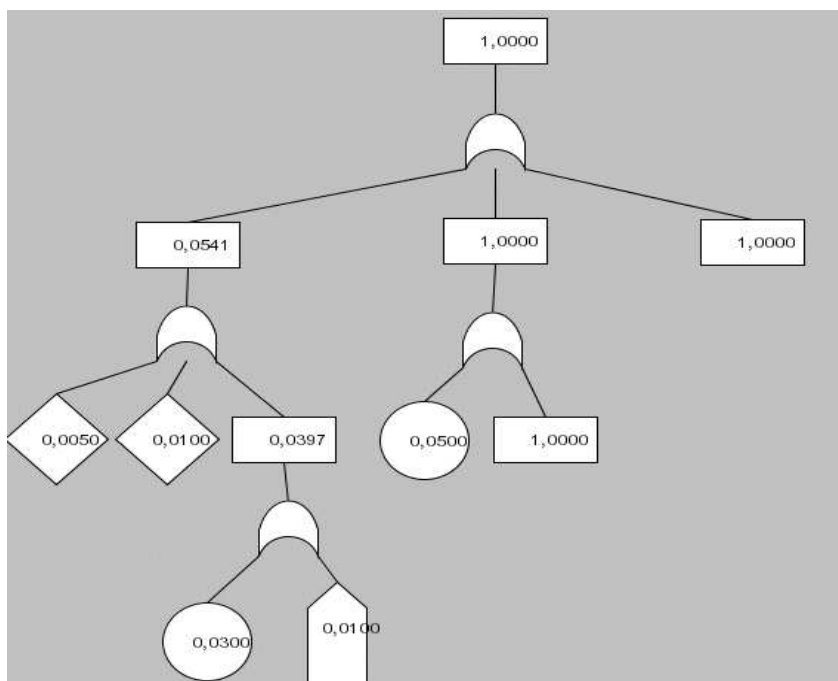
**The next level for the mechanical failure faults have been added.**

We define that power loss can happen for two reasons, either human error (someone turning off a switch that shouldn't be) or a real power loss. The human error is an event that can't be further developed so it's put as a Basic Event, whilst the power loss is out of our hands, so we put that as an External Event. Since either of these are enough to make our server fail, we put an OR Gate in between. The probability for a human error is estimated at 3% per month, whilst the real power loss is put at 1% per month.



**The last level of the mechanical failure faults have been added.**

Since the mechanical fault event can't be further developed, we continue on to the software fault event. We can further develop this to be attributed by either a hacker attack or a bad configuration. The bad configuration is a Basic Event and therefore not further developed. The hacker attack can be however, and is therefore set as an Intermediate Event. Since either of these faults are enough to make the server fail, we put an OR Gate in between. We estimate the bad configuration to happen around 5% per month.

