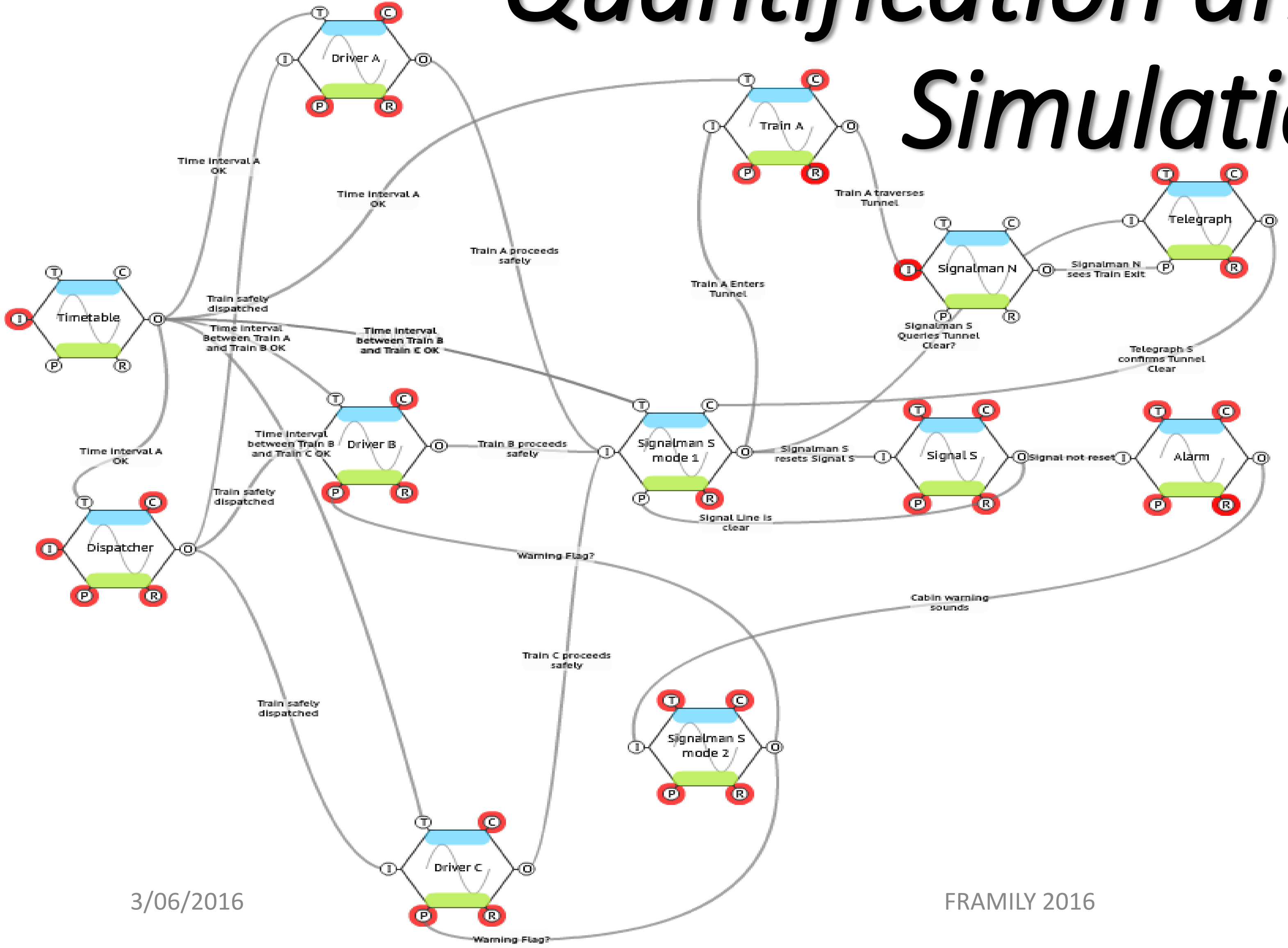
FRAM as a “FRONT END” to Quantification and Dynamic Simulation



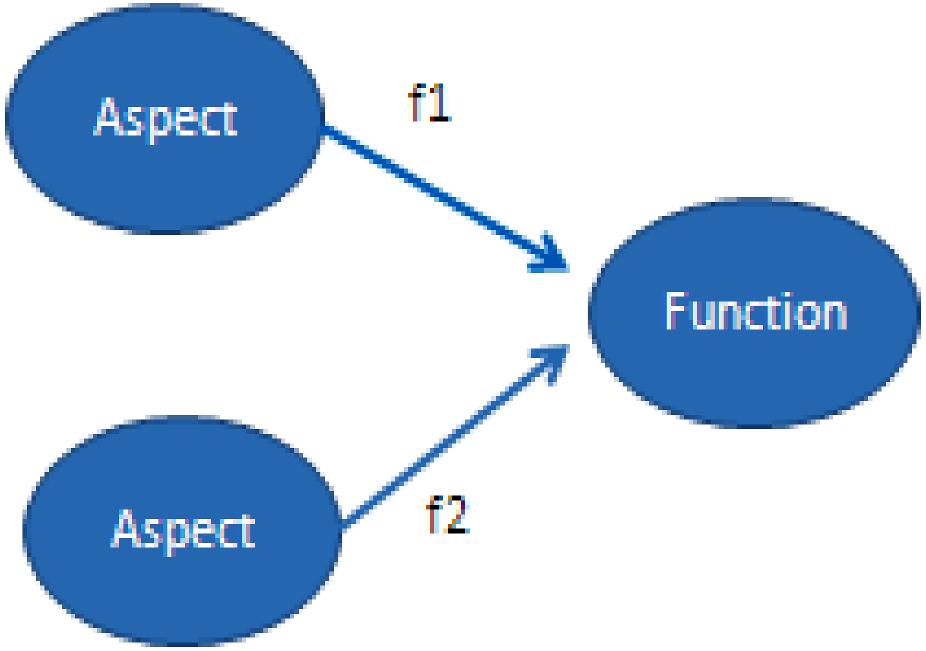
The next steps?

