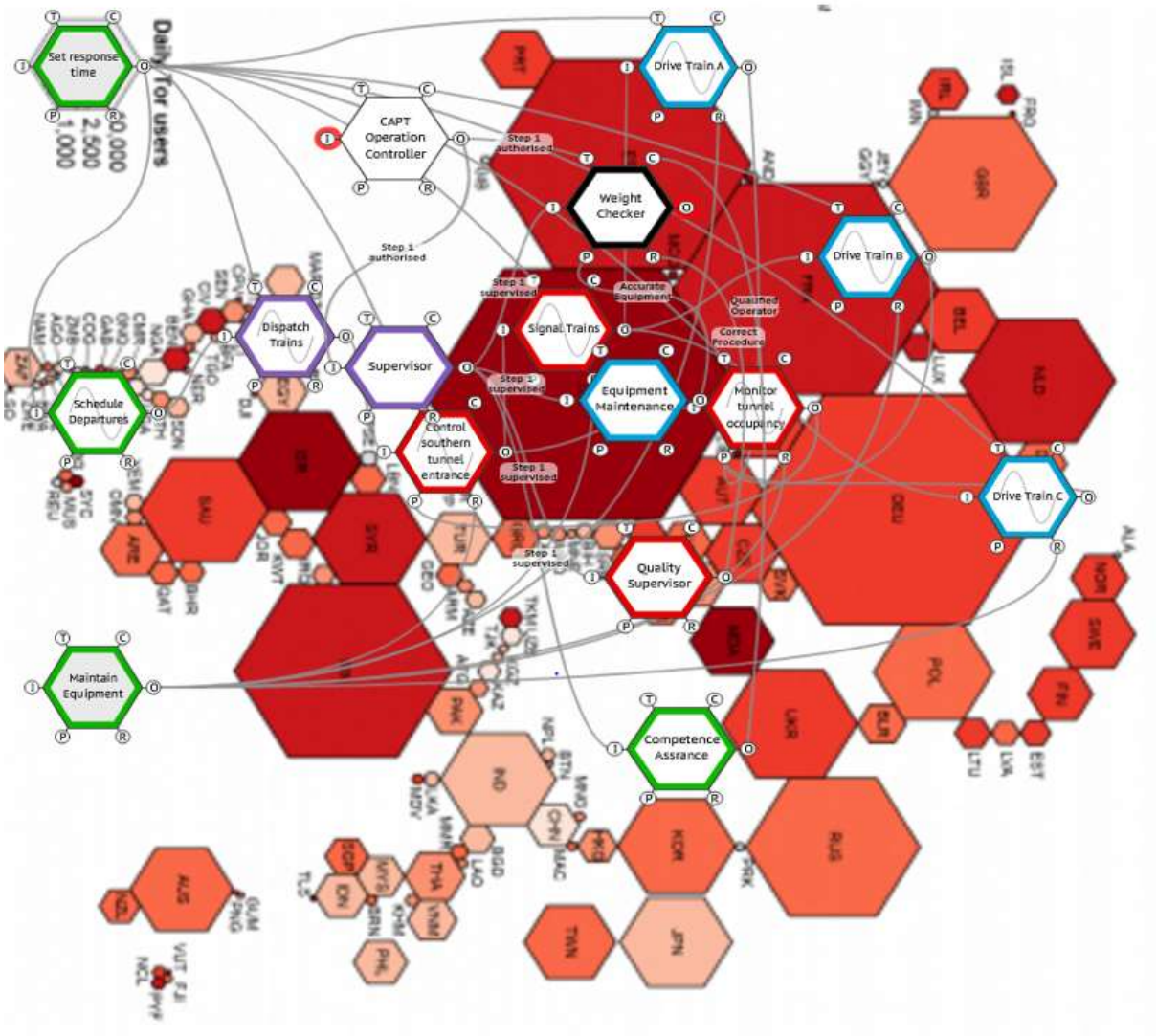
5/26/2017



FRAMsynt

FRAM Model Visualiser

Where next?

FRAM MODEL VISUALISER – WHERE NEXT?

