The Functional Resonance Analysis Method

HOW TO DO A FRAM ANALYSIS -

This is intended as a short introduction to the ideas and steps in FRAM modelling. More extensive guides are available in the literature[1]. It basically consists of a sequence of logical steps.

1. What are we modelling?

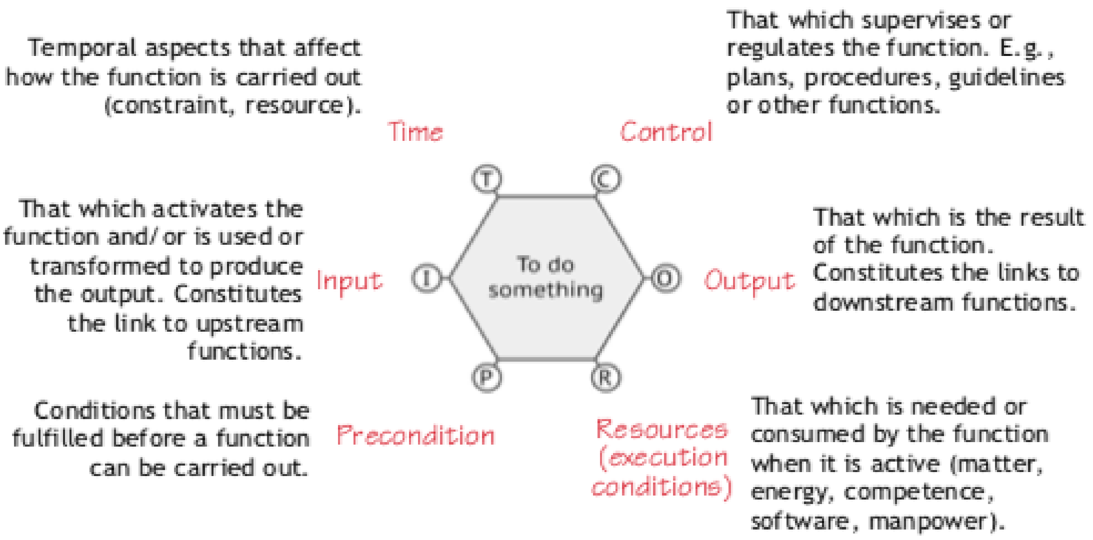
Here we build a data and information base about the process involved. This can consist of a mixture of desk work and interviews with the actual people who know / operate these systems. For example.

- What tasks are involved?
- What functions are needed to perform those tasks?
- Are there critical needs for the process to complete successfully?
- Do we have the timings, sequencing, resources, preconditions, mandatory controls, external constraints, operational limitations, etc., needed to enable the system to perform as designed?

2. How do these functions combine to define the system?

The aim here is to ***build*** a FRAM model of the system involved in the process/activity. So, the functions identified, are set out in an interacting, interdependent picture, as a "cloud" visualisation, normally achieved using the software tool, the FRAM Model Visualiser, (FMV)[2], so that we can identify and specify: -

- For each function, we then need to define the critical "FRAM aspects", or execution conditions (Figure below), needed to allow the function to operate and fulfil its task.



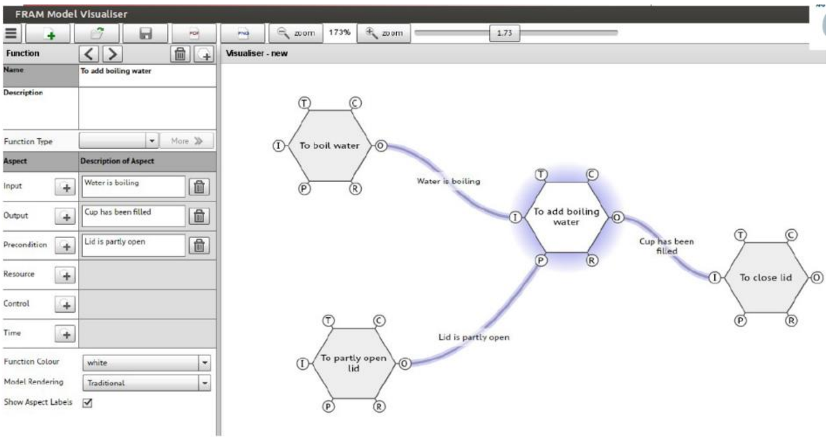
- These "Aspects" will thus consist of an **Input (s)** needed to trigger the function, plus any other requirements, such as **Preconditions**, **Resources**, **Control** states or signals and any **Timing** constraints.

---

[1]https://www.researchgate.net/publication/325825191_THE_FUNCTIONAL_RESONANCE_ANALYSIS_METHOD_AND_MANUAL_VERSION_2

[2] https://functionalresonance.com/the%20fram%20model%20visualiser/

- These interacting links between the functions can only arise as **outputs** of other functions and as such, they will build up as the analyst steps through all the functions in the system?
- Functions with O**utputs** only, are, by definition, **Background** functions, which set the boundaries of the system. They are also the sink or drain (exit) functions, which only have an **Input**. All the rest are therefore **Foreground** functions.
- Background functions can also be used as placeholders for later extensions, as FRAM Models can be developed more widely, or merged with other activities.
- Similarly, foreground functions can be expanded like Russian dolls (Functions within Functions), to probe finer details. This is also facilitated in the FMV software.