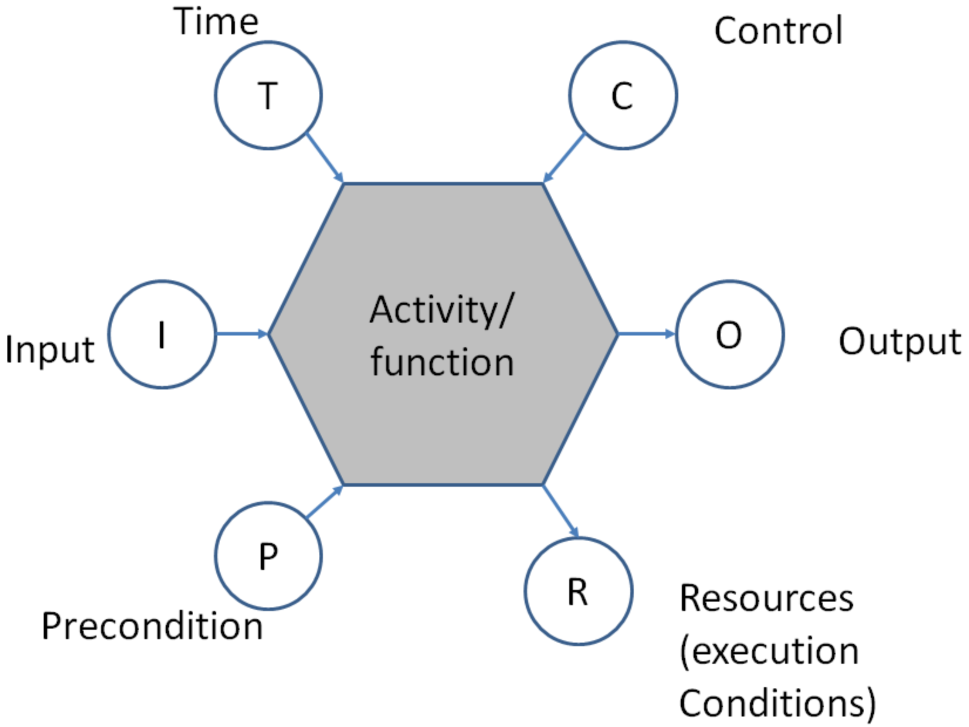
David Slater
Lisbon June 2016

Further Development of FRAM

- At its present stage of development, FRAM is meant as a purely qualitative visual aid to help groups and teams develop a common picture of how a particular project actually works (as opposed to how it was meant to work), in practice.
- In an earlier paper (vanKleef) proposed a further development of the methodology, to formalise the relationships between the FRAM functions so that logical and mathematical tests of completeness and correctness could be applied and perhaps more quantitative results obtained, for the better appreciation of the cost effectiveness, etc. of the benefits discovered.
- This presentation outlines a way of utilising these van Kleef networks as a “Front End” to simulation as Dynamic Bayesian Belief Nets

One way of representing an array of FRAM entities, is by considering the FRAM Function as a "node", whose behaviour is influenced by its "parents", called Aspects.





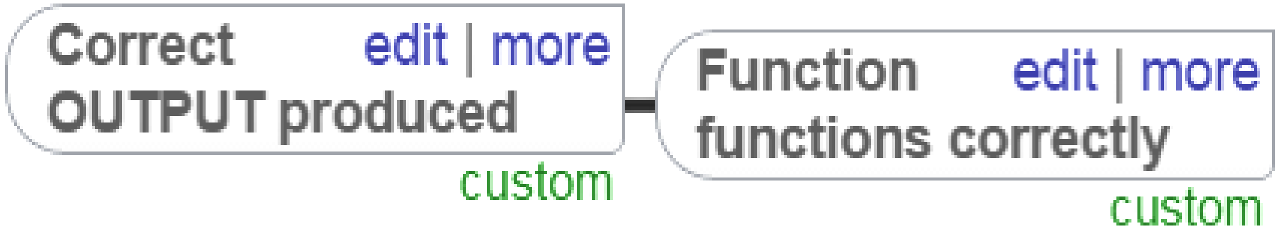
These Aspects are always produced, by other Functions (their Outputs), which can be affected by Aspects from still other Functions, so that Fractal-like, an interconnected network can grow stochastically (or biologically?), like a self replicating crystal structure , depending only on the availability and "states" of these external influences and the Functions' receptors.

Formalising / Quantifying the underlying Structural relationships

- The current development version of FMV can take the FRAM Functions it has built and transform them into a simple Bayesian Belief Net. The FRAM “Function” is thus a conventional BBN node and its associated Aspects are its “parents”. For the FRAM analysis, this node will always produce an Output which becomes a parent to other Functions/ nodes as required by the initial FMV model.
- The “Orphan” nodes (called Background Functions in FRAM) can still produce Outputs, but they are no longer necessarily controlled by Aspects in the current model. They thus form the “boundaries” of the study space. But in the FMV input screen, we can still specify what kind of Function it is and how reliable it will be in producing the required Output.

Background Functions

- The simplest Background functions with no impinging Aspects are modelled as single nodes.

