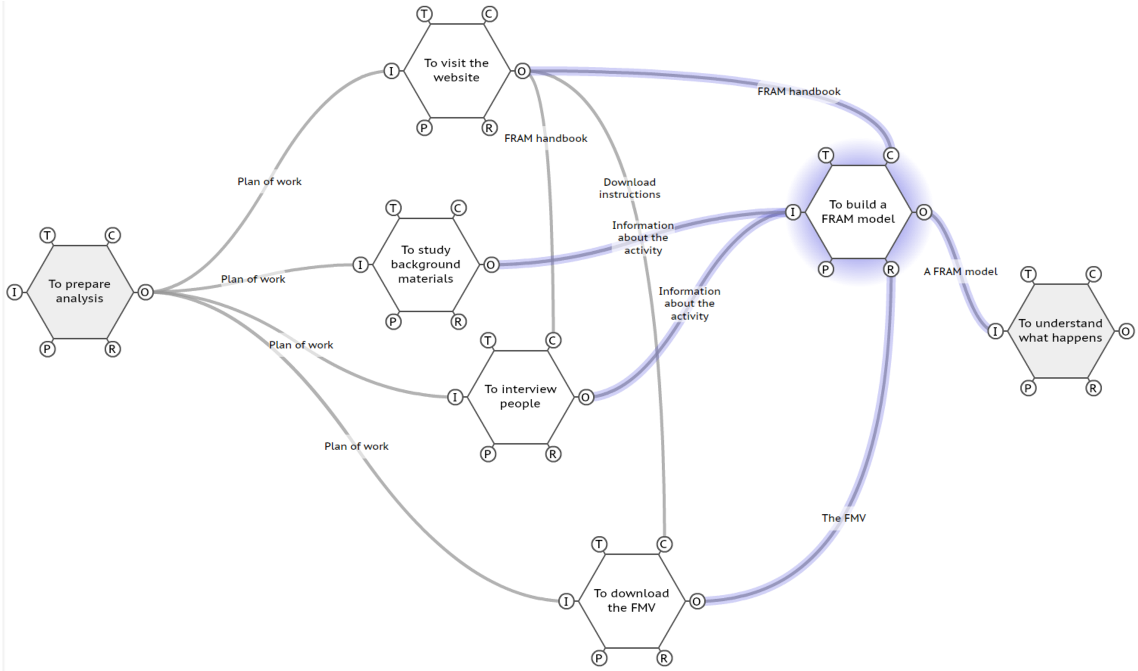
FRAM

The Functional Resonance Analysis Method

A brief Guide on how to use the FRAM



A brief Guide on how to use the FRAM

Advice to Readers

This handbook is written as a practical guidance for the use of the FRAM and the FMV. It is **not** intended to be a complete description of the method itself, and it is inadvisable to use the FRAM without being familiar with the theoretical background. The FRAM represents a way of thinking about how systems function that differs significantly from the traditional approaches. The different understanding requires both familiarity with the underlying concepts and theories, and practical experience with the method. This can be achieved for instance by attending an authorised FRAM course, by reading the book about the FRAM or preferably both.

The handbook has been written to be part of a FRAM course, hence to be used with the support of an instructor. It is not intended to be and should definitely not be used as a simple cook book or step-by-step instruction manual. It is, however, probably inevitable that many people will try to use the method on their own, not least because courses are far from regular. This is not to be discouraged, as long as the above advice is kept in mind. Like any instruction manual, this brief guide is incomplete and should be complemented by an experienced user or instructor. If you do not personally know of someone who can provide such experience, it is highly recommended to get in touch with the FRAMily on LinkedIn.

Before reading, it is recommended to download and install the FMV. The FMV is in the public domain, hence free of charge. Instructions on how to do that can be found here; http://functionalresonance.com/the%20fram%20model%20visualiser.html.

About the Functional Resonance Analysis Method (FRAM)

The purpose of the FRAM is to analyse how something *has been* done, how something *is* done, or how something *could be done* in order to produce a representation of it in a reliable and systematic manner, using a well-defined format. This resulting representation is effectively a model of the activity because it captures the *essential features* of how something is done. In the case of the FRAM, the *essential features* are the *functions* that are necessary and sufficient to account for the activity together with the way in which the functions are coupled or mutually dependent.

Although the FRAM was developed in the context of the common understanding of safety around the turn of the century – what now is referred to as Safety-I – it is not just a safety or accident analysis method. The FRAM is based on four principles of which the first is the principle of equivalence. (The four principles are described in detail in the following.) According to the Safety-II perspective and the principle of equivalence, activities that have acceptable outcomes and activities that have unacceptable outcomes happen in much the same way. When the motivation for an analysis is that something has gone wrong, as in an accident or an incident, the principle of equivalence argues that we should try to understand how an activity normally goes well before trying to understand why it failed or did not go well in a particular situation. Whereas the focus of accident analysis methods such as HFACS, TRIPOD, the Bow-tie, and STAMP is on *why* something has *gone wrong*, the focus of the FRAM is on *how* something is *done*.

The FRAM is not especially developed for or limited to accident analysis and safety management and can equally well be used for task analysis, system design, etc. Examples of how the FRAM has been used in practice can be found at

http://functionalresonance.com/publications/index.html.