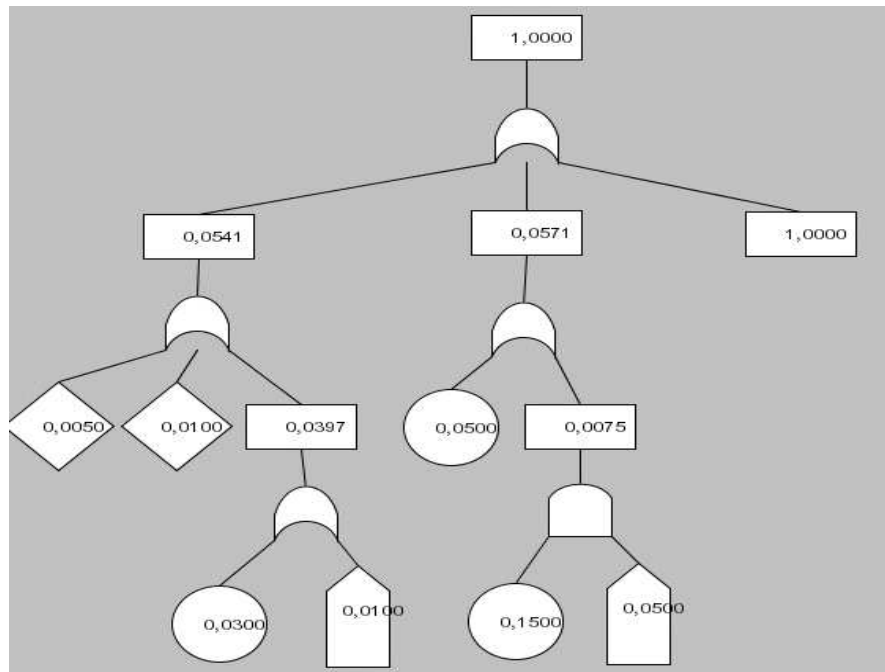


**The first level of the software failure faults have been added to the fault tree.**

The possibility of a successful hacker attack can only happen if we haven't upgraded our software, a Basic Event, and if the firewall isn't properly configured, an External Event since its not in our control.

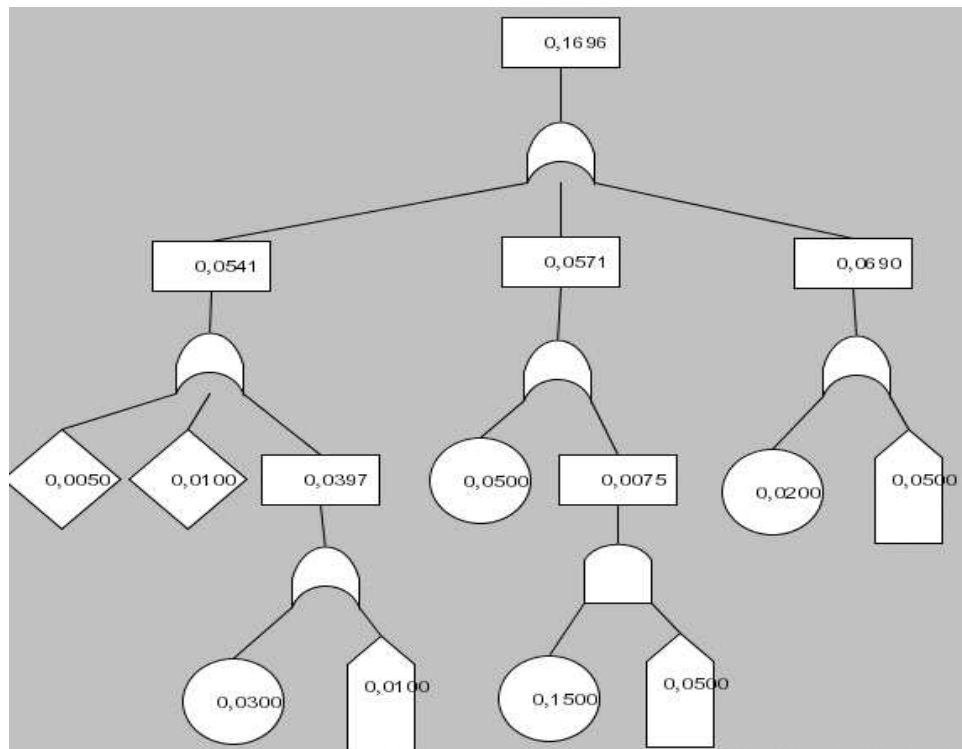
We place an AND gate between since both of them have to happen before they affect the higher level event.

The probability that we haven't upgraded our software is set at 15% per month. We set the probability of a bad firewall configuration to 5% month.



**The last level of the software failure faults have been added**

Finally, we move on to the ISP fault event. We only define two events below this, as a Basic and External Event, either that the ISP turns off our access because of failure to pay our bills, or that the ISP itself has problems that affect us. Since either fault is enough to deny access, we put an OR Gate in between. Estimated failure of paying our bills is set to 2% per month. Estimated failure of the ISP itself is set at 5% per month.