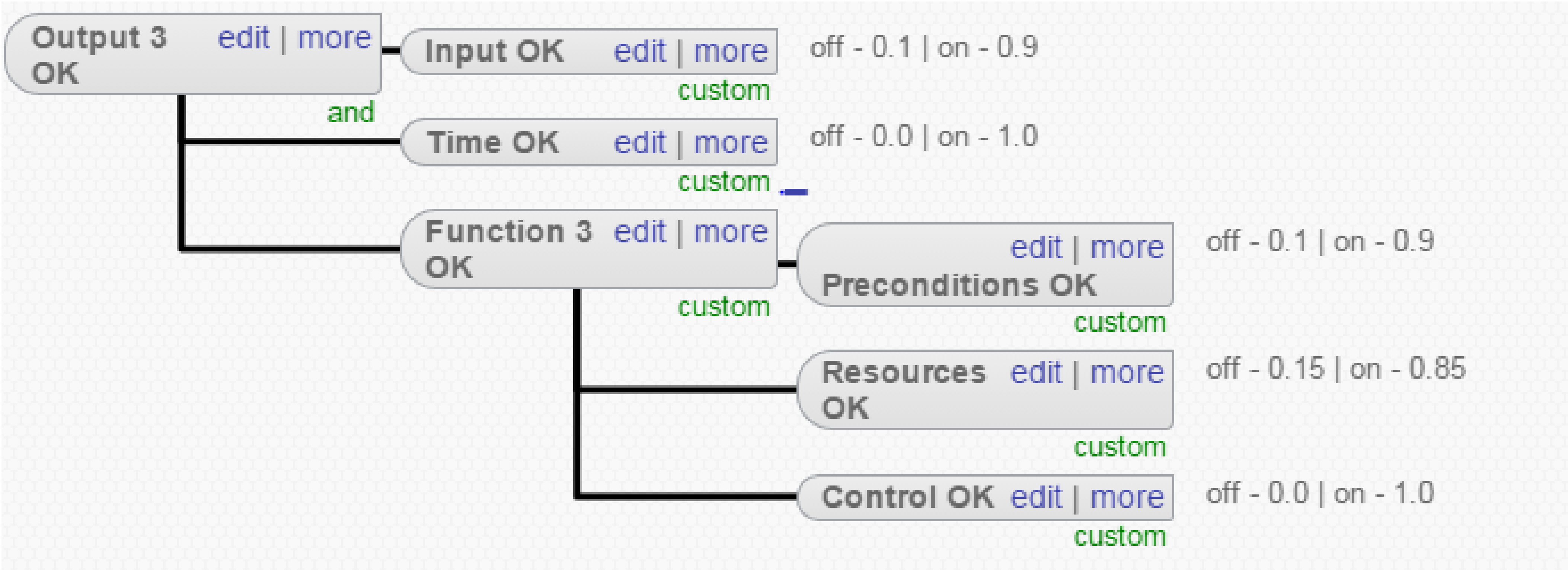


- Note the Bayesian formulation still allows us to specify the ENDOGENOUS properties of this Function, despite there being no EXOGENOUS Aspects in this case. The leaf probability is then the function functioning, not the output.

FOREGROUND FUNCTIONS

- Here the construction of the BBN Functional unit (the translation of the FMV hexagons), must obey vanKleef's rules.
- Without an input the Function can't start.
- The Function needs to be in the allowed time slot to produce an acceptable output
- Preconditions, Resources and Controls must be present initially but can vary during the execution of the Function.
- This requires a basic BBN element as shown, below.

The FRAM Function as a BBN Node



- This now allows us to build the full Bayesian Belief Network for a static BBN as an "Instantiation" at Time step 1.
- The status probabilities of the Aspect nodes are then read from the augmented FMV Table and an overall probability of completing that step can be displayed

Joined up Instantiations

- We can now progress the sequence to the next instantiation (Time Step 2) and update the states of the Aspects as they occur in this step.
- This can be repeated for all the instantiations/ Time steps needed. But this is now in effect a Dynamic BBN (DBBN).
- A DBBN consists of a sequence of sub-models (static BBNs), each representing the system at a particular point in time (time slice).
- A DBBN can theoretically work with the same or with different structures over time.

User Determined Time Slice (Step) Instantiations?

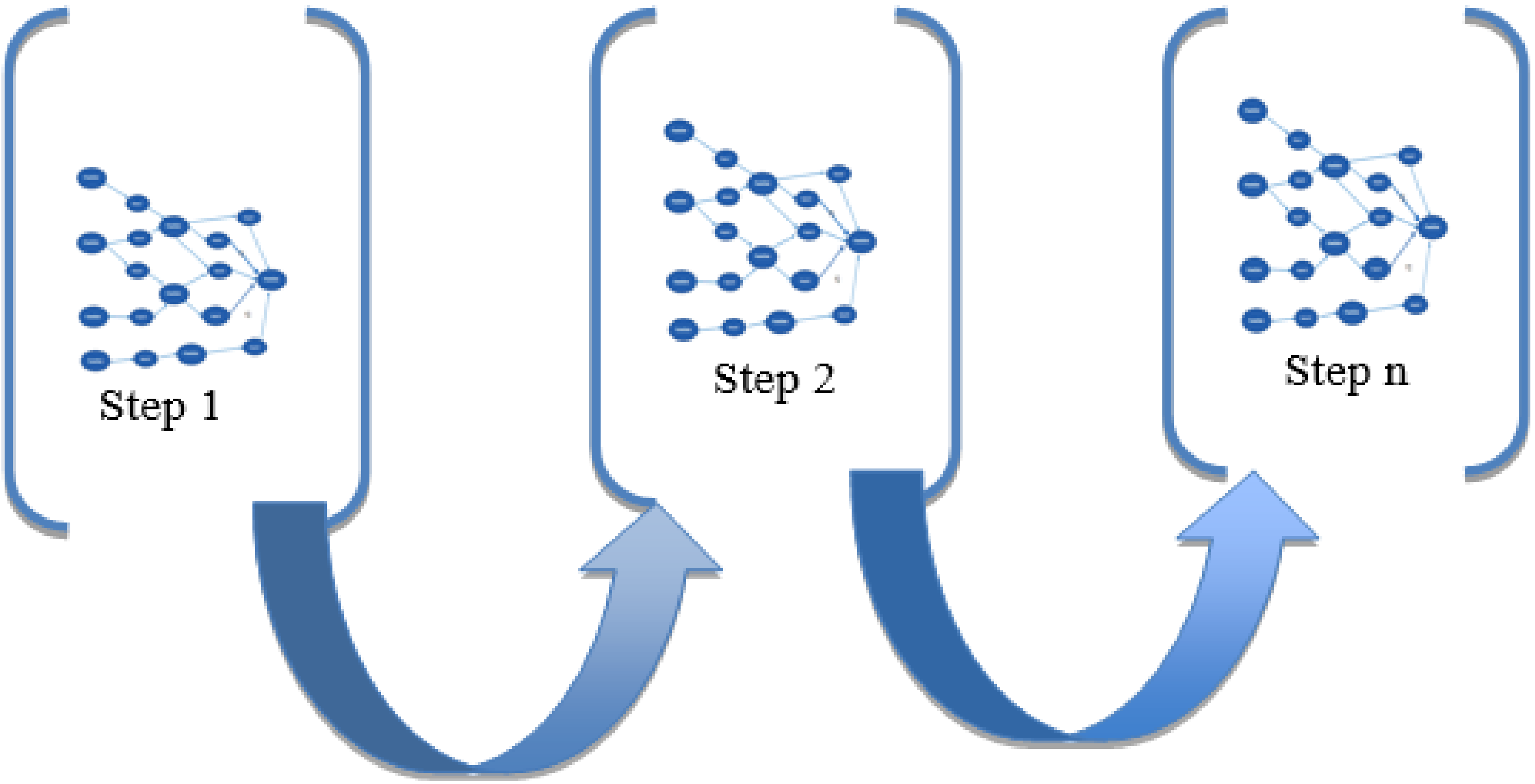
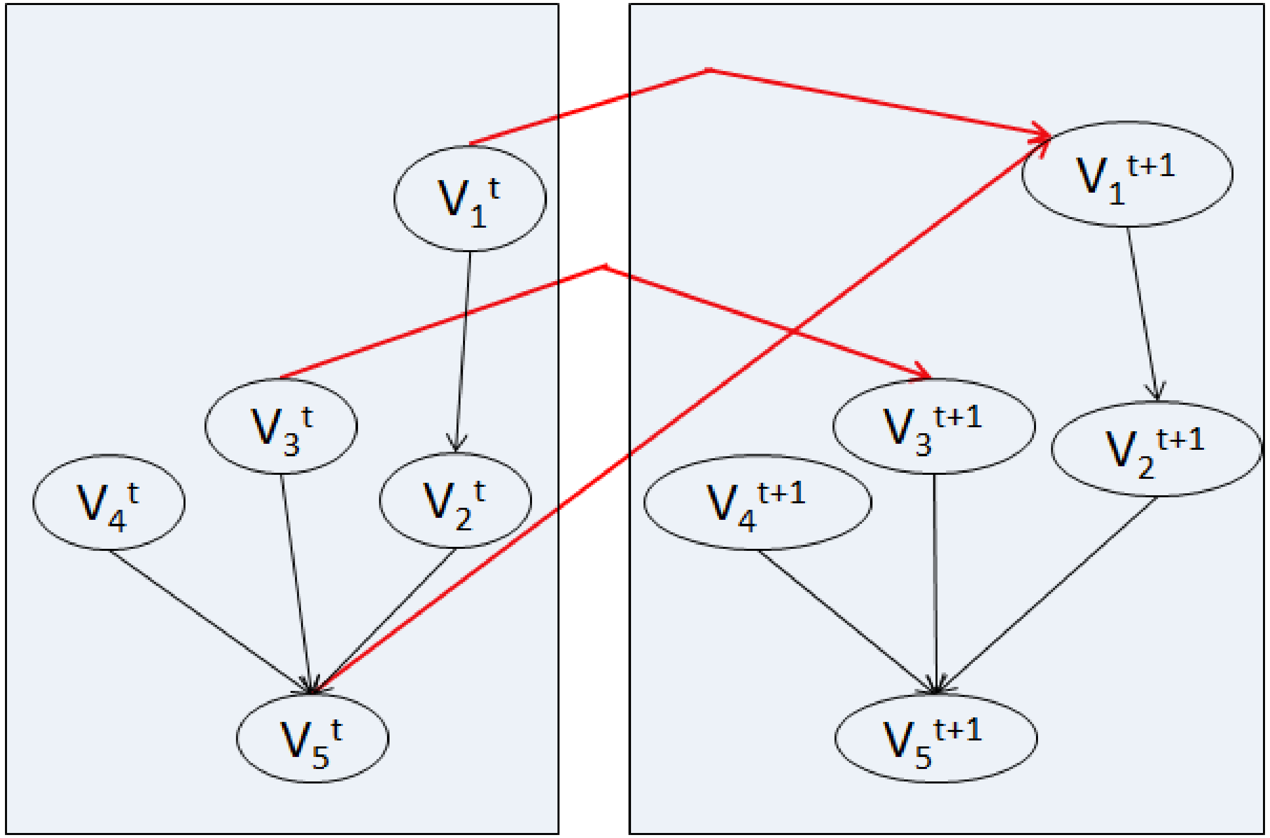


