
“DEPENDENCY MODELLING” = THE BETTER WAY

UNDERSTANDING RISK IN THE REAL WORLD

FOR WANT OF A NAIL

On the eleventh of March 2011 a magnitude-9 earthquake off the coast of Japan damaged nuclear reactors near the town of Fukushima. The earthquake also triggered a tsunami which together with the quake-damage led to the uncontrolled release of radioactive material.

The reactors had of course been carefully designed with a number of safeguards. Yet on that day, one by one they all failed in quick succession. The world watched in horror as the *fail-safe* nuclear infrastructure slowly buckled in the face of the onslaught.

Things had begun well enough as seismic detectors picked up the earthquake and triggered the automatic injection of control rods to close down the reactors. This slowed the fission reaction and stopped electricity production. But the seismic shock to Japan's national grid also cut off external power, which meant the loss of power to the cooling pumps that kept the nuclear fuel-rods safe.

Naturally a set of standby backup diesel generators had been installed, and responded; but one set was then apparently switched off by an operator because the cooling water was too hot? So at least one reactor lost cooling almost immediately. When the remaining standby sets were wrecked in the subsequent tsunami, only emergency battery power sets remained. These were located and overwhelmed in their flooded basements and so this last line of active defence was also unable to keep the fuel rods cool and covered in the reactors and their seismically cracked cooling ponds.

Apparently the generation of an explosive atmosphere due to the hydrogen from the subsequent hot metal/ steam reaction had not been anticipated and even the passive defence of the containment buildings was then blown away. (the emergency ventilation ducting had long since been ruptured by the earthquake anyway). So unconfined radiation escaped and contaminated Northern Japan. The whole infrastructure had unravelled.

The story symbolises the intricate mesh of mutual dependencies that can lead to the collapse of a vital part of a modern, high technology, interconnected system of systems. The Fukushima reactors were designed with safety in mind, but when one of the design assumptions was violated it folded. The pivotal assumption was that they would only have to survive a magnitude-8 earthquake, but on that day in March it faced one of magnitude-9¹.

The Fukushima plant was fortunate in that its design at least had clearly stated assumptions. The same cannot be said of much of the national and international infrastructures on which we depend.

It is more than merely possible that the complex inner-workings of our civilization form an interdependent array of dominoes through which chance events can trace a critical path of collapse.

George Herbert's poem *For want of a Nail*² written in 1640, recounts how a kingdom was lost through a defeat in battle when a warning wasn't received because a horse went lame, having thrown a shoe after a blacksmith saved the cost of a nail.

On such small pivots does the fate of a nation balance. Can we be certain our civilisation won't collapse for want of a nail?

Welcome to the world of **dependency modelling**.

-
- 1 An increase of 1 on the Richter scale means a 10-fold increase in amplitude or a 32-fold increase in energy.
 - 2 See references ...[George Herbert]

RISK AND UNCERTAINTY

Anyone running an endeavour - be it a government, army, utility, bureau, infrastructure, company, project or village fête – will sooner or later need to look at risk.

Though pivotal to the survival of any enterprise, Risk is perhaps the least understood of all executive disciplines. Most of us shy away even from thinking about it. It sounds horribly negative – all to do with disasters, unintended consequences and failure. We're uncomfortable with it and much happier with goals and objectives.

Risk is a term used in different contexts to have different meanings. It's used in finance, engineering, warfare, politics, project management, insurance, IT, economics, business, psychology and everyday life.

But what is it? There are as many definitions as there are authors on the subject. Some are obscure, some vague, some are descriptive and others numerical.

Here are a few:

- The effect of uncertainty on objectives - ISO 31000 (2009) /ISO Guide 73
- The potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization - often described as *IT risk*.
- (Probability of Disaster) x (Cost of Disaster) - the *expected utility*.
- A state of uncertainty where some of the possibilities involve a loss, catastrophe, or other undesirable outcome - [see Hubbard 2009]

These are all pretty uninviting definitions. We won't dwell on them.

We'll introduce a more attractive one shortly.

Gunpowder

Some of these definitions involve the probability of success or failure. But in popular imagination risk is more than this.

An illustration beloved of insurance actuaries makes the point. Traditionally 18th-century gunpowder factories were always blowing up. They did so with such regularity that insurers came to know the odds quite accurately and simply fixed the premium accordingly. That way the insurer made his profit come what may. To the insurer the risk had almost vanished. As long as the probability of disaster is well known he can reduce his risk almost to nothing.

If we take this idea to the extreme, the total certainty of a disaster means there is nothing we can do to prevent it - it's simply going to happen. This is not what people normally mean by risk. To most of us, risk is more about uncertainty.

Imagine you are an investor, and an applicant comes to you with a business-plan. Suppose it's a novel enterprise in an untried market whose success depends on many things, and you suspect that the applicant, in his enthusiasm might be entertaining unjustified optimism. You would rightly be concerned with the possibility of some unforeseen or ill-understood circumstance that could overturn all the assumptions behind the business model and take the applicant by surprise, and thereby cause the enterprise to fail.

All this leads to the following observations.

Risk is about goals

You may need to think about it, but with no goals there is no risk, and the term *risk* only makes sense in terms of those goals.

For instance if your goal is to stay alive then a dangerous cliff might be a risk, but if you are bent on suicide then the cliff could be an *opportunity*, and a safety net might be a *threat*.

Goals depend on other things for their success

Pick a goal, any goal – like throwing a successful party. This might depend on friends being available on the day; maybe on the weather; on there being no industrial action preventing people travelling; on the reliability of the sound system; on the caterers turning up; on you or some close relative not being taken ill at the last moment; on there not being some sudden National emergency ... You get the picture.

Risk is about things we cannot control, predict or understand

We cannot control all the things we depend on. If we could **control** everything we depend on there would be no risks. Everything would go according to plan and nothing would take us by surprise. But we can't. It gets worse. We can't even **predict** accurately which dependencies will let us down. If we could predict accurately then at least we wouldn't be taken by surprise and we could make adjustments to compensate – like the insurer of the gunpowder factory.

Finally, if we cannot even fully **understand** all the things that our goal depends on then we certainly can't control or predict them, and in a sense our risks are even greater.

Risk

Risk then is about goals and how their achievement depends on things we may not be able to control, predict or understand.

This is the principle we will try to capture in our definition of risk. O`

Stopping the infrastructure unravelling

If we wish to make informed decisions to try stop an infrastructure unravelling we need to be able to understand and calculate risk. Only then can we understand what it takes to keep it all together. While this doesn't guarantee success it just might be our best shot.

This is Not a Maths Book

Meanwhile on with Dependency Modelling.

We're going to describe the principles in more-or-less non-mathematical terms. You may be surprised how simple most of the concepts are. Most of them are about reality, not about computers or mathematics. If you're not a technical type, just take it slowly, one step at a time and all should become clear.

DEPENDENCY MODELLING

Dependency Modelling is a way of analysing the risks to an enterprise - anything for which a suitable model can be built. It works by analysing the model.

However it doesn't work by dealing with risk directly, but rather by studying success and what we want to achieve, and it's much more interesting and less scary than thinking what could go wrong.

It is based on the idea we already discussed, that risk is about goals and all risk springs from the fact that achieving our goals depends on many things some of which we can't control, or predict, or in some cases, even understand.

By combining its various functions, Dependency Modelling can

- Find the probability of achieving goals
- Find the true risks to those goals
- Find those dependencies which are most pivotal to goals
- Find the best ways to deploy resources for maximum benefit
- Find the cost-effectiveness of countermeasures
- Find the most likely causes of success and failure
- Find the most critical uncertainties
- Find single and multiple-cause failure modes
- Measure risk
- Find ways to reduce risk
- Find ways to reduce risk at least cost or reduce cost at least risk, or both
-

It can be applied to any complex system such as an organization, a company, a project, a machine, a doctrine, a diagnostic, a government, a charity, a business model, or a village fête.

