FRAM: First Steps

Erik Hollnagel
hollnagel.erik@gmail.com
www.safetysynthesis.com

Building a FRAM model

1. Identify and describe the essential functions in the activity/event. Characterise each function by means of the basic aspects using the FMV. The FMV will show whether the model is consistent.

2. Characterise the typical / potential variability of 'foreground' functions using the FMV. Consider both everyday and excessive cases of variability.

3. Go through a number of instantiations of the model to look for cases of functional resonance based on potential / actual dependencies (couplings) among functions.

4. Propose ways to monitor and manage performance variability (indicators, barriers, design / modification, etc.)

Identifying Functions: General

PURPOSE: A FRAM analysis aims to identify how the system functions (or should function) for everything to succeed (i.e., everyday performance), and to understand the variability which alone or in combination may prevent that from happening.

MODEL: A FRAM model describes a system's functions and the potential couplings among functions. The model does not describe or depict an actual sequence of events, such as an accident or a future scenario.

INSTANTIATION: A concrete scenario is the result of an instantiation of the model. The instantiation is a "map" of how functions are coupled, or may become coupled, under given – favourable or unfavourable - conditions.

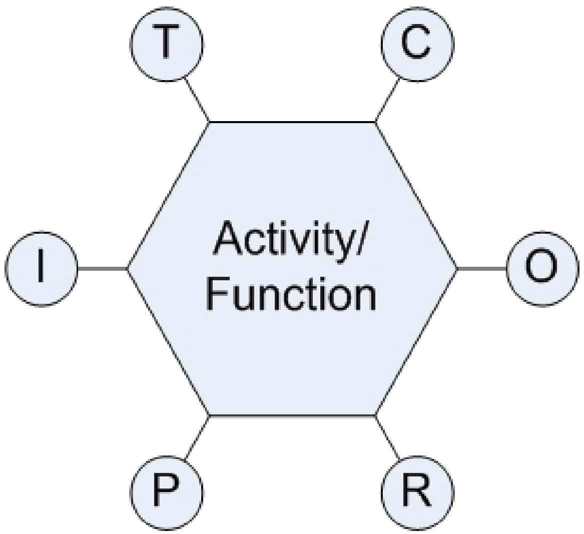
Describing a FRAM function

SafetySynthesis

Temporal aspects that affect how the function is carried out (constraint, resource).

That which supervises or regulates the function. E.g., plans, procedures, guidelines or other functions.

That which activates the function and/or is used or transformed to produce the output. Constitutes the link to upstream functions.



That which is the result of the function. Constitutes the links to downstream functions.

System conditions that must be fulfilled before a function can be carried out.

That which is needed or consumed by the function when it is active (matter, energy, competence, software, manpower).

Describing the functions

Begin to describe the functions using the FMV. Build the model using a breadth-before-depth principle.

Depth-before-breadth

The usual way to carry out an analysis is depth-before breadth.
In accident investigations this is illustrated by the search for the root cause.
In decision making this is illustrated by exploring all the consequences of each alternative.'In depth-before-breadth each possible path is explored exhaustively before other paths are considered.

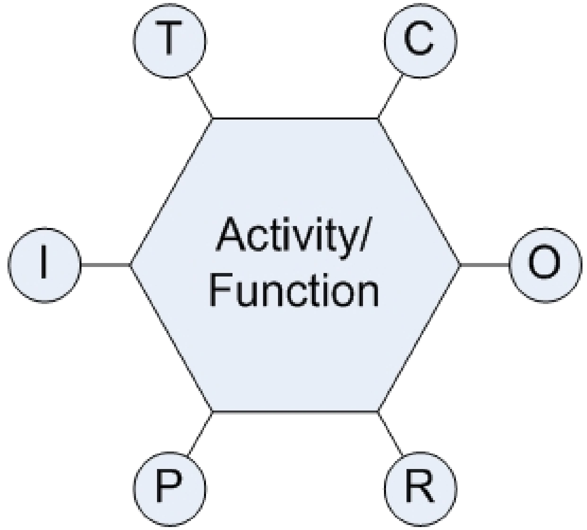
Breadth-before-depth

When building a FRAM model, try to describe the activity as a whole, before going into detail with specific functions.
Try to understand the "big picture" – the activity in its context – before getting lost in details.

