Critical Embedded Systems

Reliability modelling with fault trees

Introduction to the usage of the **SHARPE** tool

András Vörös

Version: 1.0

Budapest University of Technology and Economics
Department of Measurement and Information Systems

1 Introduction

Reliability modelling is becoming an increasingly important task in the design of today's IT infrastructures. As the services have more and more tasks, failures are becoming more costly for companies. The table below contains data of a study from 2003, which shows the cost of service outage in some industries:

Case study	Yearly income	Cost of outage	Cost/hour
Energy industry	6.75 billion $	4.3 million $	1624 $
„High tech"	1.3 billion $	10.2 million $	4,167 $
Health care	44 billion $	74.6 million $	96,632 $
Travel	850 million $	2.4 million $	38,710 $
Finance (USA)	4.0 billion $	10.6 million $	28,342 $

In order to be able to estimate the costs (and dangers) in advance, we need the means of reliability modelling and the underlying mathematics.

2 Modelling formalisms (overview)

We have multiple solutions to calculate the reliability (and related) properties of the systems:

- simulation
- analytical solution

The advantage of simulation is that any model can be analyzed, there are not so tight constraints (distributions, modelled behaviours) like the ones that characterize the analytic solvability. However, the major drawback is that the obtained information is limited: in many cases it cannot be decided whether we have run enough simulation cases.

The analytical solution has the advantage that it gives accurate results. However, it cannot be used for many models, especially for dynamic models.

The various approaches and options are shown in the following figure:

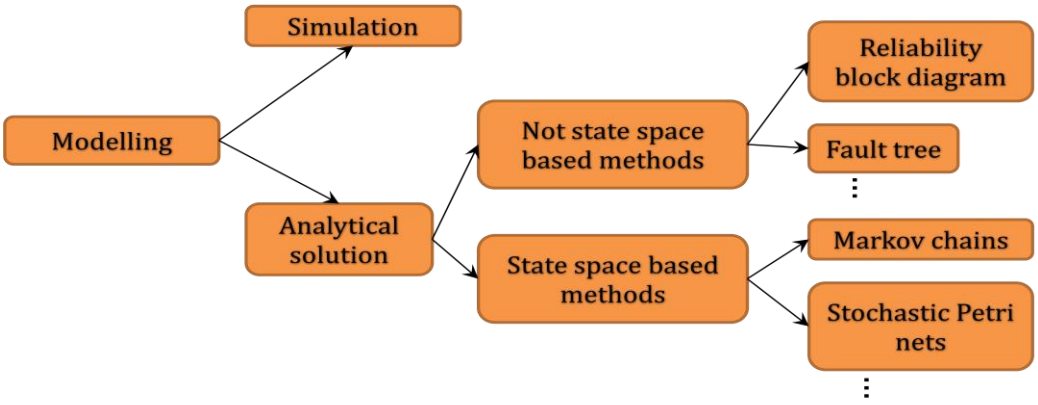

Figure 1. Reliability modelling overview

3 Modelling example

In the following section the concept of fault tree modelling is introduced with the help of the tool **SHARPE**. In addition we review the analysis capabilities of such approaches.

3.1 Example infrastructure

We are going to design the fault tree model of a simple computer infrastructure. Our example is a simple network infrastructure providing web and other services.

The infrastructure consists of the following components: one cluster of web servers, one cluster of SQL (database) servers and the Disk subsystem providing services for the database servers. This solution is redundant regarding the web servers and SQL servers which increases the availability: if one of the web servers or the SQL servers is out, the service is still on. In the following we are going to analyse the causes of the outage of the full service (provided by the infrastructure).