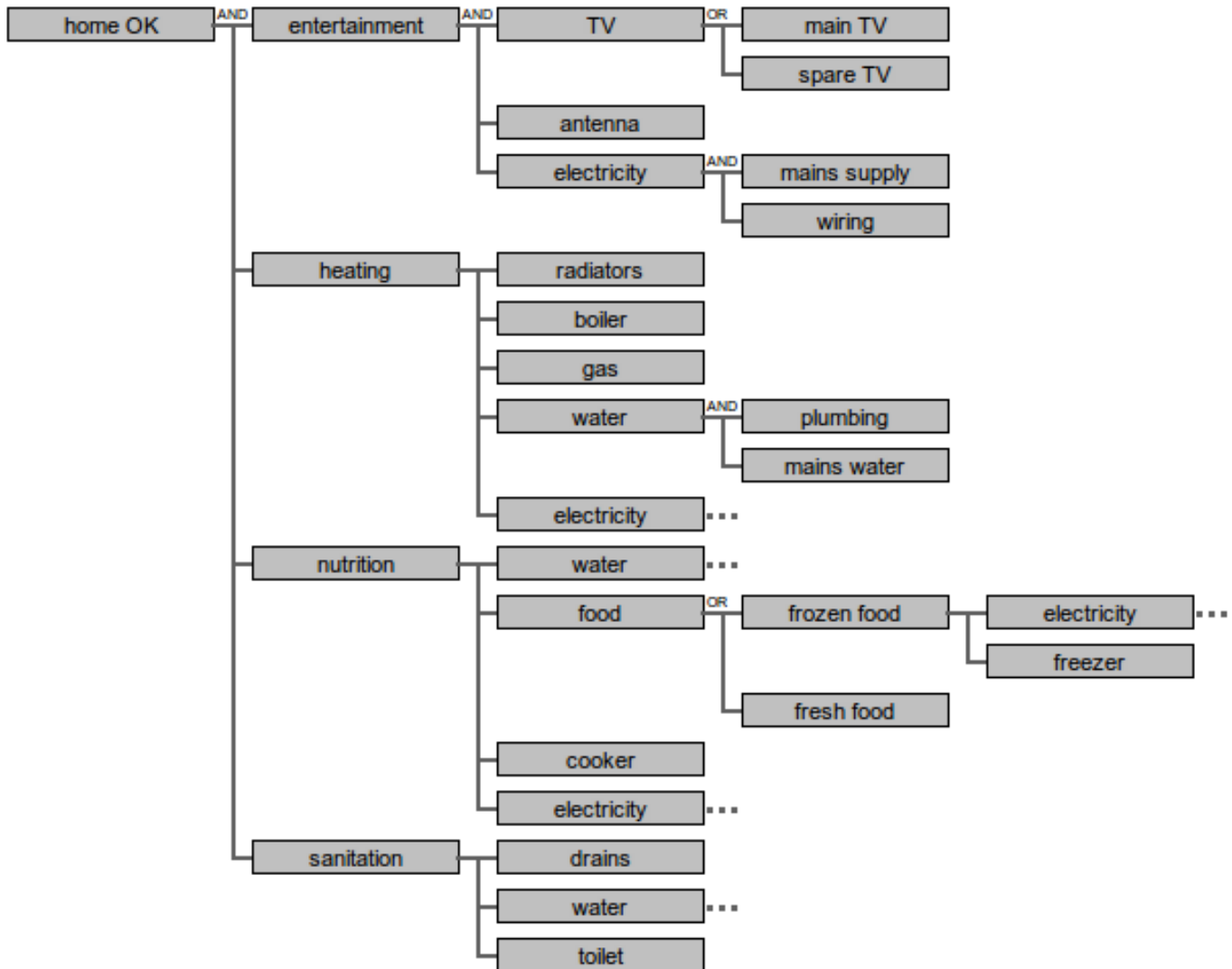
Since there is no word that adequately covers all these, we'll use *enterprise*. We won't call it a *system* since that could mean the computer used to do the Dependency Modelling.

The principle behind Dependency Modelling is that ideally the analyst should not have to work out all the things that could go wrong with his enterprise. His responsibility should be limited to providing a sufficiently accurate model of the enterprise, and software will do the rest.

While we can't totally achieve that ideal, Dependency Modelling goes a long way towards it.

So in Dependency Modelling we build an abstract model of the enterprise which software then analyses. In reality we have to build the model in such a way as to help the software along a bit.

To give the flavour at the slight risk of getting ahead of ourselves, here is a simple model of some of the entities found in a home together with their interdependencies. The goal is a properly functioning household, labelled **home OK**.



Model of entities in a home

All Dependency Models represent a goal plus the things that achieving it depends on, plus the things those things depend on, and so on.

(In the diagram above, the three dots (. . .) after the second and subsequent appearances of **electricity** indicate that the latter has already been shown to depend on **mains supply** and **wiring**, and to keep the diagram compact this dependence is not shown after the first appearance.)

Having a model

There are a large number of advantages of having such a model, and here are a few:

- If forms a language to discuss risk with other people.
- If forces us to understand and articulate what we are trying to achieve.
- Any misunderstandings we have are made visible to ourselves and others, and are thereby more likely to be uncovered.
- It allows us to analyse risk.

Good model, bad model

Having a flawed model is better than not having a model because without a model there is little hope of any misunderstanding coming to light.

The concept that a flawed model is usually better than no model is not the sole preserve of risk analysis. Indeed there is no field of science that can claim its theories are totally correct. Our state of knowledge is simply the best we have at any one time, and errors are put right as and when they are discovered. Aristotle's laws of motion were replaced by Newton's, which in turn were replaced by Einstein's General Theory of Relativity, and are likely to be overturned by a Theory of Everything. But in their own times the earlier ones worked well enough.

Without a model people tend to use fallacious reasoning such as “why should I give up smoking when I could be killed by a truck?”. If the questioner had a model showing his life goals and how they depend on his relationships, wealth, health etc., the error would be immediately obvious.

Probability versus Severity - Aside

Before we get started here's a quick aside which we hope won't insult your intelligence.

Many people are confused by the concept of probability. To be specific they confuse the probability of an event with the severity of the event, but these are quite different ideas.

The *severity* of say, an earthquake might be measured on the Richter Scale. This indicates how much damage is likely to be caused by an earthquake of that severity.

The *probability* of an earthquake is nothing to do with its severity, but rather it measures how likely it is to happen at all.

Even the designers of the Titanic were confused about the difference. The vessel was believed by some to be virtually unsinkable, and as a result it was fitted with only enough lifeboats for half the passengers. Now a moments thought shows that either it could sink or it couldn't. If it could sink it needed enough boats for everyone. If it couldn't then it didn't need any boats. There was never a logical argument for half the boats. Yet an otherwise intelligent designer confused likelihood with severity.

In the context of future risk we need to say what we mean by probability. When an event takes place that could have different possible outcomes, the probability that it will have a particular given outcome is simply the proportion of times when it actually does have that outcome.

So if a certain event can have six possible outcomes, called say *outcome number 1* to *outcome number 6*, and if say the proportion of occasions on which *outcome number 6* occurs is 41%, then we would say that *outcome number 6* has a 41% (or 0.41) probability of occurring.

(Of course if the event consists of rolling a dice and the probability of landing a 6 turns out to be 41% we would say the dice is loaded!)

When we come to discuss the probability of achieving a new goal that has never occurred before we can't easily use this idea because there is no history of successes and failures, nor probably any way to carry out repeated experiments and measure the proportion of occasions when the goal is achieved. So we need some way to visualise what *probability* means in this context.

This thought experiment may help. Suppose we somehow have access to thousands of parallel universes, all very similar, and in each one a similar entrepreneur attempts to bring a similar venture to fruition. Suppose that in 73% of these universes the venture is deemed successful, then we would say that the probability of a successful outcome to this venture is 73% (or 0.73).

Importantly, it's nothing to do with the degree of success, but rather whether the outcome falls into the category called **success**. That's the difference.

How could we come up with such a figure if we can't do the imaginary experiments? Well we can estimate it another way if the outcome is determined by the various things it depends on. If we understand the statistics of those dependencies it turns out that we *can* actually calculate the probability that the outcome is successful. Or to be more precise, we can get a computer to do it for us.

Finally note that the “chances of an outcome” is simply a slightly looser way of saying the probability of that outcome. We gloss over the fact that *chances* sounds plural while *probability* is clearly singular.

Top Down Positivism

To start at the beginning, a dependency model is based on goals and objectives, and the prerequisites to satisfy these goals. In other words it is a positivist, top down approach working from goals to requirements.

This is in strong contrast with other methodologies which focus on faults, disasters and failures.

There are a number of advantages in the positivist approach, not least being that it is easier and more intuitive to think of goals and requirements. Senior management people are more comfortable working with them than with disasters.

In what follows we're going to keep the technical discussion simple and develop the ideas through examples. We'll postpone definitions as far as possible until the need for them is clear.

Lets start with something simple. We will deliberately avoid anything involving threats to national infrastructures because we want to concentrate on the principles of modelling.

Suppose we want to visit friends who live several hundred miles away. Our goal is a successful visit, and we're going to limit our analysis just to the journey. The issues we're concerned with here are possible last minute cancellation by either party and possible travel problems. So the success of our journey will depend on:

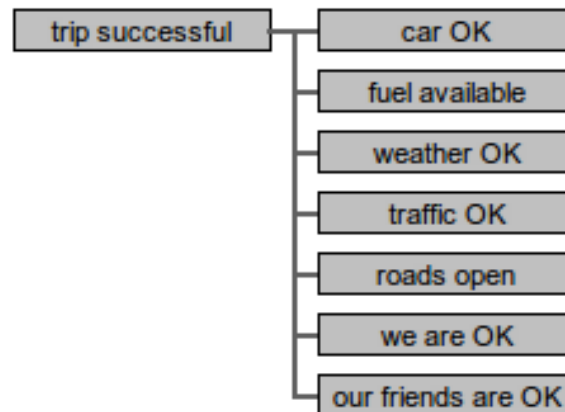
- a car that works properly
- the availability of fuel
- the state of the traffic
- possible road closures
- ourselves avoiding being taken sick at the last moment
- our friends avoiding being taken sick at the last moment

We will call these the *dependencies* of our goal. (To extend the terminology, a goal is the *dependant* of its dependencies.)

(Notice that if we were analysing the risks to a government our list would probably include items relating to ministries, departments, armed forces, police divisions, taxation, health services etc. At the other extreme if we were analysing the workings of a motorcycle our concerns would be with the parts of the engine, the fuel tank, brakes, tyres and so forth.)