**The ISP faults have been added, and our fault tree is finished.**

This was only a simple example where we've skipped over some parts of the analysing that can't be done by the "FaultCAT" application, but that has to be done by you, the user. Training yourself in setting proper outer limits and on how much detail you should have in your fault trees, will go a long way to help you make even better trees in the future.

## 7.2 Example 2 -

In this example, we build another fault tree. Although any information regarding how to use the application is skipped, it should still be useful for anyone who wants to learn more on how fault trees can be created.

*"You are responsible for the companies mail server, and you want to find the expected probability of the server failing to receive incoming mail. The mail server has no backup, and traffic has to go through a firewall and a local router before it comes to the server itself."*

The first thing we need to do is to find out where we should put the outer system limit on our analysis. We know that we just want to know about incoming mail, so that means we can ignore the local hosts in our network. We also ignore the border router itself, since we want to check for failure to receive incoming mail after it has entered our network, but we include the firewall and the local router, since they could very well be a reason for stopping traffic to our server.

We do not want to go too much into details either, and therefore we ignore the cabling, and rather focus on the 3 main elements – the mail server, the router and the firewall.

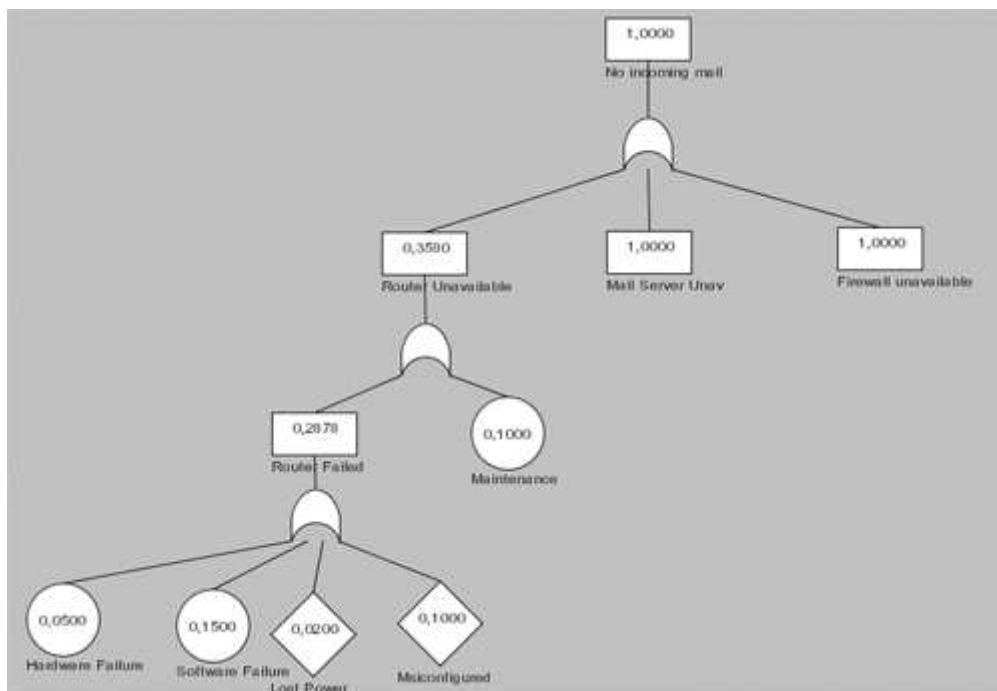
The top event is "Local server fails to accept incoming mail". Next we identify the events that can lead to this failure. We see that a failure in either the mail server, the local router or the firewall is enough to deny any incoming mail, so we put an OR Gate in between. Since these events can be further

developed, we place them as Intermediate Events – “Router unavailable”, “Mail server unavailable” and “Firewall unavailable”.