FRAM: Characterisation of input

That which is taken in, or operated on by a function, and used as basis for the output. The input can be matter, energy, or information.
The 'signal' that activates or starts a function (command, information, clearance, instruction).
A change of state of the environment, e.g., a piece of paper in an in-tray, an order to a chef, a new patient in an emergency room, no more cars waiting on the quay, etc.

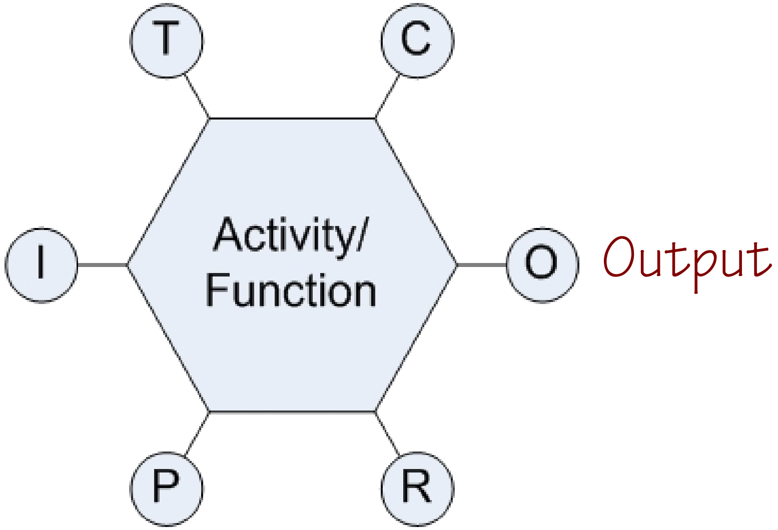
Input



All inputs must have an origin or source, which means that an input to one function must be the output from another function.

Designated foreground functions must have defined inputs, while designated background functions need not have.
If the function does not have en Input, it will never be carried out!

FRAM: Characterisation of output



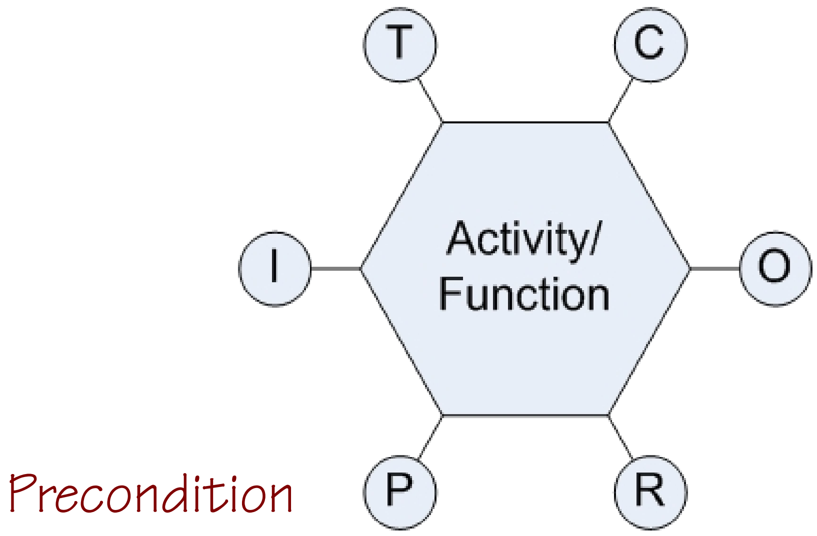
The result of the function, or what is produced or manufactured by the function. The output can represent matter, energy, or information, for instance, a decision, a command issued, or the result of some kind of deliberation.
The output represents a change of state – of the system or of one or more output parameters.

The output makes clear how variability can propagate through a system. If a function varies then it is also likely to vary in some way. Since the output from an upstream function will be the input to a downstream function, variability of the output may lead to variability in the (downstream) function, and so on.

All functions, except a designated drain (or sink) must have an Output.