The above list is not supposed to be comprehensive, it's just a starter to illustrate some ideas. You many disagree with our choice of dependencies, but we picked them mainly for the insight into modelling they provide and not because we really want to know about the risks to a trip.

The success of our trip depends then on all these things, so we could draw a diagram like the one below.



2-layer trip model

In a more realistic situation the success of a goal would probably depend on many more than just seven items, but this will do for the moment.

We can view the entity **trip successful** as a sort of goal with various degrees of achievability. It might for instance be fully achieved, partially achieved, slightly achieved or totally unachieved, or any degree of achievement we care to come up with.

But for the moment we'll keep things simple and limit it just to two possibilities: *failure* and *success*. But *bad* and *good*, or *no* and *yes*, *negative* and *positive*, or 0 and 1 would do equally well.

We can also view the seven dependencies – such as **car OK**, **fuel available** and so on - as sort of minor goals in their own right, just like **trip successful**. These too have various degrees of achievability, and again for the moment we'll limit the possibilities to just *failure* and *success*. It is important to note that these are not fundamental limitations, just convenient simplifications for the present discussion.

Paragons

To avoid later misunderstandings we're going to introduce a new word. We loosely described **trip successful** and **car OK** and all the other entities as “sorts of goals”. This turns out to be misleading.

These entities are much more specific than “goals”. Moreover for most people the word “goal” is loaded with unhelpful associations. In fact there is no word in English that adequately describes what the entities **trip successful**, **car OK**, etc. represent in our model, so we're going to invent one. The word is **paragon**.

While for the moment a paragon³ can be thought of as a “sort of goal” which is its *raison d'être*, nevertheless that's only an approximation which we will shortly modify as we add extra attributes and distinctions during the discussion.

One of the attributes of a paragon is its name or label. Our eight paragons have names like **trip successful**, **car OK**, ... **our friends OK**. The name represents the paragon's *raison d'être* – the positive idea we want to associate with it.

The degrees of achievability that we called *failure* and *success* we will now refer to as **states**. So paragons also have at least two states. These states have names describing the degree of

3 Paragon is of course already an English word, but most people are uncertain of its exact meaning and merely associate it with desirability and goodness, and it has a robust, respectable, classical sound to it, which is just perfect for our needs. Originally it meant a hard stone used to assay the purity of gold by gauging the force needed to make an indentation - pure gold being very soft. Now it means a flawless jewel, a symbol of perfection or similar.

achievement of the *raison d'être*. The states must always be arranged in an order from worst to best, which is why we referred to *failure* and *success* rather than the other way round. The reason for this will become clear later. One benefit of this – by no means the main benefit - is that we can refer to the states by their index number (starting from zero) and the higher the index number the better the outcome. So *failure* and *success* could equally well be referred to as states 0 and 1.

Sometimes verbs are used as shorthand to describe which state a paragon is in. A paragon is said to **succeed** when in its success state, and to **fail** when in its failure state.

Causality

If we regard these diagrams as describing the relationship between cause and effect, then cause is on the right and effect to the left. Putting it another way, causality flows from right to left. So **car OK** is one of the causes of **trip successful**.

What do we want from the model?

Now we need to ask ourselves “what do we want from our model?” One thing must surely be to calculate the chances we will achieve the goal. Think of this as the “probability that the paragon **trip successful** will be found in the state we called *success*”.

We might also want to be able to find the most likely cause of failing to achieve *success*, and the most important elements in our model from a success point of view, and so forth. We'll show how to answer all these questions from our model in due course. And of course we'd like to know the **risk** to our goal.

In passing we note that there is a completely different kind of risk information we can derive from our same model called **Failure-Mode Analysis** and it's discussed in a later section.

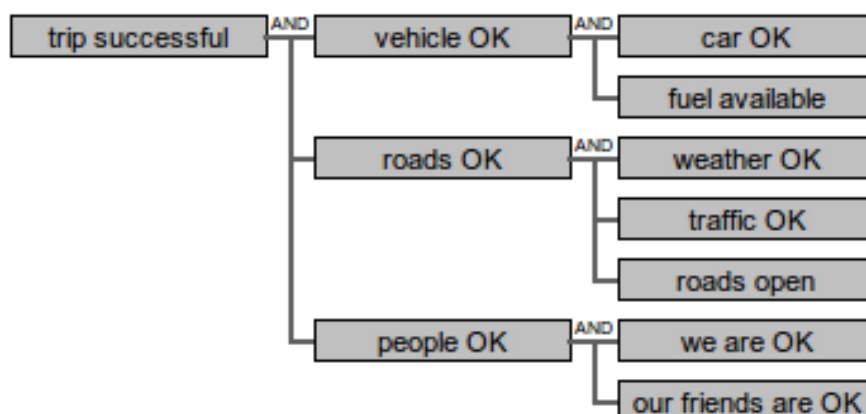
We'll start by an observation.

A list isn't a good model

Now it turns out that the above, **2-layer model** representing the relationships would **not be much use in risk analysis** because of the crude way it lumps all the dependencies together.

In this form it's no better than a list. But that's the least important reason.

For reasons we'll soon make clear, a much better model would be this:



Better model

This model is multi-layered - three actually - and consists of small building blocks, but why is this better?

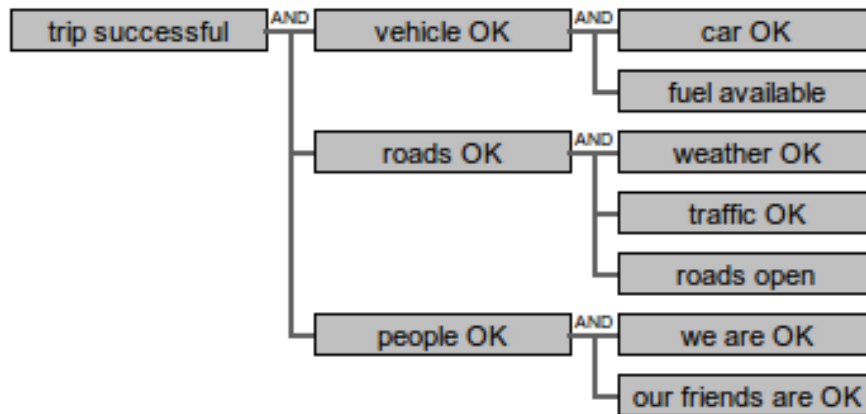
Here's why.

1. A multi-layer gives much us more insight. This will become clear in due course.
2. Surprisingly, it takes a huge amount of effort from us to specify a two-layer model.
3. It's computationally infeasible to do the calculations for a two-layer model with more than a handful of paragons.

To understand the problems with 2 and 3 we need to look at how the calculations are done.

MODELLING BASICS

Here again is a 3-layer version of our model⁴.



3-layer trip model

We have introduced 3 extra paragons, **vehicle OK**, **roads OK** and **people OK**, and the model now contains 11 of them.

However we now have a model that is much easier to specify. For instance **vehicle OK** can be specified by a table with only 8 numbers because it forms a cluster together with its two immediate dependencies, **car OK** and **fuel available**.

The small table for **vehicle OK** will now look something like this:

states of dependencies		vehicle OK state probabilities	
car OK	fuel available	0	1
0	0	0.9	0.1
0	1	0.8	0.2
1	0	0.7	0.3
1	1	0.1	0.9

Notice that we've put two entries in each row, by having two columns, one for each state of **vehicle OK**. This form of table is called a **Conditional Probability Table** because it gives the probabilities for one paragon's states (**roads OK**) in terms of the states of its immediate dependencies, **car OK** and **fuel available**. So for instance the probability that **vehicle OK** will be in state 1 under the condition that **car OK** and **fuel available** are in states 1 and 0 respectively is 0.3.

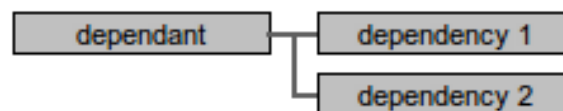
Specified in this way the total number of table entries for each paragon is as follows

⁴ You can access an online computer program to do all the calculations in this document at www.johngordon.org.uk/jsvr .

Paragon	#Table Entries
car OK	2
fuel available	2
weather OK	2
traffic OK	2
roads open	2
we are OK	2
our friends OK	2
vehicle OK	8
roads OK	16
people OK	8
trip successful	16
Total	62

Building Block

If we break the model into manageable building blocks, each one then makes sense in its own right and has its own logic. Here's the structure of a typical building block:



building block

This is a paragon and its immediate dependencies – there may be more than two dependencies of course.

If a model is an essay, then such a building block is a sentence.

Every building block must pass the **Cover-Up Test** which we describe later.

Uncontrollables, Trees and Dynasties

The paragons on the extreme right of each branch of the model don't have dependencies, or at least they're not shown in the model. That's just as well otherwise the model would never end.

Dependencies can have their own dependencies and so on to any number of layers we like and we stop adding more when we feel we've gone far enough. Any paragon with no dependencies acts as a sort of "given" - a starting point. We can think of it as the point where risk and uncertainty enter the model.

We call such paragons **uncontrollables** to emphasise that we can't do anything about changing their properties. That doesn't mean we cannot reduce the risk they pose. It turns out to be quite simple to reduce such risks, but more on that later.

We specify uncontrollables simply with a table with no dependencies. Here is an example for **weather OK**.