Figure 5 – Network Structure as a function of Time (Slices)

Assumption

- it is assumed that the state of any system described by a DBBN satisfies the first order Markov property, which says that the state of a system at time t depends only on its immediate past, i.e. the state of the system at time t-1. If these two conditions are satisfied, then the dynamic system can be represented by two successive time-slices,



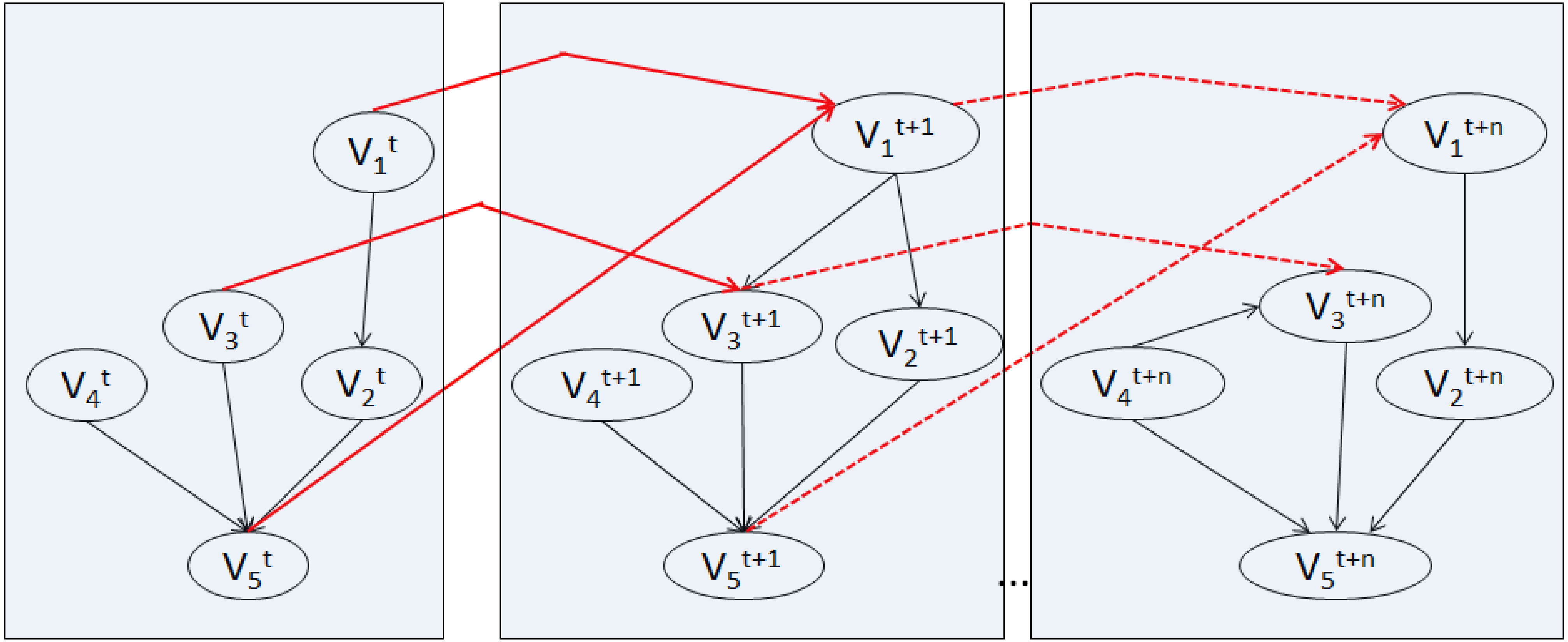
Two time-slices of a DBBN satisfying the Markov property