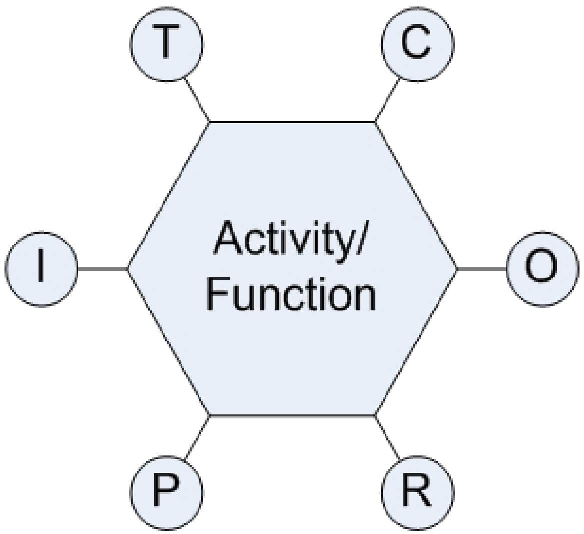
FRAM: Characterisation of precondition

The system conditions that must be satisfied before a function is carried out. The preconditions must be the output from one or more other functions. Preconditions provide demonstrate how functions can be coupled.



A precondition is a state that must be true before a function is carried out, but it is not itself the signal that starts the function. An input, on the other hand, can start the activation of a function.

FRAM: Characterisation of resource



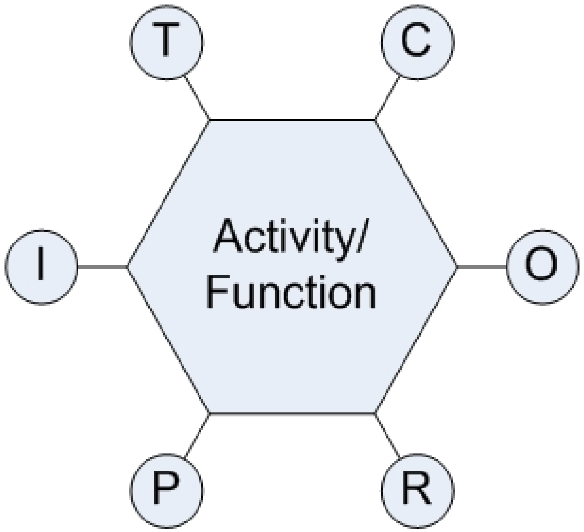
Resources
(execution conditions)

Something that is needed when a function is carried out - matter, energy, information, competence, software, tools, manpower, etc. A resource is consumed while the function is carried out. Execution conditions must be present for the function to be carried out, but are not diminished because of that, e.g., data, competence, or skills.

Resources must be produced by one or more functions. Execution conditions are assumed to be stable for the duration of the event (output from 'dummy' functions).

FRAM: Characterisation of control

Control



Something that supervises or regulates the function to ensure that it produces the desired output - plans, schedules, procedures, guidelines or instructions, etc. Control can also be the conditions that constrain the sequencing of sub-steps. Explicit control is provided by another function (procedures, instructions). Social control is the expectations of others (company, management, colleagues) or of the agent itself.

Control must be the output from an upstream function. If it is stable, then it is reasonable to assign it to a background function. If it is active and adaptive, then it should be assigned to a foreground function.

FRAM: Characterisation of time

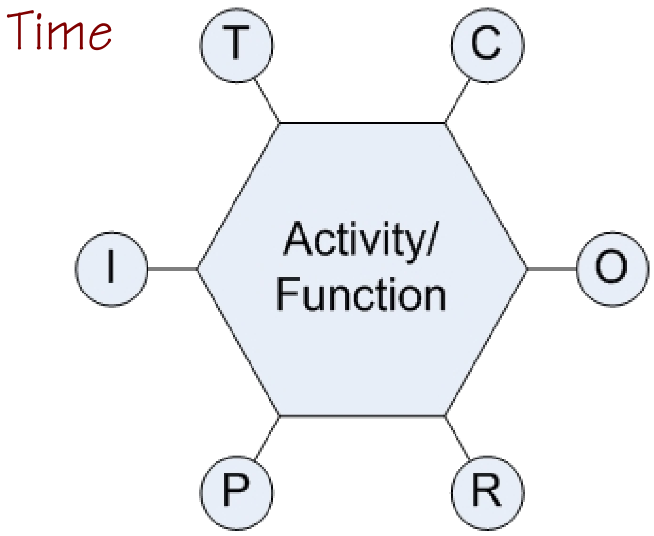
This aspect represents the various ways in which time can affect performance.