State Probabilities for **weather OK**

p0	p1
0.05	0.95

where p0 and p1 are just names for the probabilities that it will be in its bad and good states respectively. Notice that we've written them both in the same row. Think of it as one row – the only row in this case - of a conditional probability table.

Depth of Model and Granularity

Our model can have few or many layers. It is always possible to convert an uncontrollable into a dependant by merely presenting its dependencies – after all everything has dependencies – and in this way we can increase the depth to embrace more and more, lower level dependencies. We stop this drilling down process when we feel we've gone far enough. The leaves at that point are the end-stops, the givens, the uncontrollables.

A model with many layers which starts from a major goal such as the success of a department, and ends with very low-level dependencies such as the effectiveness of a stapling machine, could also be described as having fine granularity. Similarly a model with few layers would have coarse granularity.

Specific meaning of paragon

We pause to make a vital point that we'll come back to later. We must not confuse the paragon we labelled **industrial harmony** with the everyday meaning of that phrase. Here **industrial harmony** is the name of a paragon whose associated goal, if articulated would be something like:

the state of affairs whereby the likelihood of roads being open and fuel being available are not adversely affected by industrial action.

By definition, if our paragon **industrial harmony** is found to be in its failed state, then since **roads open** has zero probability of being in its good state unless **industrial harmony** is in its good state too, **roads open** will fail.

You could find yourself having arguments with non-experts about whether lack of industrial harmony always entails road closure.

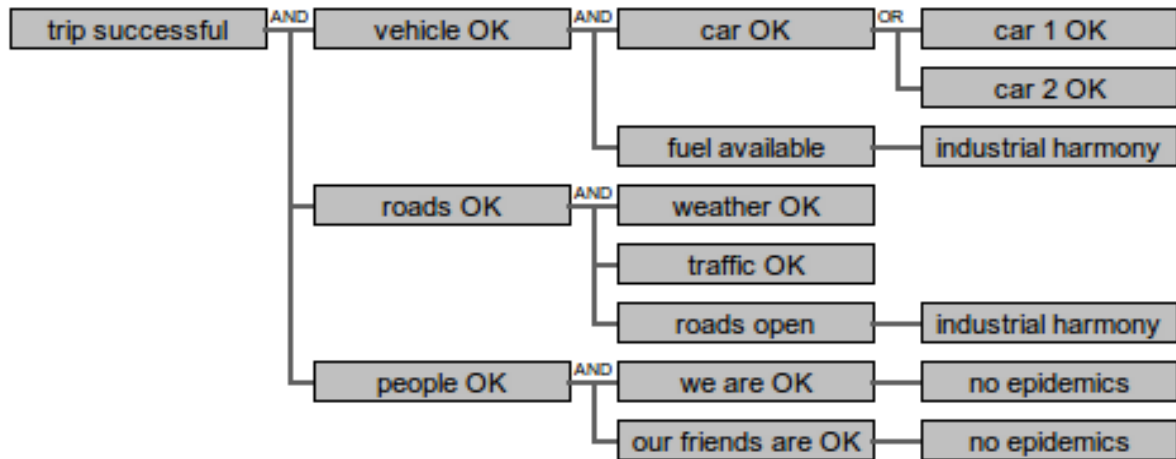
This can be resolved by observing that the inevitable logic of our paragons is not true of the everyday use of these phrases, but comes about because of the very specific meaning we've attached to these paragons. The key point here is that there is a huge difference between the technical significance of the paragon **industrial harmony** and the same phrase used in everyday speech.

There is only really a problem because the labels are necessarily short (to fit into a small box on a diagram), but the meaning behind them is complex.

In short, we must beware of confusing the meaning of a paragon with its label. The label is merely a hint or reminder of the actual definition of the paragons.

Another reason for being fussy about the definition of a paragon is that it affects the answer. How is this? It's because the statistics we give for the paragon depend on the definition.

Continuing with our model, if we have 2 cars we can take account of this in the model too by introducing the paragons **car 1 OK** and **car 2 OK** as dependencies of **car OK**:



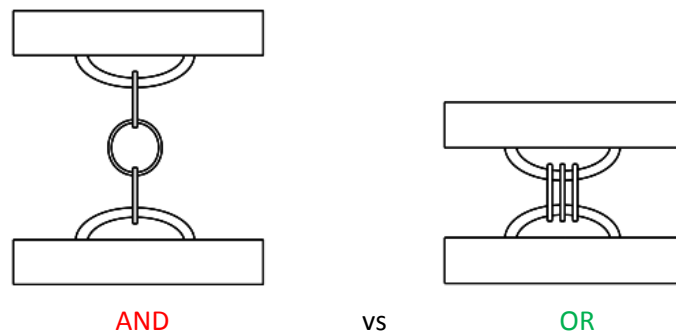
4-layer trip with 2 cars

If either of our two cars is functioning then **car OK** is in its *success* state.

AND and OR relationships

There are two kinds of relationships whose conditional probability tables are of a type that occur sufficiently frequently in models to merit special names. They are AND and OR relationships. They can only be used when the paragon involved have only two states.

The diagram below illustrates the idea. In both diagrams a heavy weight is suspended from the ceiling by three thin rings. In the left diagram – the AND relationship – the rings form a chain with the links in series. In the right diagram – the OR relationship – the rings are arranged in parallel. We assume that the only components likely to fail are the thin rings.



The names come from the fact that the left diagram requires the first ring AND the second ring AND the third rings all to function properly without breaking, while the right diagram requires only that either the first OR the second OR the third ring functions. The AND model will fail if any ring fails, whereas the OR relationship will succeed if at least one succeeds.

OR good, AND bad

A vitally important point to grasp is that clearly: **an OR relationship reduces risk while an AND relationship increases it.**

This observation is both shocking in its implications yet beautiful in its elegance.

Note that for this statement, and for many other statements to be true, all paragon must represent something positive, and low numbered states must represent something bad, and high numbered

states represent something good. This fussiness is of course just another of the reasons we insist on a strict definition for **paragon**, and why we don't just say goals. We'll define paragons more fully shortly.

If the dependencies are D1 and D2 and the goal is G, and p0 and p1 are the probabilities that the goal will be in states 0 and 1, then the tables for the two relationships are like this:

AND relationship				OR relationship			
D1	D2	p0	p1	D1	D2	p0	p1
0	0	<i>1</i>	<i>0</i>	0	0	<i>1</i>	<i>0</i>
0	1	<i>1</i>	<i>0</i>	0	1	<i>0</i>	<i>1</i>
1	0	<i>1</i>	<i>0</i>	1	0	<i>0</i>	<i>1</i>
1	1	<i>0</i>	<i>1</i>	1	1	<i>0</i>	<i>1</i>

(Note that in the D1 and D2 columns the 0s and 1s are just names of states, while in the p0 and p1 bold italicised columns they are numbers measuring probabilities.)

These are of course just special cases of conditional probability tables.

In the diagram called *4-layer trip with 2 cars* all the relationships were AND type except **car OK** which was OR type.

Countermeasures

Countermeasures can often be represented as OR relationships.

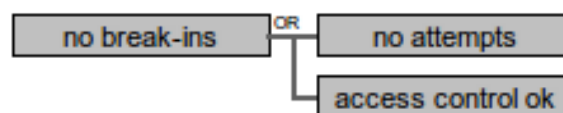
Suppose we have a threat, such as the possibility that bad guys may break into our premises, and a countermeasure such as an access control system and we want to represent this idea in a dependency model.

Firstly we decide what we are trying to achieve. In this case it's **no break ins**. Note particularly that it's **not** an access control system – that's just a mechanism to try to achieve the goal.

Secondly we turn the threat into an asset by labelling its paragon **no attempts** (i.e. no attempts to break in). This is the desirable state of affairs that nobody happens to try to break in. We give it a success probability.

Thirdly we then create a paragon to represent the **effectiveness** of the countermeasure, say **access control ok** and give it a success probability.

Lastly we create an OR relationship:



no break-ins

It is a beautifully logical and concise statement of the relationship between the goal, the threat and the countermeasure.

Paragons

We have seen hints that there is more to a paragon than just a goal. The time has come to be more specific about exactly what a paragon is.

A paragon must have the attributes of

- Name
- Positivity
- Abstraction
- No box-ticking
- Strong definition
- Consistency
- Aspect
- States
- Statistics
- Statistical Independence

We'll take these points in turn.

Name

A paragon must have a name or label. It identifies the paragon on diagrams. The name needs to be short. It is helpful if the name suggests the proper definition of the paragon. It should not hide its import in euphemism. So if you mean nobody is to die don't say *safety addressed* - which could mean anything - say **nobody dies**.

There is a tendency in business to use fancy names, probably because they sound sophisticated and disguise the fact that we're not sure what we're talking about. This doesn't work properly in risk analysis where we try to take into account our own ignorance (see later). Short, brutal words are good, like nobody dies, no crashes, we make big profits, we don't get sued, I keep my job, whatever.

Positivity

A paragon represents a state of affairs that is desirable or necessary - a sort of goal. We deal in goals, not problems. This is consistent with the philosophy of Dependency Modelling. But it's more than that. To mix good and bad paragons would require our logic to include NOT, NOR, NAND etc. which would be very confusing, and risk analysts are not always mathematicians.

The desirable goal might simply be the requirement for things to stay as they are, i.e. the absence of problems. For linguistic reasons some goals can only be expressed as the absence of something. It's fine to have a goal that means no earthquakes, or no financial downturns, or no referendums.