Variabilities in the FRAM structure for the next instantiation

- The van Kleef rules now specify what the structure of that next step must look like as they can automatically change what functions interact , how and when, such as:-
- Inputs must be available at correct times and for long enough or functions will not operate?
- Background aspects can be environmentally or time sensitive or subject to other demands/ priorities.
- The totality of necessary Resources can be consumed and depleted over time and therefore progressively unavailable for all the functions that need them?
- There are real time windows of opportunity for correct interactions!
- Control is necessary (as imagined) but functions may ignore controls (as is)

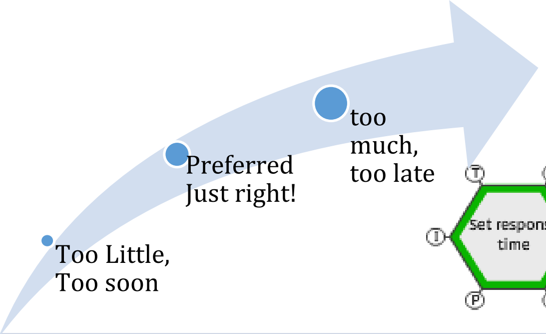
Non Predetermined progressions

- A DBBN can theoretically work with the same or with different structures over time. The Markovian assumption then allows the structure of subsequent steps to be randomly determined by the randomly particular mixture of Aspect states in the prior step.

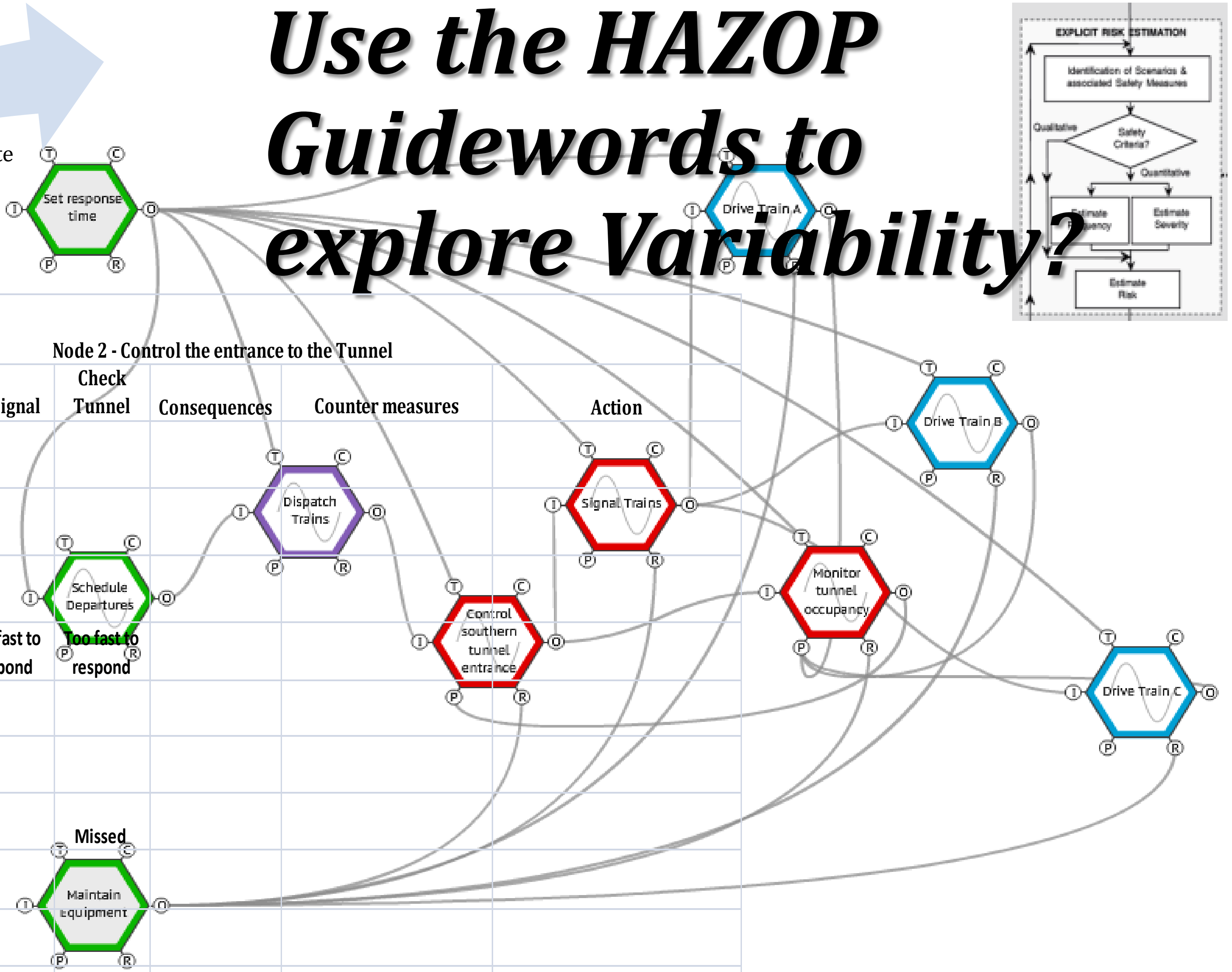


Dynamic BBN with different structures for each time-slice

Use the HAZOP Guidewords to explore Variability?



Function Variabilities	Node 2 - Control the entrance to the Tunnel						
	Control Trains	Signal Train	Set signal	Check Tunnel	Consequences	Counter measures	Action
Control Train A	Consequences						
Control Train B							
Control Train C							
More, Less, None	Too fast	Pass before reset	Too fast to respond	Too fast to respond			
	Too slow	Queuing					
Other Than, As Well As,	Two Trains	Queuing					
Too Early, Too Late	As above	Pass before reset		Missed			
Out of Sequence	Wrong Train						
Set Signal	None	Semaphore breaks					
Signal Train	Wrong signal						
Check Tunnel	Wrong Train						

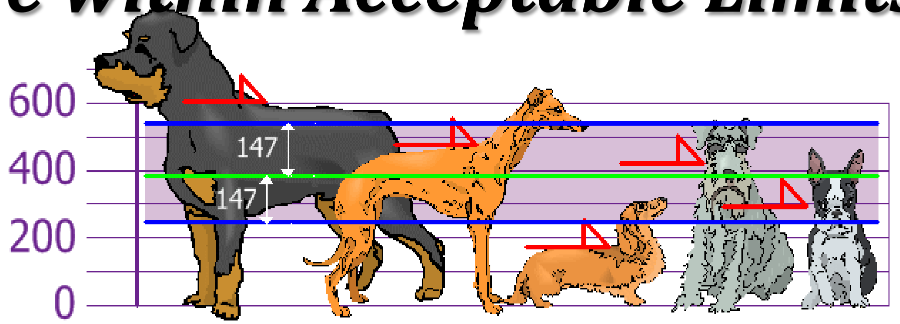


- **FRAM looks for AS IS Variability**

SO WHAT ABOUT SIMULATING THESE REAL LIFE CONSTRAINTS BY ASSIGNING RANDOM VARIABILITIES TO EACH AND ALL OF THESE INTERACTING ASPECTS AND FUNCTIONS?