Rees Hill, Riccardo Patriarca and David Slater

What can we do to enhance FMV and provide more tools for FRAM Users?

PROPOSAL

The basic FMV software has been responsible for (the single most important development in) an explosion in the use of the FRAM.

It is suggested we can build on this and add a sequence of "extensions" which researchers and current Users are interested in (from papers and discussions).

> *See (e.g.) Patriarca, R., Bergström, J. Di Gravio, G. (2017). Defining the functional resonance analysis space: Combining Abstraction Hierarchy and FRAM, Reliability Engineering & System Safety. http://www.sciencedirect.com/science/article/pii/S0951832016302514.*

Rees Hill suggests that we:

"*Focus on the improvements in a way that is general and helpful to many applications, and not distracting from the original FRAM method. Here are the three main improvements that we can be making, in preferred order of priority:*

1. *The ability to group and layer functions in the Visualiser. This would allow expansion and contraction of the model to view different levels of detail. We expect other models can be imported as a group, and likewise groupings exported as their own file for use in other models.*
2. *Ability to add custom metadata to the functions in the FMV data file. This would allow the user to add any number of name/value pairs to a function, to store additional data, and could be used in other visualisations such as ordering by abstraction.*
3. *External 'data harbour'."*

IMPLEMENTATION OF THESE IDEAS

One of the first things we have to agree is how we make it available. The current FMV is an unsupported download. This will be more difficult to sustain if we have to fund resources and people for development and support of the extended suite.

One suggested approach could be to leave and support the basic FMV as a free starter for 10 but think about a website/Portal to access it as a service (see Figure 1). Then for current User Groups we could offer some test versions of extensions we agree to develop.

A COMMON USER INTERFACE

The standard (opening) page would have a normal FMV button. A User login opens up a whole range of widgets and access to layering, harbour, variability modules etc.This means the different modules can be developed and tested in parallel with the portal dictating formats, interface details etc. - i.e. an interactive common testing / communication workspace?

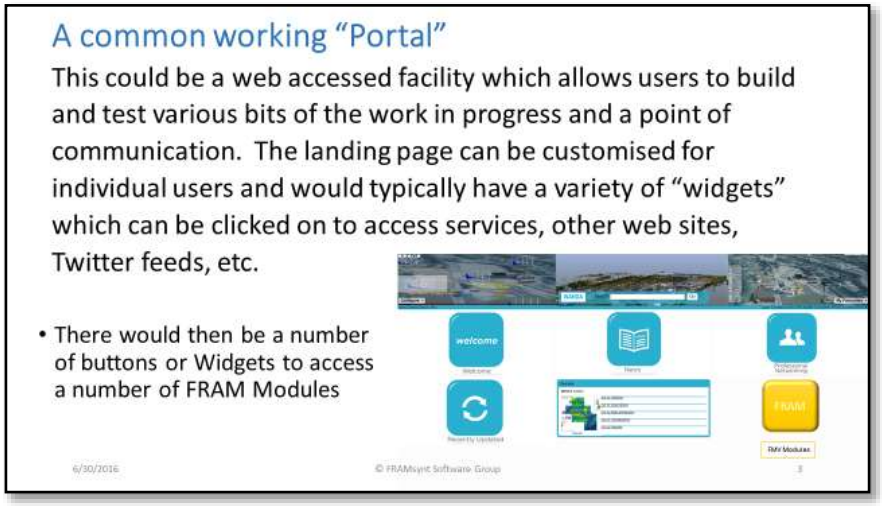


Figure 1. A sketch of a common working portal for FRAM.

So in addition to the basic FMV software, this portal could additionally offer to

PROVIDE A FACILITY TO "LINK" THE FMV SCREENS/ INSTANTIATIONS / TIME STEPS.