Abstraction

A paragon is abstract, not material. So a goal might be to "keep out the bad guys", not "an access control system" because we can have an access control system yet still fail to keep out the bad guys.

Keeping out the bad guys is the goal and the access control system is merely hardware used in an attempt to achieve it. We may tick a box to say we have an access control system, but this is not the same as saying we have achieved our goal of keeping out the bad guys.

The insistence on all goals being abstract states of affairs prevents risk analysis being hijacked by, or confused with box-ticking.

No Box-Ticking

If a paragon meant the mere existence of say an access control system we could give it a 100% probability of success, because success would merely mean existence. We could then tick a box on a form and say to ourselves “done that”. This would tell us almost nothing about the risks.

The risk resides in *precisely* the fact that the mere existence of the physical entity does not guarantee the goal. This is why goals are abstract.

Strong Definition

As well as a name, paragons have a *meaning* expressed through their definition. The name cannot convey much information, but it alone appears on the diagram. It is important not to confuse the definition of a paragon with its name, which merely serves as a shorthand reminder.

Paragons must be precisely defined. When we said they were *abstract* we didn't mean *vague*.

A well defined paragon would be along the lines of

To prevent attackers or their agents from entering company premises, or introducing malicious software into company computers or networks at any time between midnight January 1st 2011 to midnight January 1st 2012, or to detect their presence in a sufficiently timely manner as to be able to prevent any of the losses catalogued in list 1.

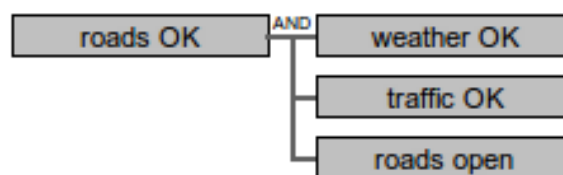
We've already seen an example where we described the significance of the paragon *industrial harmony* as *the state of affairs whereby the likelihood of roads being open and fuel being available are not adversely affected by industrial action*.

A *state of affairs* is a good opening phrase in a definition. Another good template is *the effectiveness of (whatever) in doing (whatever) in such a way as to achieve (whatever)*.

Having the paragon definition right is vital to the accuracy of dependency modelling. Many models are flawed due to inconsistent definitions, or to the use of the same paragon at several points in the model, each one requiring a slightly different definition.

Consistency

Another way to view the definition of a paragon is to observe that each paragon is *defined by its relationship* with its dependencies.



Roads OK

Take for example the paragon *roads OK* in the above fragment. We can write a definition for car OK to be anything we like, but *in reality the paragon is defined as*

roads_OK = weather OK AND traffic_OK AND roads_open

Unless the definition written down by the model's author boils down to this then the model is flawed and any inferences drawn from it are unreliable.

A good check on consistency is given by the *Cover-Up Test* described later.

Being fussy about the definition is more than just pedantry. The precise definition of a paragon pins down the statistics we assign it. If we were to change the range of conditions under which it would

be deemed to be a success this would change its probability.

The **success of an air-bag** might mean anything from its existence (100% probability) to “its ability to deploy 7 litres of gas within 2.5 milliseconds of a deceleration of between 6G and 190G at any temperature between -10 and +50C for at least 10 years after installation” (60% probability), to “it's ability to prevent lacerations and bone-fracture in a 75 Kg relaxed male during any collision with deceleration of more than 6G” (29% probability).

Aspect

We have seen that paragons represent an abstraction and there is **not** a one-to-one correspondence between a paragon and any physical entity. If we are concerned with say a nuclear power station it might be imagined that we could somehow devise **the nuclear power station paragon**. Not so.

There are many possible abstract goals associated with a nuclear power station and we could be interested in any one of them. For example, one aspect might be its success as a source of electricity. Another might be its ability to prevent the release of radioactive particles into the environment; another might be its effectiveness as a supplier of local employment; another with the speed with which it could respond to sudden demands; another might relate to its cost-effectiveness as an energy source and so forth.

Each of these different aspects would need to be the subject of the definition of a different paragon. Each such paragon would have different dependencies and have different statistics. It's all part of getting the definition right.

A particularly common error when building models is to use one paragon to represent different aspects of the same physical entity.

We repeat, there is no such thing as **the nuclear power station paragon**.

States

A paragon must have at least two, distinct, named, ordered states, indexed in increasing order of desirability, which typically represent degrees of attainment of the underlying goal, from non-attainment (failure) to full-attainment (success). This is necessary to allow the automatic inference of whether a paragon has succeeded or failed and to measure risk, and to give meaning to conditional probability tables.

Statistics

These states must have meaningful, statistical probabilities which, if the paragon has immediate dependencies, are conditional on the states of those immediate dependencies. Values of 0 or 1 are meaningful, permitted probabilities for paragons with dependencies but not for uncontrollables, since in the latter case they are equivalent respectively to non-existence and non-dependence.

The statistics may relate to the reliability of something, such as the effectiveness of a countermeasure, in which case it would be the proportion of occasions in which it succeeds.

Alternatively it may be the absence of some undesirable state of affairs, such as absence of attempts to break in. Such statistics must be related to a clearly stated period of time, because in a short enough time-frame nothing happens, while in a long enough time period everything possible will certainly happen. To say the probability of no break-ins is 0.99 is meaningless. Do we mean during a second, a day, a week, a year, a millenium?

Independence

Uncontrollables must be statistically independent. Any statistical interdependence must be explicitly expressed by cross-linking with one or more common dependencies, which may entail introducing extra paragons.

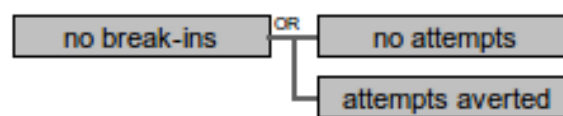
For instance, recall that we introduced a correlation between *roads open* and *fuel available* by arranging that they shared the mutual causal uncontrollable *industrial harmony*.

The Cover-Up Test

A good test of whether we have the definitions of paragons logically consistent, and to check if the building block in which it is embedded is logically coherent is the **Cover-Up Test**.

For example we might be modelling the security of a vault and wish to include the relationship between attempted break-ins and countermeasures.

We might introduce the paragons *no break-ins*, *no attempts* and *attempts averted* in this building block:



Break-in model

This diagram is our statement about the relationship between prevention, attempts and countermeasures.

The definitions of these paragons might be something like this:

<i>no break-ins</i>	The state of affairs whereby either there are no attempted break-ins to our vault or such attempts are successfully averted by countermeasures.
<i>no attempts</i>	The state of affairs whereby no attempts are made to break into our vault.
<i>Attempts averted</i>	The state of affairs whereby all attempted break-ins are deal with in such a way that <i>no break-ins</i> attains the success state.

Notice that this is a **tautology** - a set of self-reinforcing statements with redundancy. It is the redundancy that guarantees the correctness.

The **Cover-Up Test** is simply to cover up one of the definitions and ask yourself (or someone else) to supply the missing wording in such a way as to make the set of definitions a tautology.

This may seem rather obvious, but it is common for beginners to throw together ill-conceived models and expect them to provide accurate answers.

Note too, the abstract nature of the paragon definitions. There is no paragon representing a vault or a countermeasure for instance. They are all **states of affairs**.

RISK TRADE-OFFS

Because we are dealing with a model of an enterprise and not the enterprise itself we can easily answer the what-if question: “What would happen if such-and-such were different?” We don't have to modify enterprise, just the model. For each variation we can evaluate the risks.

We have seen there are many ways of measuring risk, such as sensitivity, criticality or failure modes. By making changes to our model we can compare the various different ways of setting-up or modifying our enterprise in terms of the different risks each entails.

Each of these variations would have a different cost which an accountant could evaluate. And each would have different values on intangible scales involving subjective judgements of an aesthetic, social or moral nature.

But what has been missing from the equation is how the variations compare in terms of risk.

However we have seen that Dependency Modelling now brings several way of comparing their risks.

This should enable us to make informed decisions involving trade-offs between risk, cost and intangibles. We could for example reduce cost at least increase in risk, or reduce risk at least increase in cost. Or we could increase the aesthetic appeal with the least increase of cost or risk.

Incidentally reducing risk does not always involve cost. We have already seen the garage filling station example in which merely re-arranging existing capital assets reduced risk at no significant cost.

Model Building Example

If we analyse a large enterprise such as the infrastructure of a country the immediate dependencies might be things like the machinery of government, armed forces, security services, telecommunications systems, transport systems and so forth.

If instead we were concerned with the success of running a village fête our top level paragon would involve the safety of the events, the insurance, recruiting stallholders, printing of tickets, reliability of bouncy castles and so forth.

If we pick a household our immediate concerns might be weatherproofing, sanitation, cooking, relaxation, security and so on.

We quickly see that there is no such thing as a typical example. But whatever we pick similar issues will arise.

However if we pick something of serious concern to everyone such as National Security, the reader is likely to focus on whether the model reflects his particular concerns, rather than on the modelling issues themselves. Therefore we will pick something more neutral that most people can understand but few will have issues with.

We're going to pick the successful completion of the flight of a passenger-carrying jet.

We want to restrict the complexity of the model so we'll limit ourselves merely with the flight from the moment the passengers are seated on the plane to the moment the plane comes to rest at its destination.