We add some detail to the router. The first two causes we can think of that makes a router unavailable is either because the router really failed for some reason, or that it’s down for maintenance. The “Router failed” can be further developed, so that’s an Intermediate Event, whilst we do not want to go any further with the “Router down for Maintenance” and we identify that as a Basic Event (note: If we felt we could explore further *why* the router was down for maintenance, we should place an Undeveloped Event instead). Finally we place an OR Gate in between.

There could be many reasons for a router failure, and we list a few here. “Router Hardware failure” and “Router software failure” are Basic Events, whilst “Router lost power” and “Router misconfigured” are Undeveloped Events. We feel that the last two can be further developed, but we are not interested with this at the moment.

As for the probabilities, we’ve concluded with the following estimates. Router maintenance is set to 10% per year. Software failure is set to 15% per year, whilst hardware failure is set to 5% per year. The router loosing power happens approximately 2% per year, whilst a router misconfiguration can happen as much as 10% per year.



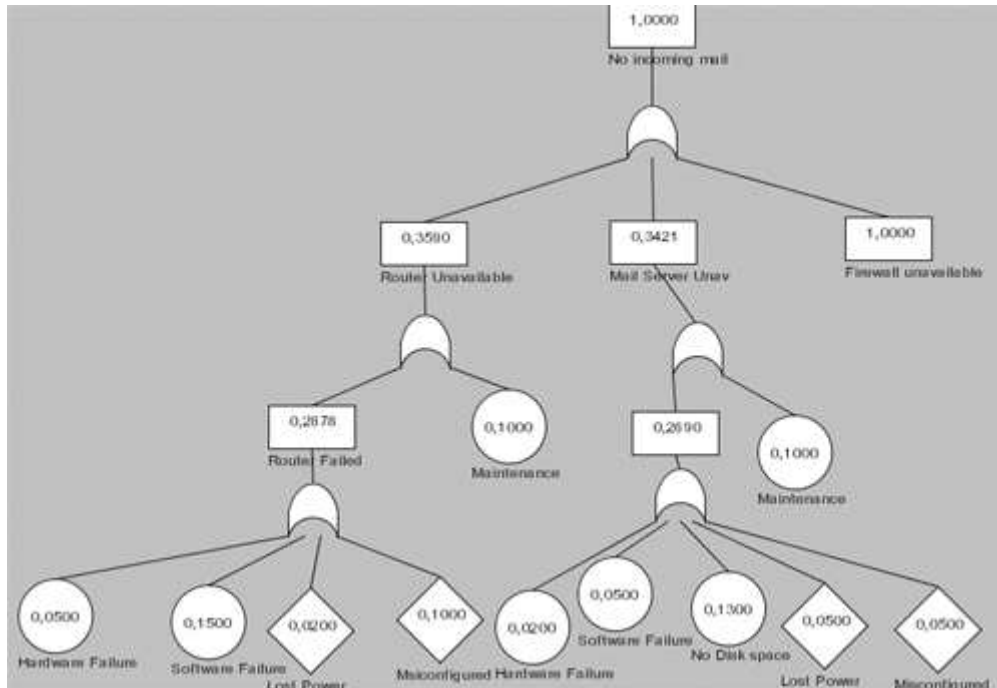
**First part of the tree has been created**

Since there are no further events to detail in the “Router Unavailable” section, we move on to the next one. The “Mail Server Unavailable” can happen either because of maintenance or mail server failure. As with the router section, we place the same events here.

The events below the “Mail server failure” are also the same as the router example, although we add one more Basic Event “Out of disk space”, since the server could very well be filled. An OR Gate is

placed between the higher and lower level, since either of these faults is enough to make the router unavailable.

The probabilities, however, are not the same. We set the “Mail Server Under Maintenance” to be 10% per year. The “Mail Server Hardware Failure” is set to 2% per year, and the “Mail Server Software Failure” to 5% per year. For the “Mail Server Power Loss” we set the probability to 5% per year, the “Mail Server Misconfigured” to 5% per year and the “Out of Disk Space” to 13% per year.