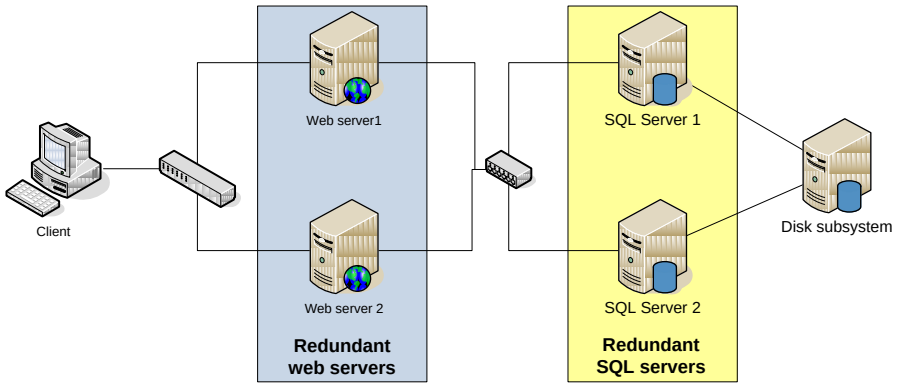

Figure 2. web infrastructure

3.2 Designing the fault tree

We start the design by choosing the top level event: this is the outage of the service. In order to be able to run the service, we need at least: one working web server, one working SQL server and the working disk subsystem.

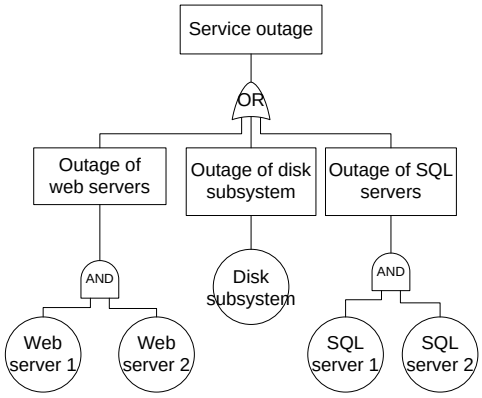

Figure 3. Fault tree of web infrastructure

In the following we show the fault tree model of Figure 3. designed in the tool SHARPE (events *ev* represents the failure of the components):

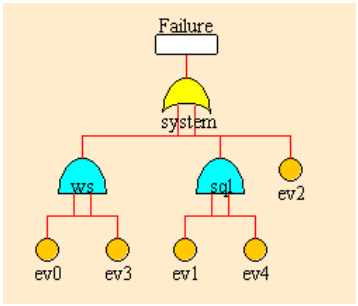

Figure 4. SHARPE fault tree of the infrastructure

As we can see from the figure, the SPOF (Single Point of Failure) of the system is the disk subsystem: the outage of the disk subsystem is enough in itself to cause system level outage.

Some remarks for using the tool **SHARPE**:

- It was developed long time ago, but it is free of charge for academic usage
- After designing a gate, it is only possible to change some of the properties: this can be done by the "*Modify*" button
- The name of the gates cannot be neither „*and"* nor „*or"* (nor „*AND*" and „*OR*"). The gates being put on the canvas, we cannot modify the name of it
- In the names we cannot use the following: space "␣", dash "-", special characters (for example !,%)
- When changing to design the next element, use the *Validate* button! It accepted the values if it became orange! (actually it is preferable to use this button as frequently as possible)
- Decimal point (".") signs the end of the integer
- You can add a new gate by clicking on the event (or node)

3.3 Measurements

We are going to model the reliability of the components by exponential distribution (with parameter λ), where the expected value (or expectation, mathematical expectation, EV, mean, or first moment) of the failure is $1/\lambda$.

The components have the following reliability parameters:

component	λ
web server	0.05
SQL server	0.01
disk subsystem	0.2

We can calculate with the help of **SHARPE** the reliability curve of the system:

We have to choose the *Analysis Editor →Analysis* point in the menu to reach the analysis window.