Current users find that they are including too many Functions and end up with a forest of Hexagons, which makes it difficult for them to see the wood from the trees. (See Figure 2):

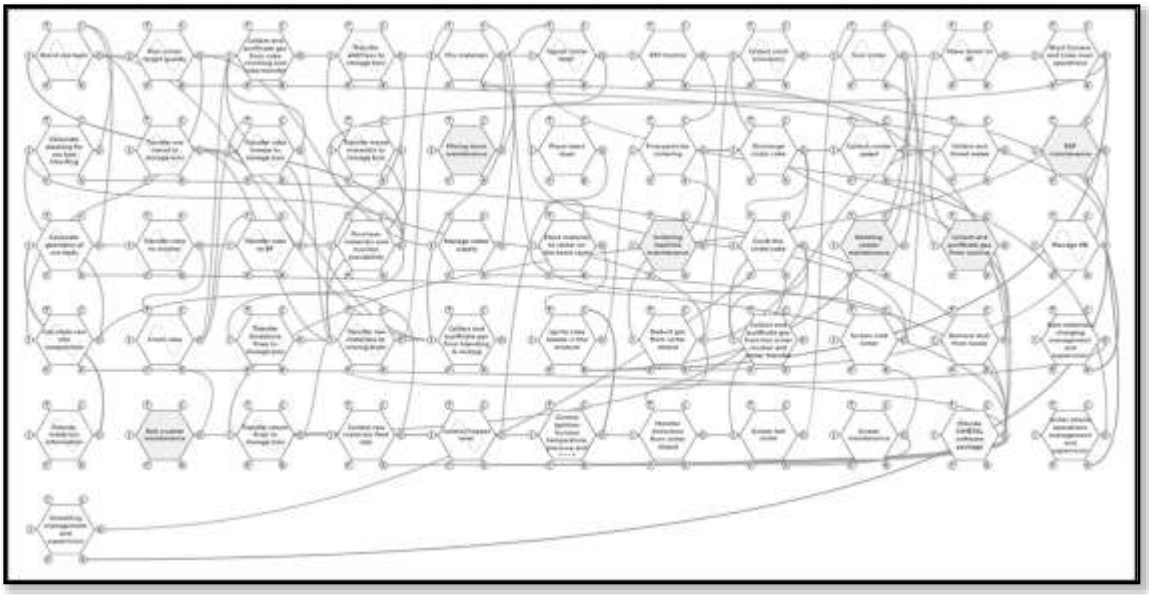


Figure 2: An example of a FRAM model for a sinter plant[1].

1 *See: Patriarca, R., Di Gravio, G., Costantino, F., Tronci, M. (2017). The Functional Resonance Analysis Method for a systemic risk based environmental auditing in a sinter plant: A semi-quantitative*

This development sounds very attractive in principle, but there are real practical problems to implementing it.

- First of these is that as Hollnagel predicted we have a potentially very large number of applications and instantiations, each of which will have their own Aspects and Functions to keep track of.
- Each of these can have any number of interactions, links and relationships with any of the other entities in the model's scope, canvas, boundaries, environment or universe.
- In fact some most probably can have links to other modelling universes.

This means we need to have a way of selecting "bite size chunks" of the modelling universe to really explore the effects of intended and non-intended variations in the critical interactions (Ref). This may just as simple as giving aspects unique identifiers as well as labels? The key to this development then is to build on the FRAM Model Visualisation software (FMV ref).The workspace utilised as a modelling canvas is shown below in Figure 2.

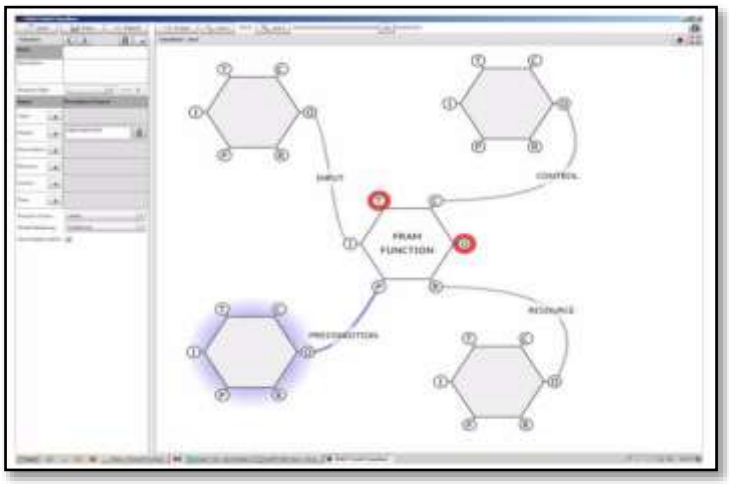


Figure 2: An example of a FRAM model for a sinter plant.