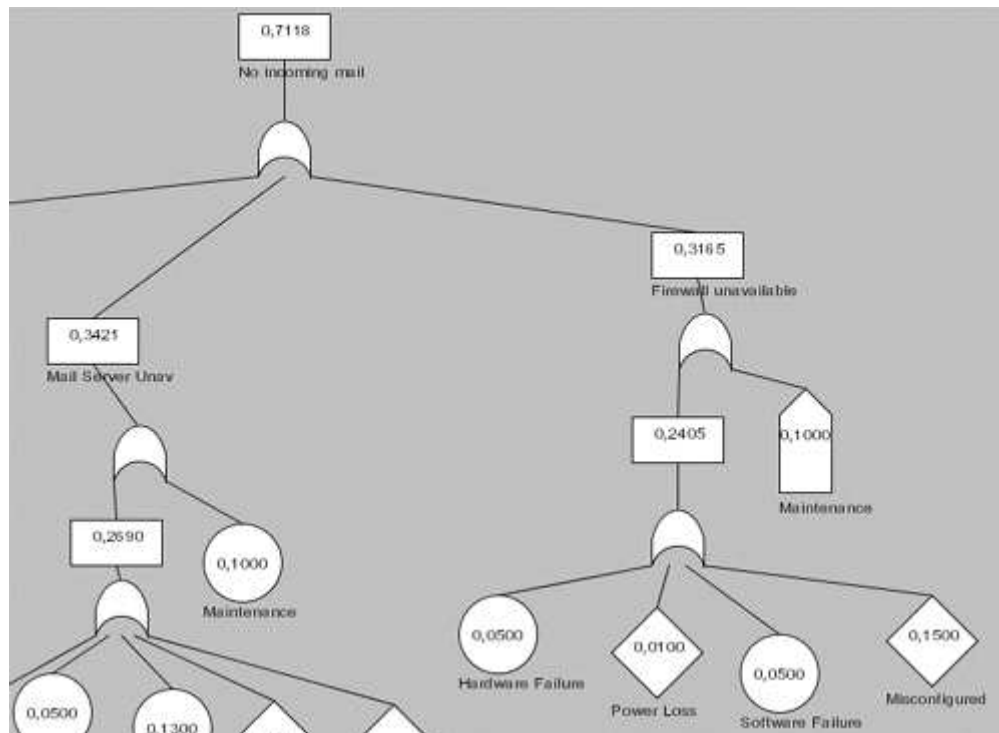
**Second part of the tree has been created.**

We’ve only got the “Firewall unavailable” left, before we’ve got ourselves a fault tree.

Again, the firewall could be failing for two reasons, either because of maintenance or because of a failure, so we place an Intermediate Event and a Undeveloped Event after an OR Gate, and combine with wires. The probability for the maintenance is set to 10% per year.

Expanding the “Firewall Failure” Event, we add almost the same events as we did with the router and mail server failure events. We add the Basic Events “Hardware Failure” and “Software Failure”, as well as the Undeveloped Events “Firewall Misconfigured” and “Firewall Power Loss”. Either of these events are enough to make the firewall fail, so we put an OR Gate between.

The probabilities are as follows – “Hardware Failure” is 5% per year, the “Software Failure” is also 5% per year. The probability of a power loss is set to 1% per year, and the “Firewall Misconfiguration” is an expected 15% per year.



**Final part of the tree has been created.**

## **8. Glossary**

Fault Tree Component - all objects that can be drawn on the drawing area except the Wires.

XML – Extensive Markup Language. A format to encapsulate information into a file.

mySQL – A database.

JRE – Java Runtime Environment. The files that are needed to run Java applications on your pc.

## **9. Bibliography**

Below we have included some sample books or papers for the interested that describe fault trees in more detail.

[1] Fault Tree Handbook – U.S. Nuclear Regulatory Commission

[2] A Probabilistic Approach To Estimating Computer System Reliability – Robert Apthorpe