- The BBN's are capable of being processed sequentially by the Bayesian engine to use (estimates) evidence of satisfactory Aspect availabilities and behaviours, to predict probabilities of each of the FRAM Functions in turn behaving satisfactorily.
- This requires a set of initial background Function output states (Leaf Nodes in BBN speak), which the engine normally acquires from what is essentially the FMV Excel spreadsheet in the cloud – a data harbour.
- These state variables can be either True or False (1, 0) or return a probability of being a 1 or a 0.
- These states, or probabilities, can be generated from the FMV spreadsheets by using the corresponding built in Excel functions and stored as data sets for the BBN.

Then what is the Probability that we are within Acceptable Limits?



Calculate Z, given X, mean and Std Dev

Rand Var X	500
µ Mean	400
σ Std Dev	70
Z	1.428571

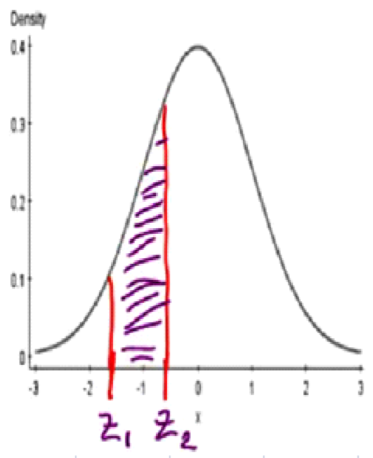
$$z = \frac{X - \mu}{\sigma}$$

Area	
Minimum	Maximum
0.008633972	0.248166091

Acceptable	
Z1	-2.380952381
X1	50
Z2	0.680272109
X2	500
Mean	400
Std Dev	147

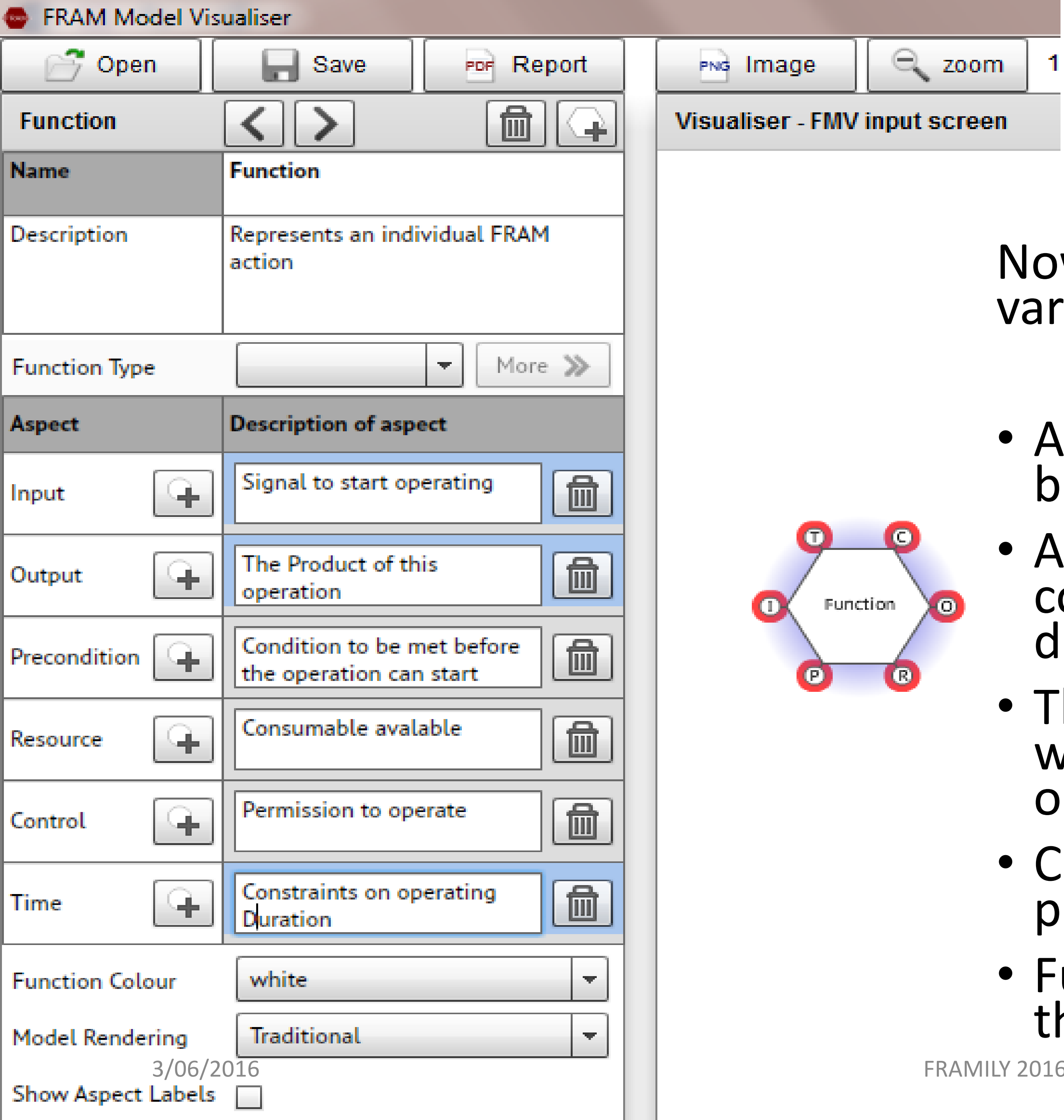
Type; Standard Normal Curve
Mean
Std Dev

PA	0.743199937



- **1. SHAPE?** – What kind of variability are we talking about, what does it look like? (Standard “shapes” – Normal, Lognormal, Triangular, Square, etc.)
- **2. SIZE?** - What is the **“Average”, “Mean”, or “Expected”** value of our “parameter” and the appropriate estimated **“Confidence limits”** in that number**(+/- %?)**?