The new version of the FMV thus will have to deliver more. Every Function created will have to have a unique identifier. Over time, therefore the modeller will build up their own pool of functions. Each functions will have Aspects. As now, these are identified, labelled and linked in each of the relevant steps or instantiations and uniquely labelled by the FMV. These functions and Aspect attributes will then be stored in the data "Harbour", a relational database containing a live and historical record of functions, their identities and connections (FRAM Aspects).

approach, Environmental Impact Assessment Review, Volume 63, March 2017, Pages 72-86, ISSN 0195-9255, https://doi.org/10.1016/j.eiar.2016.12.002.
http://www.sciencedirect.com/science/article/pii/S0195925516302104

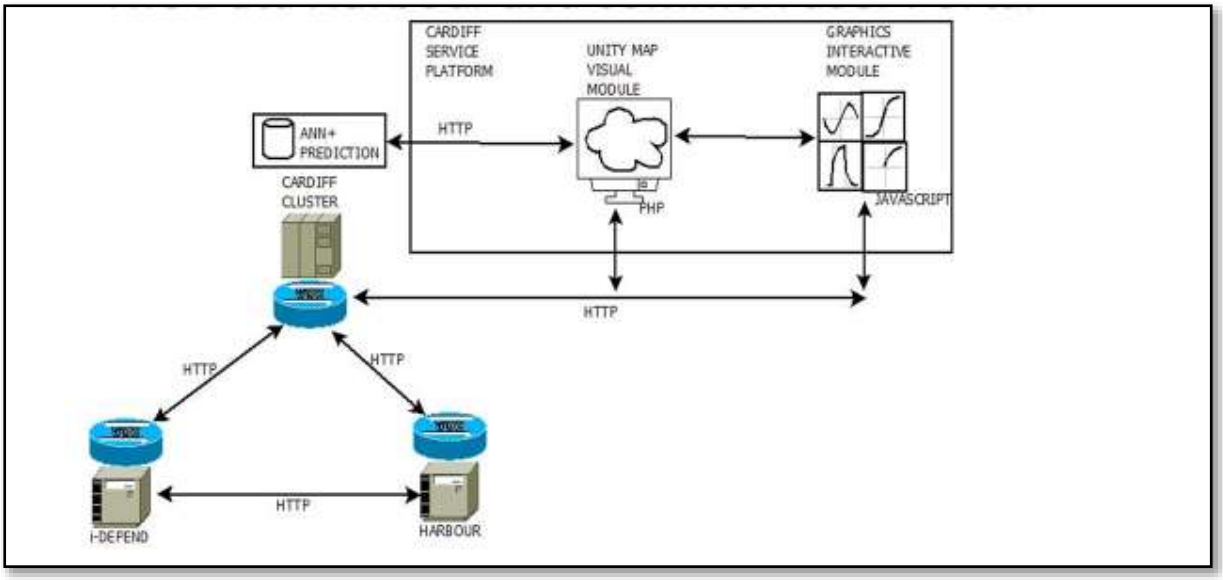


Figure 3: The DATA Harbour and common user Portal.

This will enable users to create simple instantiations and link them up, or drill down to link with "the bigger picture" (an overview of the total system), as recently suggested in a research paper published in Reliability Engineering & System Safety [2]:

- common to a whole series of different instantiations at different levels of abstraction or
- owned by different groups in the same organisation
- different levels in an abstraction hierarchy
- different "agents"
- Different time steps in FMV's record feature

So a user can select a specific step of a process and create, link and import those critical functions and aspects as needed on its own workspace.

The identifier can also be labelled as to the time of that particular instantiation, so that temporal developments can be modelled in the same emergent way.