In the analysis window choosing the *Parameters* tab enables us the calculation of the following values:

- *Reliability* (in a single moment)
- *Unreliability*
- *Mean Time to Failure* (MTTF), the expected time of the failure
- *Variance*

Running the analysis for the fault tree model, the tool calculates the expected value of reliability of the system a time point $t = 10$. The output of the tool is the following:

```
**************************************************************
 *********  Outputs asked for the model: ft **************
 Reliability at time 10
 Reliability:   1.13347087e-001
```

The *Graph* tab at the analysis window brings us to the following window:

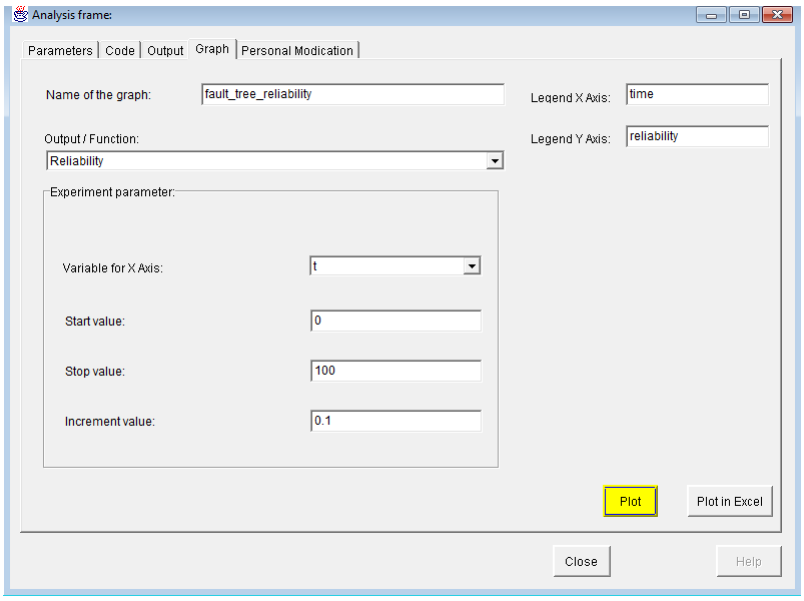

Figure 5. Fault tree analysis settings

It is important to choose the proper parameters, it is unnecessary to display the results longer than it is relevant. After setting the parameters to the proper values, we can get the following plot:

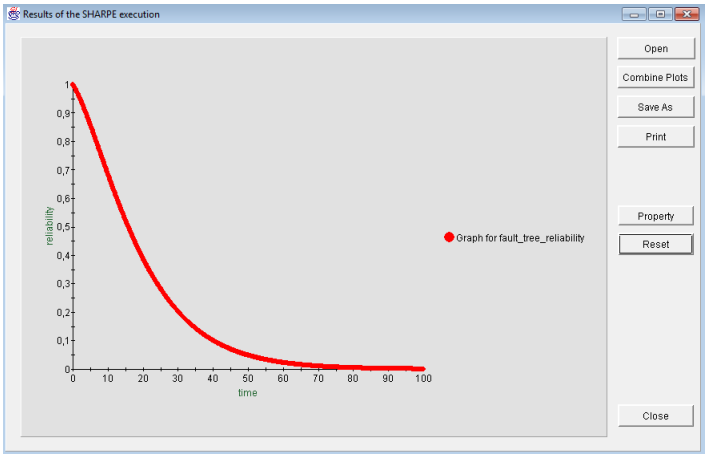

Figure 6. Reliability curve of the infrastructure

We can save the diagram by choosing the *Save As* button. This is very useful if we want to depict more functions (results) in a single diagram.

As we identified from the fault tree, the SPOF is the disk subsystem in our infrastructure. We would like to increase the reliability; a straightforward approach is to use a redundant disk subsystem. In the following we are going to examine the reliability of a modified system: we are going to use a redundant disk subsystem, where there will be two of them. The modified fault tree is depicted in the following figure: we are going to use an AND gate joining the events of the failure of the SQL server 1 and SQL server 2. As they provide the service in a redundant way: any one of them is working means that the service of the disk system is up.