  (These are approximations for **µ** and **σ**)
- **3. BOUNDS?** **-** What are the Upper and lower bounds of Tolerability/ Acceptability for that parameter? ($\mathbf{X_1}$ and $\mathbf{X_2}$)



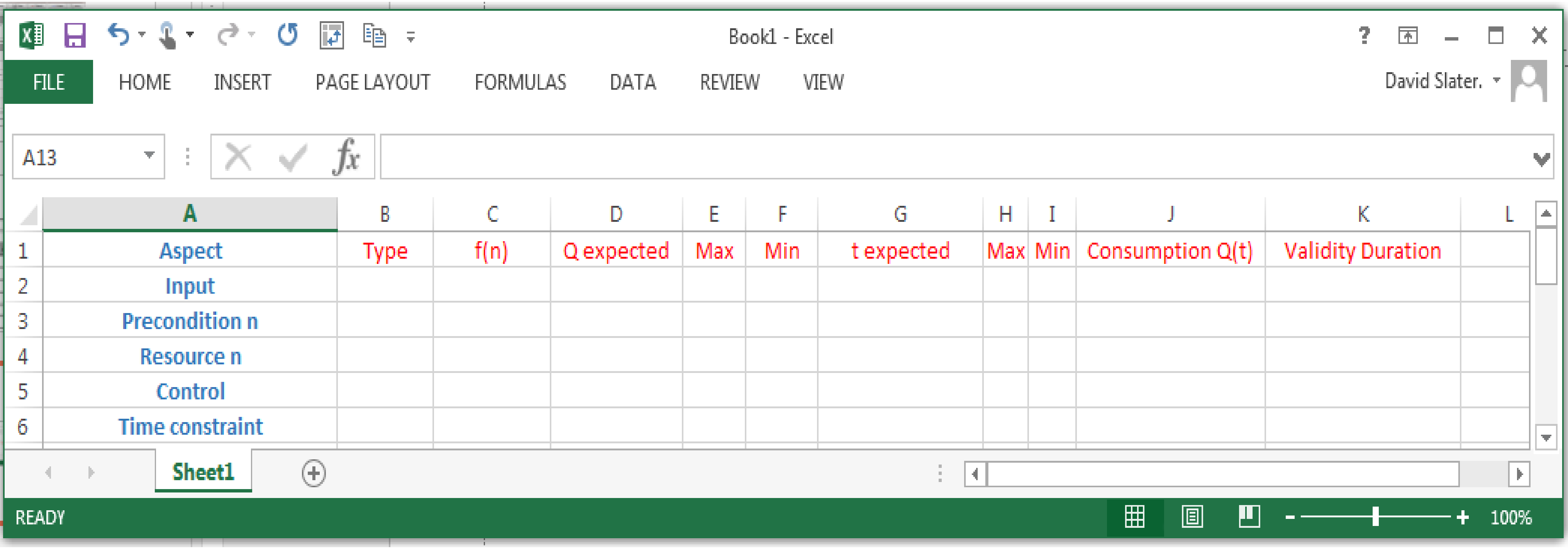
FMV Input Screen

Now add the vanKleef Rules and the variability predictions

- An Input – has to be present (True?), before a Function can Start
- A Resource needs to be present but could be consumed and needs a decay function
- The Time aspect sets a window within which a step must either start or finish
- Control needs to indicate permission, but
- Functions may act out of control if this aspect is not present (True)?

Name of function					**Type**	**Distrbn.**
Description					**Technological**	**Normal**
Aspect	**Description of aspect**	**Expected Value**	**Max.**	**Min.**	***f*(Consumpt,t)**	**(Duration,t)**
Input						
Output						
Precondition						
Resource						
Control						
Time						

Aspect State Probability Estimation

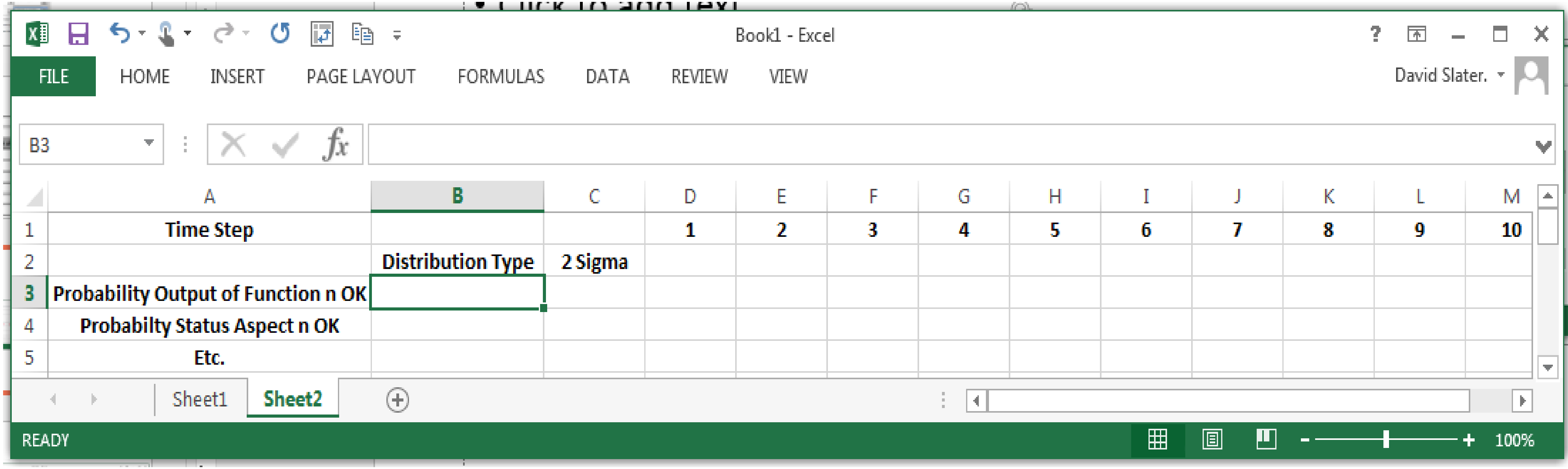


	A	B	C	D	E	F	G	H	I	J	K	L
1	Aspect	Type	f(n)	Q expected	Max	Min	t expected	Max	Min	Consumption Q(t)	Validity Duration	
2	Input											
3	Precondition n											
4	Resource n											
5	Control											
6	Time constraint											