Then, every new workspace opened to build a new model, will have access to the data harbour. This will allow new functions and aspects to be labelled and stored, but also a dialogue can check whether these are related to pre-existing data, or linked to existing functions which can be imported.

This means that links can be made automatically with any of the existing functions in the modeller's personal universe.

This means that there is an ability to discover emergent states where unexpected links can cause unpredicted behaviours.

---

2 *See Patriarca, R., Bergström, J. Di Gravio, G. (2017). Defining the functional resonance analysis space: Combining Abstraction Hierarchy and FRAM, Reliability Engineering & System Safety, Volume 165, September 2017, Pages 34-46, ISSN 0951-8320, https://doi.org/10.1016/j.ress.2017.03.032.* http://www.sciencedirect.com/science/article/pii/S0951832016302514

By breaking down the modelling universe into a series of bite size, but fully linked canvases or FMV work spaces, the analyst can handle more comfortably that particular local effects of aspect interactions and their potential variations, but also be automatically prompted by interactions or resonances from other off sheet functions.

Finally there is the possibility of linking previous analyses or other modeller's databases to provide the same spontaneous interaction flagging.

- *A HIERARCHY OF FRAM FUNCTIONS*

There have been a number of suggestions that functions can themselves be composed of a number of subfunctions which together act as if they are a single higher level function.

These sub functions in turn can be composed of yet more functions. So that it is possible to explore in more detail exactly which interactions are involved.

Patriarca et al. (2017) proposed to utilise Rasmussen's Hierarchies to help to organise and visualise these functions into these sub groupings at different levels – a hierarchy of nested functions, which can be developed further, or just as easily subsumed in higher level generalised functions.

- The first, are the different levels of influences from foreground to background or, immediate to external.
- Secondly, in each of these levels, there are individual and identifiable entities or Agents operating in combination (Rasmussen's Actor (Authority) Hierarchy)
- Thirdly, there is the level of detail at which these functions are defined (it can be based on Rasmussen's Abstraction Hierarchy)
- And finally there is the inexorable Temporal Hierarchy in following emergent behaviours as a sequence of instantiations in particular event sequences (the March of Time?)

The Figure 10 below from Patriarca et al. (2017) illustrates the way sets of nested functions are related. But because these links can happen at any level of abstraction and with any other influencing entity /agent, the unique identifier needs to include the relationship attributes shown in Figure 5, adapted from Patriarca et al. (2017).

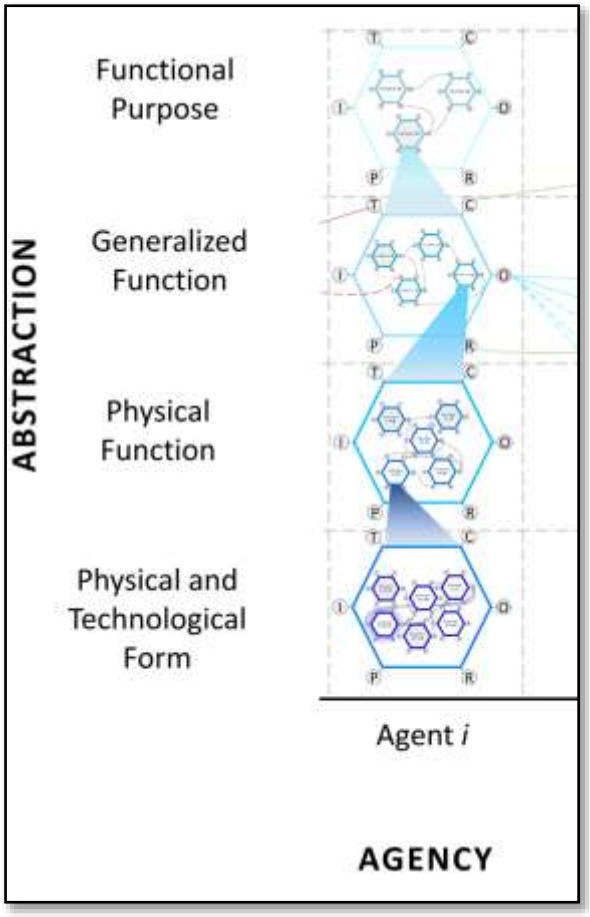


Figure 4. An example of fractal FRAM in an Abstraction/Agency framework, adapted from Patriarca et al. (2017).