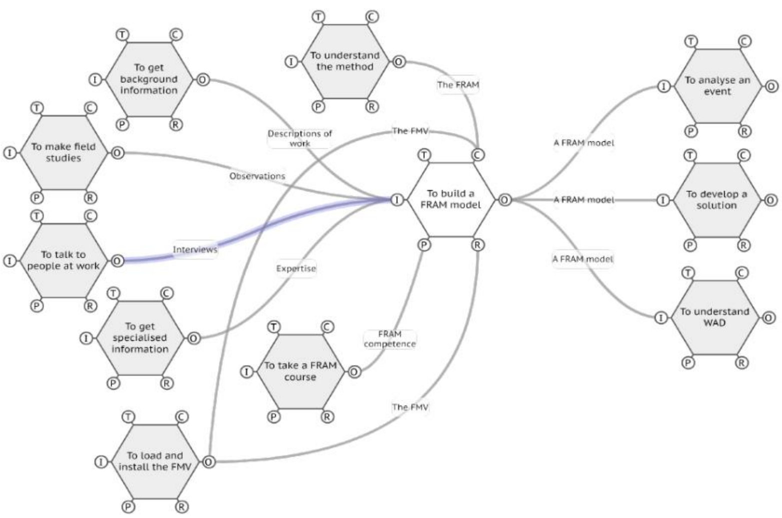


- In a sequence of tasks in a particular instantiation of the model, (a specific instance or "snapshot" of the system's operations), those functions which have to deliver before others can begin, are called **Upstream** functions.
- Then subsequent functions are obviously **Downstream** functions.
- In any particular **instantiation** of the model, outputs from downstream functions cannot link to aspects of upstream functions. For example, they can't initiate upstream functions if they themselves have not been initiated.
- Similarly **closed loops** are not allowed, where a function's **outputs** are linked back as aspects of the same function.
- In contrast, aspects not linked to specific **outputs** of other functions are known as "**Orphans**", which need removing before a model can be validated.

3. Testing the System Model's validity

Normally this "As Imagined" system model is checked in two ways: -

- **Peer Review** – Most FRAM analysts will, at this stage, look for confirmation of the model's accuracy and check the visualisation as a run through session with the teams involved.

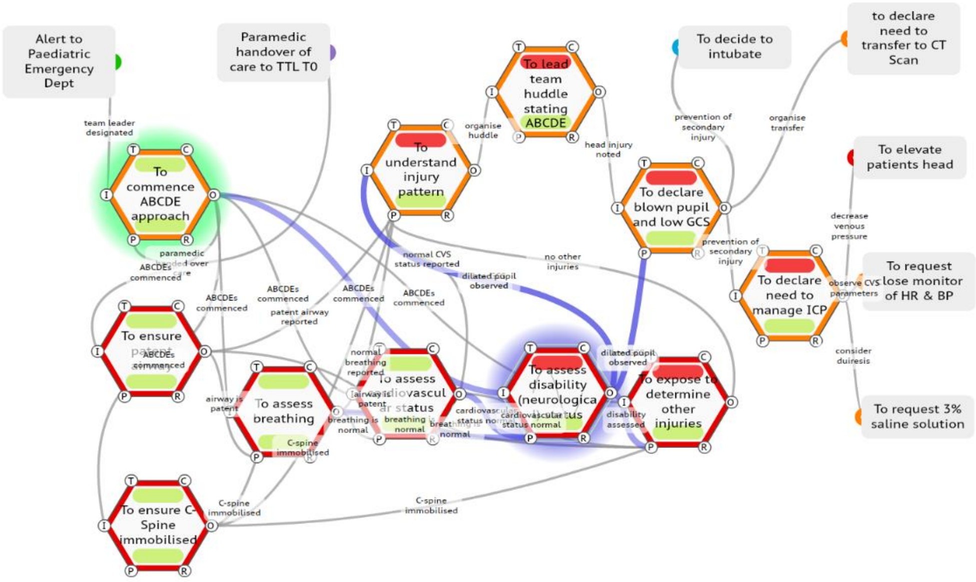


- **Formal Validation** – Having agreed that the system model now accurately reflects the process "As Done", the FRAM model can now be checked and adjusted for consistency and completeness, using another software facility, the FRAM Model Interpreter, (FMI)[3].

4. What insights can we obtain from observed and expected variabilities in these systemic interactions?

The validated FRAM model can now be used to analyse the system's behaviour in two ways: -

- **A. From observations on actual or simulated work situations**, where the FRAM visualisation now allows a recording and analysis of the effects of actual variabilities in the status of the necessary links, in real environments.
- In working through the effects of these variabilities, the analyst and the work teams can then elicit and identify elements and situations where there are opportunities to correct and compensate for problems and enhance/ enable more successful and resilient operations.



- **B. By a formal process**, much like a HAZOP, where systematically, starting with the entry function, the question is posed, what if the output of this function is normally, in terms of timing, on time, what happens to the system if it's too early, too late or not at all,
- Similarly, in terms of precision (Scale, quantity, etc,) what happens if the output of that function is instead of being precise is variable within acceptable limits, or imprecise.
- Again, the FMV software tool allows the analyst to highlight and identify the consequential and often unexpected emergent downstream effects of these variabilities on other functions (the resonances implied by the FRAM acronym), and hence the system's expected performance. This has been termed a **SWI-FRAM** (– "so what if?") analysis.

5. Applying the insights gained

As we now have a validated and consensual "model" of the system actually employed in practice, the analyst, and / or the team, can start to look more closely at the system "As Designed". It thus allows them to incorporate the lessons learned from what affects the system, both negatively (SAFETY I) and positively (SAFETY II). This is the opportunity to modify the system, not only to eliminate the constraints and shortcomings, but also to incorporate those adaptations, that the work teams have adopted, as ways of working around, or like lean teams and quality circles, minimising "defects", optimising operability and thus improving productivity and safety.

[3] https://functionalresonance.com/the-fram-model-interpreter.html