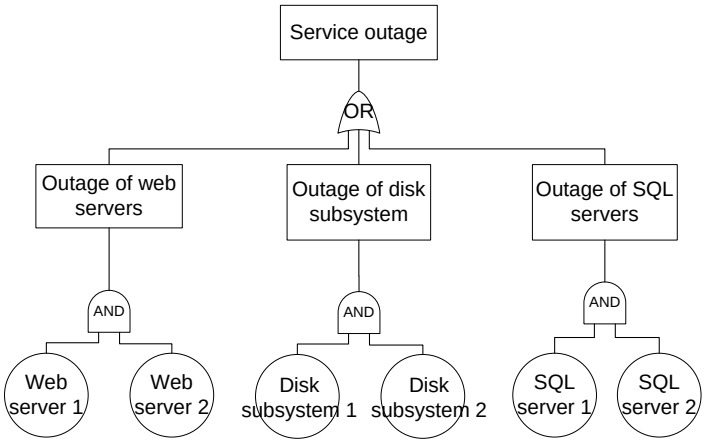

Figure 7. Modified fault tree

The formerly used *Reliability* option produces the following output:

```
**********************************************************************
 *********  Outputs asked for the model: ft **************
 Reliability at time 10
 Reliability:   2.11354313e-001
----------------------------------------
```

In order to be able to compare the reliability values of the former and recent models, we have to depict the new figure at first. The reliability values of the modified model can be depicted by setting the parameters and then choosing the *Plot* button. It is very important to set the same parameters for different runs: otherwise we will not be able to combine the plots (so we cannot compare them). Let choose the same "Start value", "Stop value" and "Increment value", otherwise the **SHARPE** is going to give an error message! After depicting the reliability curve of the modified model, choosing the *Combine Plots* button will combine the two plots into a single diagram. This enables us to compare the different curves and the reliabilities of the two systems:

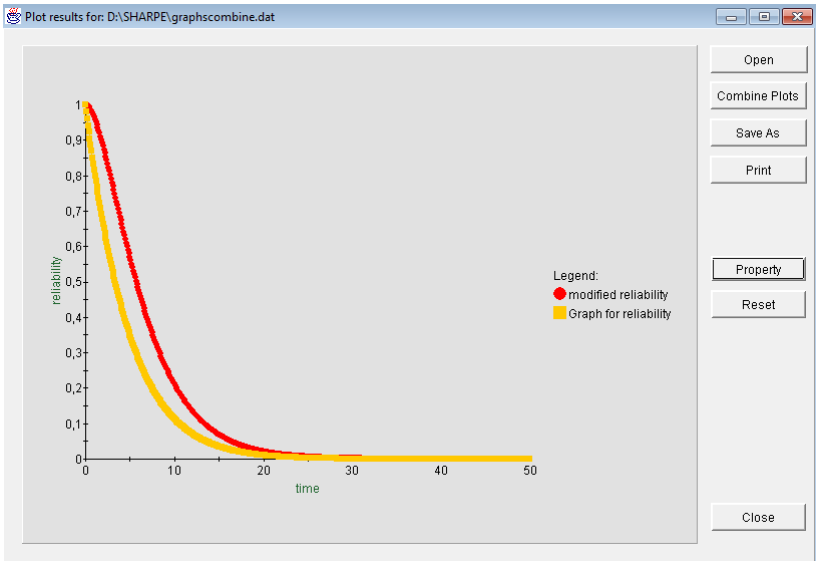

Figure 8. Comparing the reliabilities of different models

The yellow curve is the original reliability of the system, the red curve is the reliability curve of the modified system where redundant disk subsystem is used. It is easy to see that the reliability of the system increased.

Modify the fault tree model and further increase the redundancy by using one more disk subsystem! Analyse the reliability curve: how does the added disk subsystem modified the reliability? Does it have the same effects as it was for the second one?