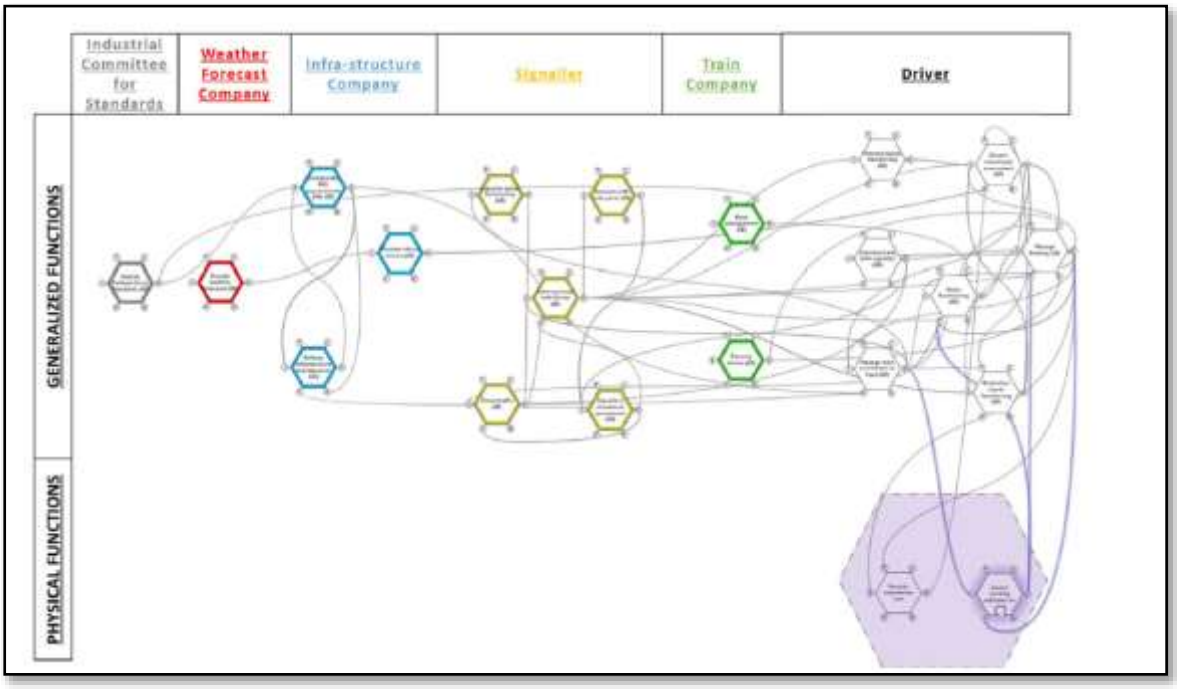


Figure 5. The Abstraction/Agency framework for 6 agents, showing the interactions among functions at two different abstraction levels, adapted from Patriarca et al. (2017).

ADD A FACILITY TO INCLUDE AN ESTIMATE OF VARIABILITY

Initial (obvious?) suggestions:

- Resurrect the "automatic" assignment of variabilities on type of function H, T, O as before and make it an add-on feature?
- Add Riccardo Patriarca's VBA code to develop a user-friendly model builder and data entry (based in Excel), which includes Monte Carlo simulation as an option [3].