We ask ourselves what would have to happen for the flight to be deemed successful. In other words what would be the definition of our main goal paragon, *flight OK*?

We might decide that a successful flight would entail the following requirements:

<i>no crash</i>	The plane must not crash
<i>no bomb</i>	No bombs must explode on board
<i>no hijack</i>	The plane must not be hijacked
<i>takeoff ontime</i>	The take-off must not be seriously delayed at the last minute
<i>land ontime</i>	The landing must be more or less on time
<i>tranquility</i>	Passengers must not get into fights or cause disruptions
<i>smoothness</i>	The ride must not be bumpy
<i>no stack</i>	Air traffic control must not require the aircraft to stack for more than 15 minutes
<i>no diversions</i>	The flight mustn't be diverted to another airport
<i>food safe</i>	There must be no food poisoning
<i>food hot</i>	The food must be hot
<i>entertainment ok</i>	The in-flight entertainment electronics must work OK

We could go on, and different analysts would have different lists, but that's enough for this example. Basically we want all those things to happen before we deem the flight successful, so an AND relationship suggests itself.

The consistency rule tells us that the main goal paragon that results from this choice is in effect ***defined by*** its dependencies, so by definition a successful flight is one with all these attributes.

We have an immediate problem here. The attribute ***plane must not crash*** would probably be regarded as much more important than ***entertainment must work OK***. If we lump them together in an AND relationship they will be forced to have equal weight.

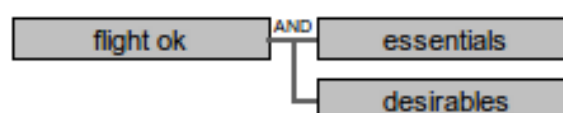
There are at least two ways we can handle this. Firstly we could let ***flight OK*** have three states with meanings something like:

- State 0 = ***failure***
- State 1 = ***at-least-there-wasn't-a-crash***
- State 2 = ***success***

A second way to have ***flight OK*** have two immediate dependencies representing ***essentials*** and ***desirables*** and to let the former cover the life-threatening aspects while the latter handles the niceties.

This gives us two goals rather than one, but if we feel this is aesthetically unsatisfactory we can always tie these two together for the sake of appearances as sub goals of a main goal ***flight OK***, but of course the sub goals are really the only important ones.

Like this:

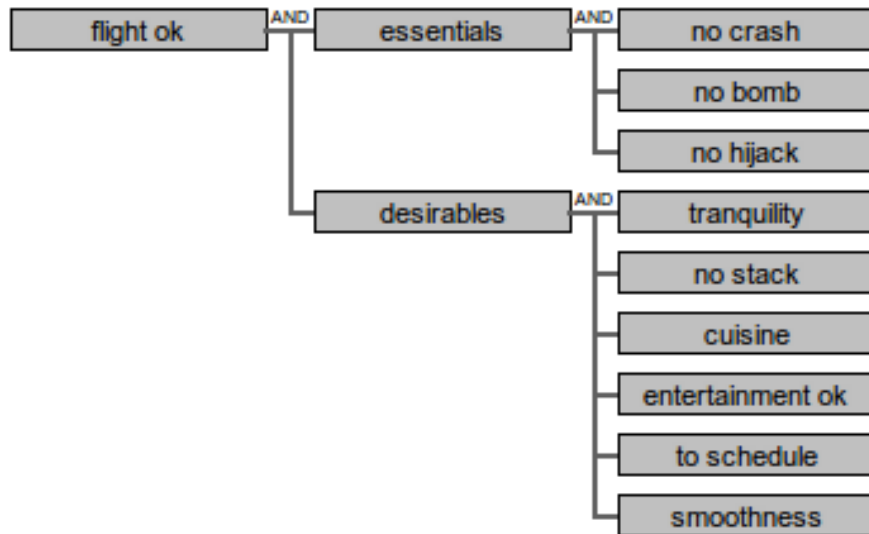


First level

There is nothing wrong with either approach, they are just different styles. Here we'll pick the second because if we introduce a 3-level paragon we make failure mode analysis problematic.

Let's see if we can partition some of the other paragons together.

Under essentials we'd probably group **no crash**, **no bomb** and **no hijack**. Everything else would come under **desirables**, like this:



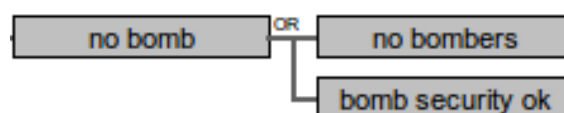
Second level

We'll lump **takeoff ontime**, **land ontime** and **no diversions** together under **to schedule**.

Let's fill in some details.

When modelling a something like **no bomb** – meaning that no bomb is brought onto the plane and goes off during the flight - we would like to represent in our model the tension between potential terrorists and the security specialists trying to thwart them.

We can do this by introducing two paragons, **no bombers** and **bomb security ok** and use an OR relationship type countermeasure that we saw earlier. **No bombers** is the desirable quality that no bombers make attempts to bomb the plane – an uncontrollable. **Bomb security ok** is the desirable quality that the security specialists manage to thwart the attempt. We simply arrange these in an OR relationship. That way we can separate out the threat and the countermeasure, and give each a probability. Here's the result:

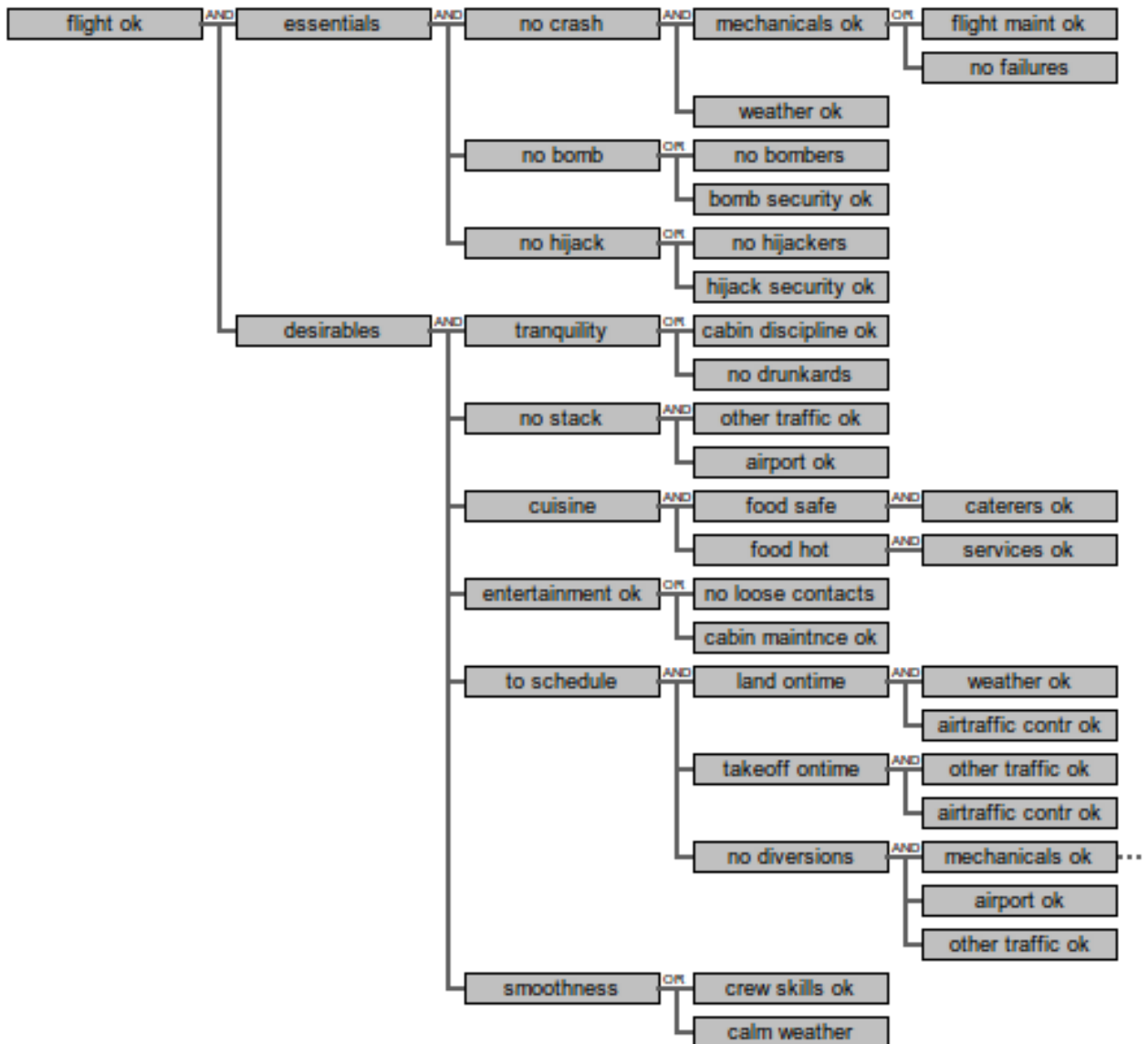


no bomb paragon

We next ascribe the success of the paragon **no crash** to a combination of the plane mechanical parts being in good shape (**mechanicals ok**), and the weather (**weather ok**). In turn **mechanicals ok** is a paragon with dependencies the effectiveness of flight maintenance (**flight maint ok**) OR **no failures**, in an OR relationship.