Time as control: relative or absolute sequencing conditions.

- Earliest Starting Time (EST),
- Latest Starting Time (LST),
- Earliest Finishing Time (EFT), and
- Latest Finishing Time (LFT).

Time as a resource: time available, deadline, etc.

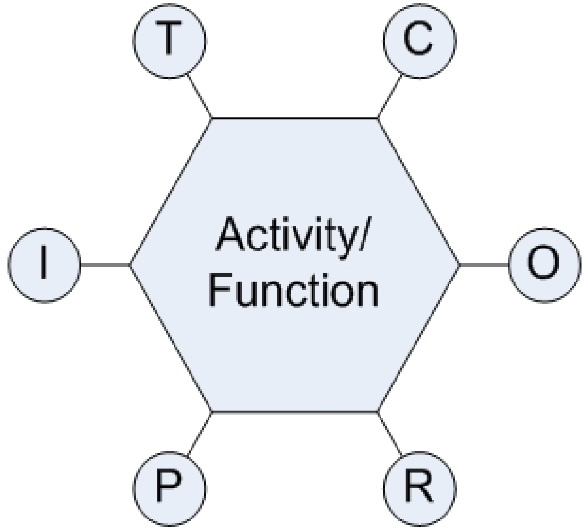
Time as a precondition: clock-time or relative to other functions.



More about the aspects

There is no requirement to define all six aspects for every function.
In practice, you should only describe aspects if it seems to be relevant or necessary, based on knowledge about the activity.
For each type of aspect there can be more than one entry.
A function may, for instance have a single Output that used by several downstream functions.
A function may also have two – or more Outputs.

A function may also have two – or more – upstream couplings to its Input, Precondition, Resource, Control, or Time aspects.



Data collection

SafetySynthesis

"Work-as-imagined"	"Work-as-done"

"Work-as-imagined"

The various sources, descriptions or specifications, for the type of work being studied.

- Policies, strategies, guidelines, instructions, checklists
- Documented experiences (cases, events, statistics)
- Workplace layout and organisation, equipment, facilities.

"Work-as-done"

The various ways in which information about actual work practices can be obtained.

- Interviews – open and appreciative questions
- Checklists to guide interviewers
- Information about attitudes, habits, assumptions, tacit knowledge
- Typical adjustments (variability)
- Dependencies among functions, conditions that affect performance

Identifying Functions: Details

Where to begin	A FRAM analysis can in principle begin with any function. The analysis will show the need for other functions to be included, i.e., functions that are coupled or linked through various relations. FRAM defines six types of relations.
Level of description	There is no single, correct level of description. A FRAM model will typically comprise functions described on different levels.
Foreground background	Functions are pragmatically labelled as being either foreground or background functions.
Level of detail	If there can be significant variability in a foreground function, then it is possible to go deeper into the analysis of that function, and possibly break it down into subfunctions.
System boundary (stop rule)	The analysis may go beyond the boundaries of the system as initially defined. If some background function can vary and thereby affect foreground functions "inside" the system, then it should be considered a foreground function.

Foreground and background (functions)

FRAM uses a distinction between foreground and background functions, which may all affect performance variability.
Foreground functions are directly associated with the activity being modelled and may vary significantly during a scenario.
Background functions refer to common conditions that may vary more slowly.

The distinction between foreground and background functions is relative rather than absolute.
A 'background' function may be analysed further, and thereby becomes a 'foreground' function.

Both sets of functions should be calibrated as far as possible using information extracted from accident databases.

Upstream and downstream

Function A is upstream, relative to function B

Function B is downstream, relative to function A