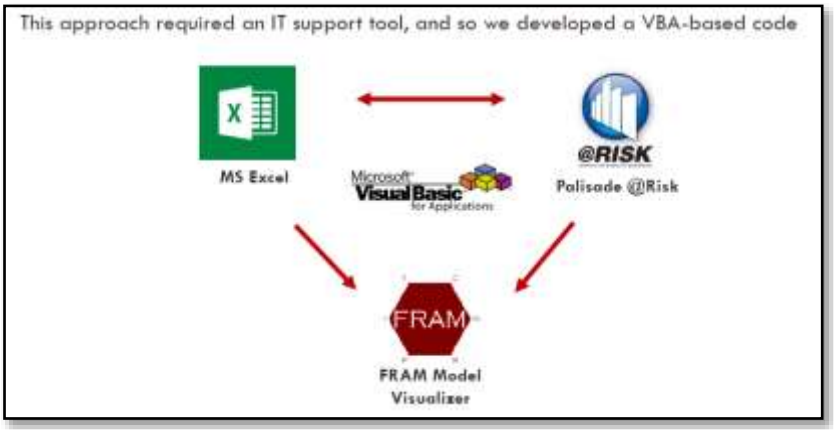


Figure 6. Logic sketch of Patriarca's VBA code for FRAM model builder (in Excel) and Monte Carlo simulation (@Risk), allowing a graphical representation (FMV).

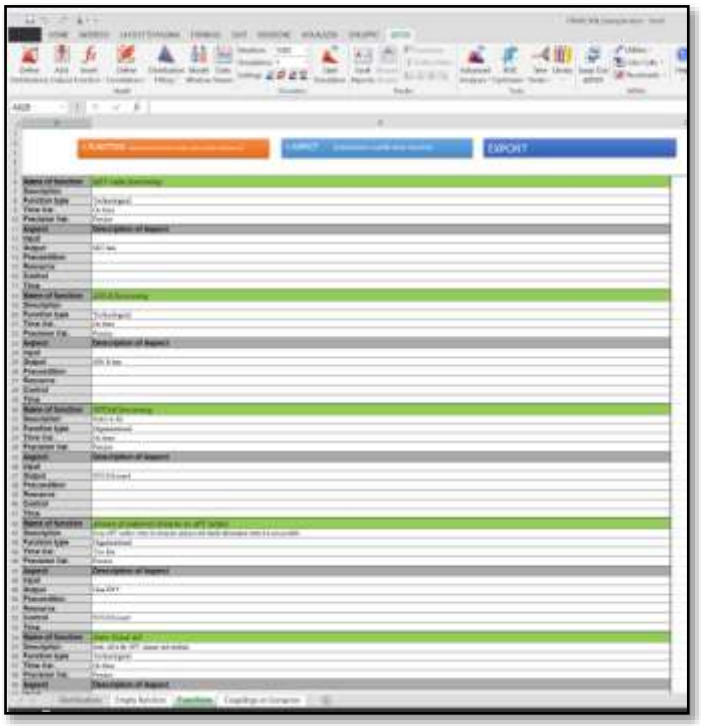


Figure 7. Screenshot of Excel data entry for developing the FRAM model for FMV (from Patriarca's code).

---

3 *Patriarca, R., Di Gravio, G., Costantino, F. (2017). A Monte Carlo evolution of the Functional Resonance Analysis Method (FRAM) to assess performance variability in complex systems,* **Safety Science***, Volume 91, January 2017, Pages 49-60, ISSN 0925-7535, https://doi.org/10.1016/j.ssci.2016.07.016.* http://www.sciencedirect.com/science/article/pii/S0925753516301576

And from our "Quantification of FRAMs" paper[4]-

"If we set the limits of variability that a given function can tolerate, (i.e. **upper** and **lower** limits), then the probability of a function operating inside these limits can be predicted, given a mean and an estimate of the "deviation"(or shape) of the distribution. The bigger the deviation the more likely it is that the probability will be outside the limits allowed.

To quantify the probability of a function's output being "off limits" we need to specify:

- The "expected" value of that parameter. **(the mean μ)**
- An estimate of its "spread" – the **(standard deviation σ)**
- The upper and lower "tolerance" bounds **(Z_{max}. And Z_{min})**

The probability estimates can then be calculated as the areas under the probability distribution function. (e.g. using the Excel function NORMDIST)".

These probabilities can be displayed on the FMV hexagons directly, or we can utilise the iDepend link as before to translate this into an estimate that the output will be successful?