FRAM Function “Endogenous” probable behaviour Estimation

	A	B	C	D	E	F	G	H	I
1	Function Name	Type	Condition	Environment	Reliability	Time needed	L	H	Decay f(PerformanceT)
2	Description*	T	A	H	B	10 secs	5	15	T exp-2
3									

Estimated Probability of a Background Function generating the Output needed in the Correct State



Integrate with the FMV tool

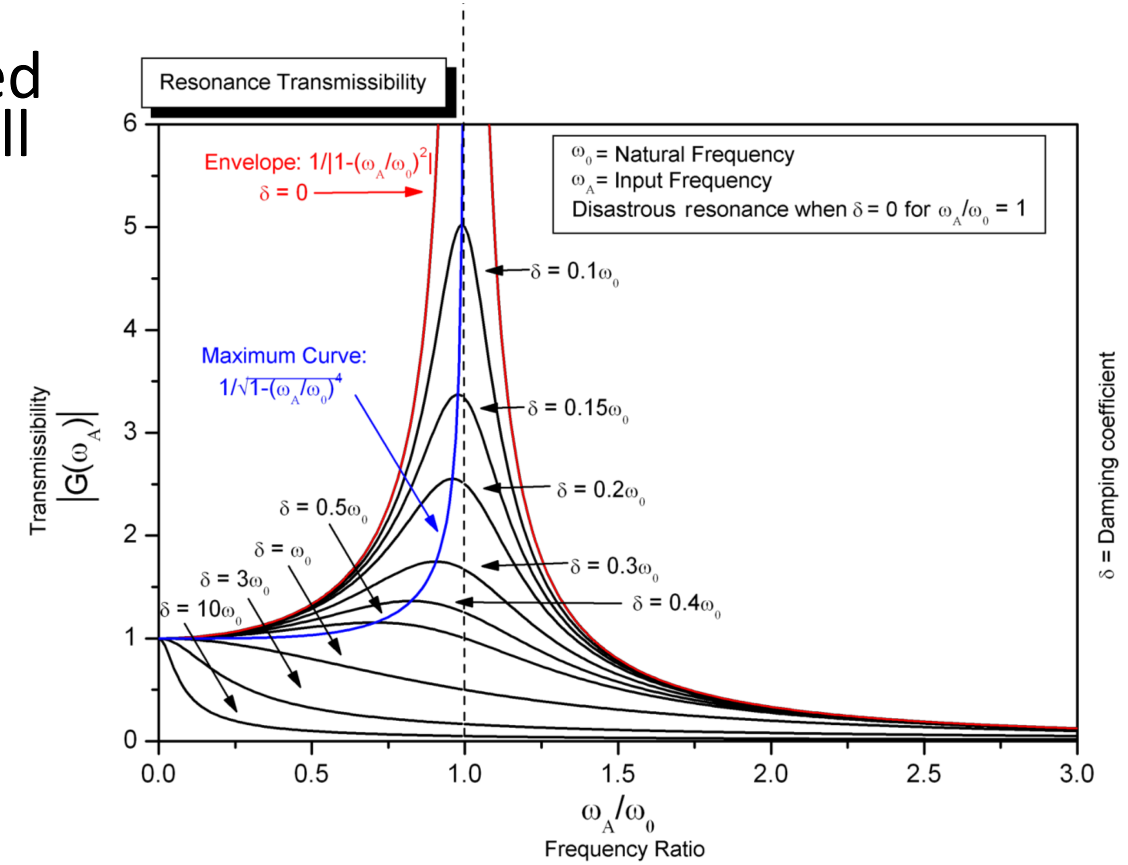
- Click to **add a Function**
- Identify it as hu**M**an, **T**echnological or **O**rganisational (**M,T.O** as usual)
- This determines the endogenous settings on the underlying BBN template
- Characterise the “environment” in terms of pressure or stress (**H**igh, **M**edium, or **L**ow)
- This determines the exogenous settings on the BBN template
- Click on **Aspects**
- For each identified Aspect, in addition to a unique label add **E**xpected and **A**cceptable ($\mathbf{A_1, A_2}$) range of values. This further calibrates the BBN template, as-.
- This describes the shape of the expected distribution of values observed in practice
- Build the first step/ Time slice by completing the above for each Function involved in that step.
- Repeat until the full visualisation of this first step is complete as normal, then:-
- Click on (NEW BUTTON) **Formalise?**
- This organises the inputs above as a set of XML files generated by BBN “Function” templates behind and populated by the normal FMV building blocks.
- In the FRAM visualisation, the “starting” points are normally triggered by the status’s(?) of the background functions .
- In the FMV, this is flagged by the red circles indicating an orphan aspect.
- For each **flagged Aspect**
- I’m suggesting we have a similar input dialogue for these orphan aspects, which asks for most likely expected value(E) and an indication of likely spread (A_1, A_2 - max and min) as before.
- for these aspects, this defines a starting distribution from which the software can pick a random value.
- Click on (NEW BUTTON) **run simulation**

Next Step?

- This in turn generates and stores in the same spreadsheet a new set of values for the next Time step in the sequence; and so on until the Process is completed (or otherwise). Depending on the limits specified in the FMV model building, some aspects will "Probably" fail to propagate into the next step (e.g. run out of time), so as well as intended (predetermined) changes in subsequent instantiations, there may well be unforeseen gaps.
- Similarly, because each Aspect is uniquely identified, it can couple in the network wherever it is called. So for cases where Resources for example are needed by several nodes and whose probability of availability can change with time, then here unexpected interferences can now emerge between steps.

BUT WHAT ABOUT RESONANCE?

- With all the data now accumulated, each instantiation step can be run with a random selection of input probabilities from the specified probability distributions nominated for the Aspects' state variables. Each run will give a series (EXCEL row) of output probabilities for each Function.
- Running a Montecarlo simulation (1000 iterations) should then produce a set of Output probability curves that can be compared to the input curves.
- Anomalies, where unexpected interactions cause Abnormal distributions can then be prima facie evidence of the Functional Resonance phenomenon that this methodology was developed to detect.



Conclusions

- It is possible to model single FRAM instantiations as Bayesian Belief Nets
- Hence an overall (quantitative) estimate of the probability of successfully completing that step can be obtained?
- Successive Steps can obviously be similarly treated (in) as a sequence.
- Aspect Statuses and variabilities can be user specified, collated using vanKleef rulesand/or randomly generated from Excel functions to allow the specification of the BBN node probabilities.
- A further extension into non predetermined Markovian Dynamic BBN’s then would allow quantitative simulation of a FRAM analysis?
- Why not?