Notice again the use of the OR relationship to represent a threat and a countermeasure with the threat being inverted into a goal in effect as as no threat.

We continue along these lines, and leave the reader to work out the details for himself. We arrive at something like this:



flight ok

It's important to note that this is a very simple model that does not go into much detail. We could expand uncontrollables by giving them their own dependencies, thereby drilling down as far as we like. The depth, or number of layers in a model is governed by the purpose for which is developed.

To make it more interesting we've made the success probabilities of the uncontrollables unrealistically pessimistic. Nothing has a higher success rate than 0.99.

At first sight the probability of a successful flight is scarily low, around 60%, but on inspection this is almost entirely due to the success probability of **desirables**. The success probability of **essentials** is about 99%.

For most practical purposes we can regard this as two models, loosely connected for convenience onto one diagram, one each for **essentials** and **desirables**. However they're not statistically independent since they share the paragons **weather ok**, **mechanicals ok**, **flight maint ok** and **no failures**.

Flight model flaws.

This model has a number of flaws of varying degree of seriousness. However rather than just fix them, we'll give the reader a chance to see how many he can find.

(Give up? Then here are some faults in the flight model, we identified earlier!)

The paragon **mechanicals ok** is used twice, as a dependency both of **no crash** and of **no diversions** and it probably means something slightly different each time. While many mechanical faults might affect both paragons, there would be others that would only affect one.

The paragon **weather ok** is also used twice, and clearly with different meanings. It is a dependency of **no crash** and **land ontime** and clearly different kinds of weather are implied.

Similarly **airport ok** is used twice, once as a dependency of no stack and again as a dependency of **no diversions**. In this case it is possible that the same kind of problem could have these two distinct outcomes. For instance closure of the destination airport could cause both effects. So this is a less glaring error.

Similarly **other traffic ok** is given as a dependency of **no stack**, **takeoff ontime** and **no diversions** and it doesn't necessarily mean the same thing each time.

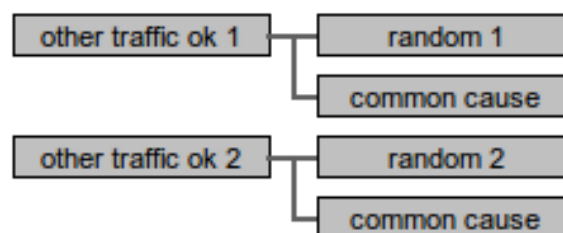
Again **air traffic contr ok** is common to **land ontime** and **takeoff ontime**, possibly with different meanings.

When we re-use a paragon as a dependency of two dependants, these dependants become strongly correlated, i.e. when the one fails the other is much more likely to fail.

This may be exactly what we intended, but then again it may not.

We can take precise control of correlations by using different paragons for the different contexts, such as **other traffic ok 1** and **other traffic ok 2**. These two would then be uncorrelated.

If we want to introduce a carefully controlled amount of correlation between them we can do so like this



Controlled correlation

Here **common cause** is common to both **other traffic ok 1** and **other traffic ok 2**, but each also has its own independent random component. The relative contribution of these components can be controlled by the probability tables.

These are very common flaws, and easy to fix.

Tips and Pitfalls in building models

When people first start to build models there are a number of common pitfalls that often occur. They all come down to building a model with logical flaws in it.

It's worth going through a few.

Failure to rip up and start over – sunk costs

People want their model building to be a logical linear process, but that's not how it works.

When you first start to build a model it usually happens that you soon realise there's a much better way to do it than your initial choice. This is fine - just rip it up and start over.

You learn most from the bad models you discard. The person who rips up nine models and presents only the tenth has a much better grasp of the risk structure of the enterprise than the person who only sees the final version.

A dogged insistence in *not wasting effort already spent* is false economy and a failure to learn from experience. It's also incidentally an example of the **Sunk Costs Fallacy** that fuels gambling addiction.

Failure to define a paragon

Many faults really come down to ill-defined paragons. This is an easy mistake to make and comes in many forms. The version we'll look at here is when we hastily create a paragon and use it in several places in the model without noticing that it needs to have slightly different meanings in some places.

For example we might make a statement to the effect that all our widgets will turn out perfect if either there are no manufacturing errors on our widget production line OR our widget quality-control procedures weed them out.

We could probably come up with paragon definitions that would render this statement correct.

But if we don't, it can be awfully tempting to use say **no_manufacturing_errors** somewhere else on the model, where it now refers to a different production line, or to use **quality control ok** in a second context where it refers to something slightly different.

Inappropriate correlation through re-use of paragon

Whenever a paragon is re-used in a model it refers to **exactly the same instance of the same paragon**, not another one just like it.

So if we have a paragon called **interruption_free_electricity** and we use it twice, then **it refers to exactly the same electricity supply**, right down to the same generator and supply network. Whenever the one fails so does the other at exactly the same moment because they're the same thing. This implies a perfect correlation between events that will make a mockery of any countermeasure exploiting the fact that they should be uncorrelated. If you need two similar paragons that behave independently you must create two, with different names, like **supply1** and **supply 2**.

Definition drift

Sometimes as a model is worked on, the meaning of a paragon subtly becomes changed without the designer noticing. For example a paragon might mean the state of affairs whereby a computer performs faultlessly between two given dates which are common to all paragons in the model. Later when the time-frame is altered the designer may forget to re-adjust the probabilities.

Another example is where a paragon refers to the reliability of a particular computer, then to another computer, and later still to the **set** of all computers in a department

Paragons with just one dependency

Since a paragon plus its single dependency may both be replaced by just one paragon, we should be suspicious of single dependencies as they may point to a risk analyst who doesn't understand what he's doing.

However there are sometimes good reasons to show single dependencies. One is where several paragons each has the same single dependency, i.e. it is a common cause deliberately introduced to

produce correlation. Another reason might be to emphasise a point to a particular audience.

Mixing different philosophical planes

Models can come at various philosophical levels or planes. For example one model might be concerned with the reliability, efficiency etc. of various physical entities and the effectiveness of various procedures. We can think of this as a **concrete** approach.

By contrast there could be a model concerned with the effectiveness of rather **ethereal concepts** such as

SUCCESS = INTEGRITY AND CAPABILITY AND COURAGE AND LUCK

This is an extreme example, but I've seen people writing models this way. There might be a problem tying down the meanings of the various paragons, but given suitably creative definitions it could probably be made to work to some extent, especially if it were used tongue-in-cheek to make a point as part of a staff training program, because then one could get away with sloppy reasoning and it wouldn't really matter.

But what wouldn't work is trying to mix the two approaches, since it would be hard to find any paragons that could join the two approaches together.

Use of non-paragons

Some concepts can't be rendered as paragons. For instance the success of expanding an established business into another country might depend on

- the business climate in that country
- whether certain pre-requisites are legal in that country
- whether staff costs are affordable there
- etc

The second of these, whether something is legal, is a perfectly sensible consideration, but it **isn't a paragon** since it can't be given meaningful statistics. It's either true or false. In a given country It's not true say 80% of the time at random.

Such a model may look sensible but it's sloppy reasoning, and anyone looking at it would realise the author didn't understand the modelling process.

It comes down to the everyday use of sensible statements in common language, and the need to be precise in a risk model: **most sensible statements don't convert to risk models.**

Use of ready-made lists

Organisations are fond of compiling lists. Beware of adopting them as paragon statements.

For example an organisation might have a department hierarchy diagram and it would be tempting to make a top-level statement along the lines

ORGANISATION_OK = DEPT1_OK AND DEPT2_OK AND DEPT3_OK AND DEPT4_OK ...

The reality is this will turn out to be a big mistake unless you pay a great deal of attention to the precise definition of each paragon. You would have to define exactly what is meant by say dept1_OK. It would have to include defined states each with a probability; a reference time-frame

for those probabilities; it would have to be spelled out exactly what constitutes success and what failure; it would need a precise list of its dependencies; you could not use the same paragon somewhere else to have a slightly different meaning, and so forth. Getting it right could be a nightmare.

Also by implication the success of the organisation is **defined** as the success – whatever that means – of each of the departments, which might be not what was intended.

Basically don't do it.

Mission statements and respectable goals

Catchy statements don't usually make good paragons. Many organizations have a mission statement. It is usually a mistake to adopt it as the enterprise's main goal paragon.

For instance a mission statement for a hospital could be along the lines

To provide a quality service and value to the health authority while respecting the dignity of our patients and treating our staff with courtesy and respect.

However if you ask the senior management in a hospital what risks they fear most they are likely to be

- Closure due to a change of government policy.
- Losing budget due to failure to meet government targets
- Losing out to competition from the private sector
- Giving permanent brain damage to a one-year-old child and being sued.

Another organisation's management might have as its aims

To provide value and courtesy to customers, returns to shareholders and respect to employees

But in reality their true concerns might be

- The government will declare us a monopoly and we'll have to merge
- A change in government policy will wreck our profitability or stop us paying our senior staff big bonuses
- Our CEO will lose interest and sell off the business to an organisation we dread
- Lower labour costs in the Third World will lead to closures

These don't get a mention in the list of goals.

A consciously instrumented risk management policy stands a fair chance of achieving its aims. But these may not be what the management really wants. Unless your overt goal reflects your true concerns you may achieve what you said, not what you meant.