Another option is then to use the BBN quantifications, which could be extended to give:

- MonteCarlo runs to "discover" functional resonances
- Dynamic BBN's as FRAM emergences
- Simulation of what if's

Quantification by Structuring the "Function" as a BBN

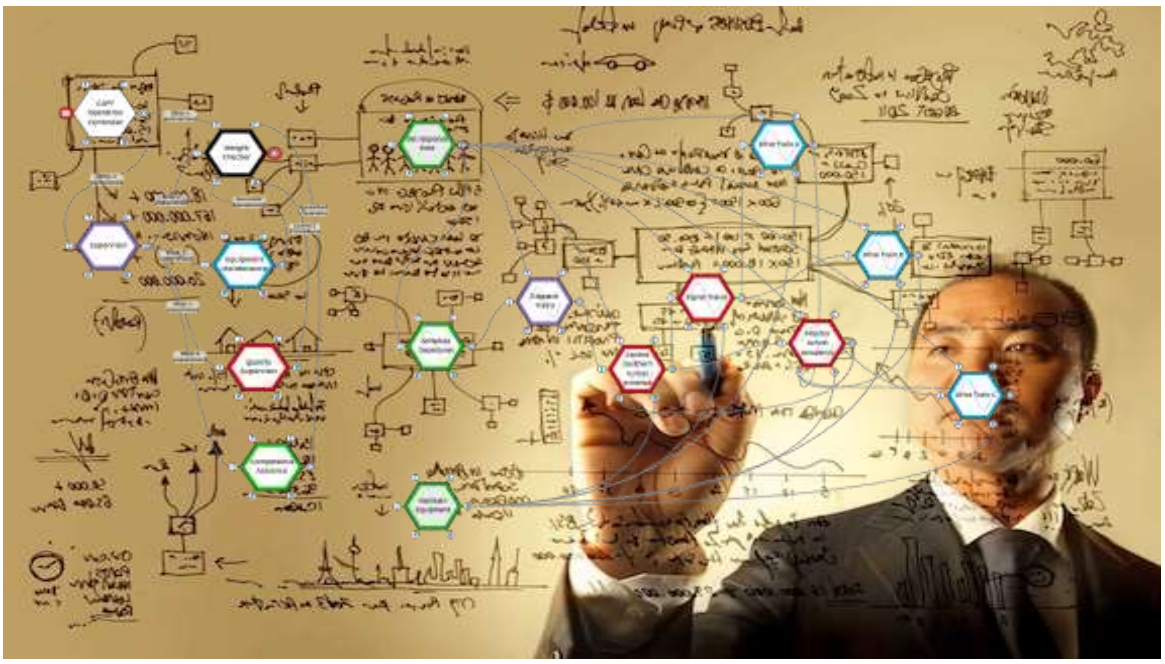
The Hollnagel FRAM function is essentially a BBN node (the function) influenced by external links to other factors (Aspects). Its successful functioning thus depends on the probabilities of those influences being of the right type, quantity and timing. Therefore, each function is in effect a BBN or Dependency model (Open group[5]).In a FRAM instantiation, where a number of these functions link together, they are then an emerging sequence of linked dependency modules, or each instantiation is effectively a separate Bayesian Belief Net (footnote 4).

So we can show this by building a "dependency model" of how the OUTPUT of our FRAM function – is dependent on other things happening. (It depends on how the function varies and that depends on the Outputs from upstream functions, among other things.)

Using the dependency tool then, we can assign probabilities to the "status" of the outputs of these background functions, which will then be processed by the dependency model of each of the other functions of which they are "Aspects". The dependency model is in effect "structuring" the internal workings of the function, quantitatively, rather than heuristically (common sense is always needed of course!). The structured function could then display (in the FMV), its chances of its own output being "on spec". This is the "expected" performance of the function based on the probability of the correct status of the Outputs of upstream functions

This can then be displayed as red/ green bars on the individual function as in footnote 4.

4 Slater D.H. *https://www.researchgate.net/publication/305886439_QUANTITATIVE_VARIABILITY_IN_FRAM*
5 Slater D.H. et al. *https://www.researchgate.net/publication/305884742_Open_Group_Standard_Dependency_Modeling_O-DM*



Minority Report Technology for Improving System Performance?