Be careful what you wish for.

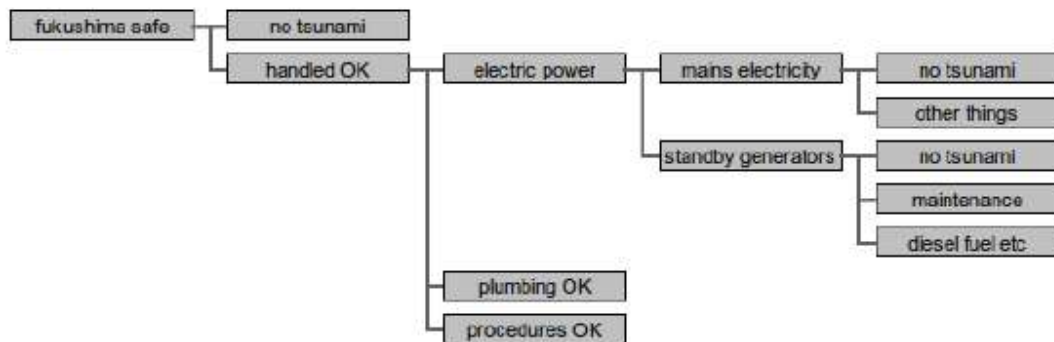
Finally, how would this have helped the Japanese Operators (TEMPCO) better understand their dependencies on the adequacy of the safeguards in place?

Just how independent and individually resilient were their lines of defence?

A “Dependency” Risk model of Fukushima

It's very straightforward to use dependency modelling to show up the obvious flaw in the original design concept for Fukushima.

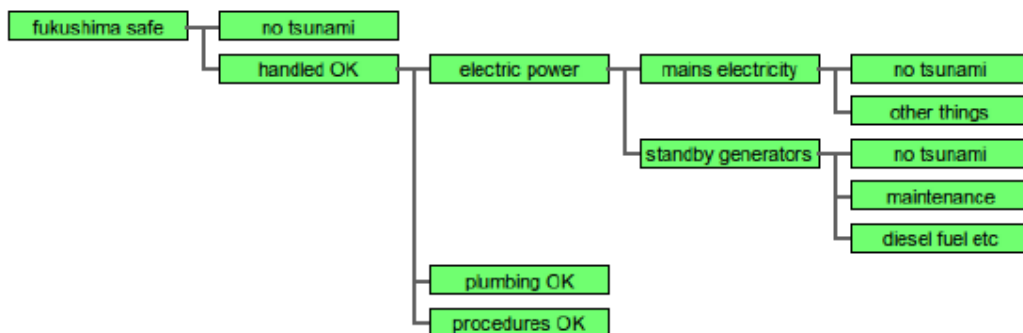
Here's a simplified model:



Model

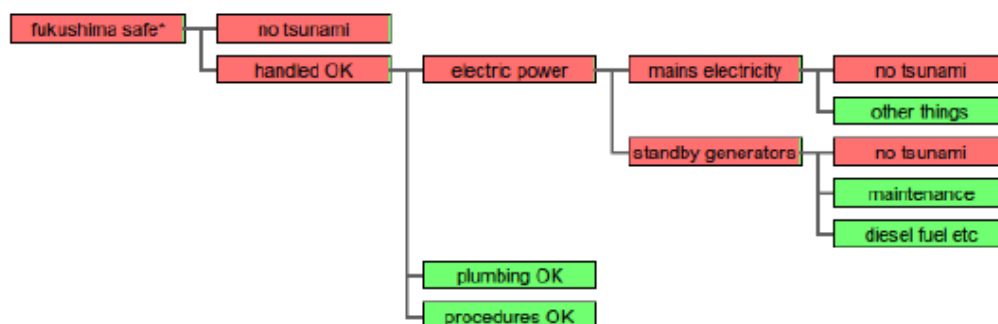
The probability of tsunami has been set to 1 in a million.

Predictably the probability of disaster is small as witnessed by the vast expanses of green, and no visible red:



Probabilities of the states

But probe a bit deeper and the flaws show up. Here's what you get, given that a disaster occurred:

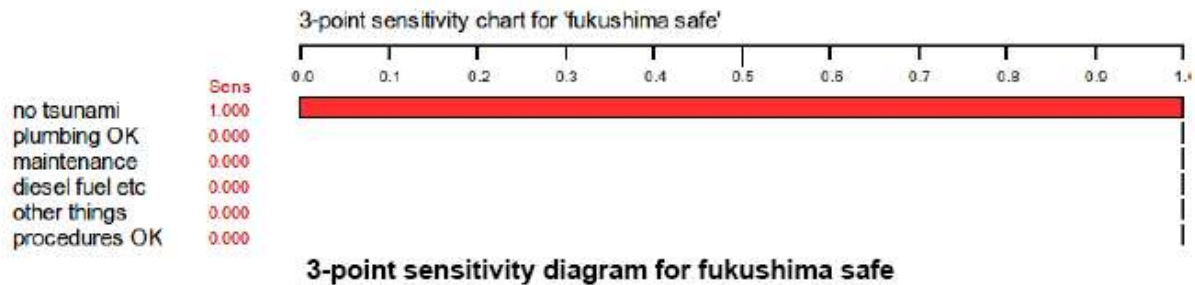


Probabilities, given that fukushima was a disaster

This is an *a-postiori* result, showing the most likely causes of a disaster, given that it has happened.

Risk

And finally, the *coup-de-grace* the 3-point-probability chart:



If ever there was a diagram that justified this methodology as a measure of risk, it's this one.

It was a:-

Classic Common Mode Failure

In our dependency terms, the disaster seems to be due to the fact that the critical countermeasure paragon –

“the ability to keep sufficient cooling water over the “hot” fuel rods ” –

- depended on electricity,
- **And** both the mains and the standby generators were themselves casualties of the same disaster.
- **And** the battery backups were located and failed in vulnerable basement locations
- **And** the cooling ponds leaked from seismic shifts

This is the classic common mode failure and, although some of the defences designed for Richter 8 actually survived Richter 9, they seem to have forgotten about Tsunamis.

Where would you spend your resilience money?

Dependency modelling would have enabled them to concentrate on showing (achieving?)

- Adequate Resilience rather than Acceptable Risk.

Discussion

if they forgot about Tsunamis, they would not have it in a dependency model either would they?

Agreed, but, their fault trees are, by default, inverse dependency models (albeit with limited causality scope), that would have had coolant systems, power, backup generators etc. Again, this raises questions others have also voiced in the past, what distinguishes our approach from the rest? What distinguishes us is the ability to drill down to critical leaf dependencies (i.e. Tsunami no more than X metres); and see how many of these leaves there are; and how they dominate all the other criticalities.

The fault trees can find common causes, but only if you thought them all through in detail from the bottom (horseshoe nail) up!

Now our “top down” approach to establishing your system integrity (Kingdom), actually depends, not only on all these clever safety systems and their back ups, redundancies etc., but whether or not they are able to meet the success criterion of surviving a 60 foot Tsunami wave here and here and here!”

Our system can also draw the Tsunami contours in real time and knock over all these separate vulnerabilities at the same time (the classic definition of a simultaneous common mode failure!)

What if nobody knew that tsunamis existed (they were a completely un-known phenomenon)?

That is where dependency modelling is really useful in being goal oriented. You don't need to specify the cause of failure, only the goal you are aiming to achieve. If an unknown element comes along and causes your goal to fail, you need to know what the consequences of that are on the rest of your system. And so on ---- so do all your dependents for that matter!

That is the differentiator with Dependency Modelling: “Don’t worry about what might cause the failure – just imagine that a mysterious black cloud* descends...”

All you need is criteria for success. You are then vulnerable if you don’t meet them.

If anything comes along (event or condition) that breaches the criteria your status changes from successful to “Fail”- wholly or partially.

So even if you don’t know what a Tsunami is , but have designed for max water level in plant, and the dependency status is >Max,

Then you fail regardless of ignorance.

We have to live in the real world. There's a limit to how much can be achieved by risk analysis. We can only work with actual knowledge.

Conclusion?

The problem is that people do know about tsunamis and still don't do proper risk analysis.

Glossary

Branch

A paragon that has both dependants and dependencies.

Conditional Probability

The probability of something occurring given (the condition that) something else has occurred.

Dependant

A paragon's dependant is another paragon that depends on it.

Dependency

A paragon on which another (it's dependant) depends.

Paragon

The basic element in a Dependency Model.

Probability

The probability that an event X will occur is the number of instances when X actually does occur expressed as a proportion of the instances when it could occur. For example the proportion of occasions when a rolled dice shows a two is $1/6$, which is therefore also the probability of rolling a two.

Risk

Risk is the degree to which the chances of achieving our goals are affected by things we cannot control, predict or understand.

Uncontrollable, leaf, given

A paragon with no dependencies

References

[George Herbert] (1593-1633) *For want of a Nail* Poem:

“For want of a nail the shoe was lost.
For want of a shoe the horse was lost.
For want of a horse the rider was lost.
For want of a rider the battle was lost.
For want of a battle the kingdom was lost.
And all for the want of a horseshoe nail.”

[Hubbard 2009] Douglas Hubbard "The Failure of Risk Management: Why It's Broken and How to Fix It, John Wiley & Sons, 2009.