The Four Principles of the FRAM

All methods that are used to analyse something refer to an embedded model that describes and explains how the system being studied works or functions. (More formally, a distinction can be made between the *object system* that is being studied or analysed and the *model system* – or system model – that is the result of the analysis. The embedded model, clumsy as the term may be, provides the rationale for the method and is usually taken for granted, hence forgotten. The embedded model defines or describes a set of parts and how they are related while the associated method provides a way to interpret what is being studied in terms of those parts and relations. The relations typically invoke the principle of causality (causes leading to effects or effects being preceded by causes) or some kind of hierarchical or temporal relation. A method can in this way be seen as implying or embedding a generic model of the phenomenon, or as being a model-*cum*-method. The embedded model thus guides the analysis, and in practice often imposes an *a priori* interpretative structure on the system being studied, hence on the model that is the result of the analysis. The value of the results of an analysis therefore depends on the appropriateness of the model that is behind the method.

Where commonly used methods for the study of socio-technical systems try to describe relations derived from an embedded model, and therefore represent model-*cum*-method approaches, the purpose of the FRAM is to do the opposite. The FRAM proposes that everyday events and activities can be described in terms of the functions involved without predefining specific relations, levels, or structures. Instead the FRAM assumes that the behaviour of functions, hence the outcomes of an activity or process, can be understood in terms of the four basic principles described in the following. The FRAM thus does not have an embedded model, not even a non-linear one, and makes no assumptions about how the system under investigation is structured or organised, nor about possible causes and cause-effect relations. It describes the functions using six aspects, but the method does not exclude that additional aspects might be useful, nor that a completely different set of aspects might be possible. The FRAM can be described as a method that is used to produce a model, instead of a method that is derived from a model, and can therefore be seen as representing a method-*sine*-model approach. Since the main purpose of the FRAM is to build a model of how a set of activities are carried out (and to create instantiations from that model) the method must stand on its own hence be more detailed than most other methods.

The FRAM focuses on describing what happens in terms of functions. These are derived from what it takes to achieve an aim or perform an activity, hence from a description of Work-as-Done rather than Work-as-Imagined. The functions are not defined *a priori* nor are they necessarily ordered in a predefined way such as a hierarchical (superordinate-subordinate) or temporal (preceding-following) relation. Instead they are described on their own and any relations between them are based on experience from working with the system, from generic practice, and/or from design assumptions, thus on empirically established functional dependencies rather than implications of an embedded model.

One advantage of this is that a description of functions using the FRAM is scale invariant. This basically means that the processes that give rise to phenomena on a small or a large scale can be described in the same way. If, for instance, we look at things from the point of view of failure, then there would be no difference between high level failures (tactical) or low level failures (operational) – not least because there are no levels in a FRAM model. The obvious advantage of scale invariance is the simplicity of the method and the parsimony of explanations. The latter is perhaps most important, because it eliminates the need of large sets (or even taxonomies) of categories, that both can be cumbersome to use and that artificially constrain the depth and breadth of an analysis.

First Principle: The Equivalence of Successes and Failures

Explanations of how systems function or how everyday events develop typically rely either on decomposing the systems into meaningful physical parts or components, such as people and machines, or on decomposing the event into individual actions, process stages, or steps in a sequence. Outcomes are explained in terms of linear cause-effect relations among the parts or steps. Unacceptable or adverse outcomes in particular are attributed to malfunctions or failures of components or the incorrect performance of an activity or a step, thus preserving a congruence between the valence of causes and the valence of consequences. When a cause in this way has been found, the response is to try to fix it, preferably by removing or eliminating it and if that is not possible then to try to reduce the likelihood that it will happen again.

This "find-and-fix" approach implies what may be called a "hypothesis of different causes" which states that things that go well and things that go wrong happen in different ways and have different causes. If that was not the case then the endeavour to "find and fix" the causes of unacceptable outcomes would also affect the occurrence of acceptable outcomes. This hypothesis explains why safety management usually pays little or no attention to the expected and acceptable outcomes, to activities that go well.

The FRAM – and Resilience Engineering – takes a different approach, namely that things that go well and and things that go wrong happen in much the same way. According to this *principle of equivalence*, acceptable outcomes as well as unacceptable outcomes are due to the ability of organisations, groups and individuals successfully to adjust to expected and unexpected situations. When something is done, it is done with the intention of bringing about an acceptable outcome. This is also what happens in most, and indeed nearly all, cases. But since the adjustments are approximate rather than precise, the outcomes may every now and then be different from what was expected or even be outright unacceptable. The fact that the outcomes are different does, however, not mean that the explanations also must be so. As Ernst Mach observed, "Knowledge and error flow from the same mental sources, only success can tell one from the other".

Second Principle: Approximate Adjustments

There are two main reasons why today's socio-technical systems cannot be specified in minute detail. One is that neither humans nor organisations are "machines". The other that socio-technical systems are not designed and built but "grow" in a more or less predictable fashion. Indeed, this has been the case for a long time, although it only recently has become widely recognised. Since work and work environments are always underspecified, it is necessary that work is continuously adjusted to the existing conditions (resources, time, tools, information, requirements, opportunities, conflicts, interruptions) in order for it to go well. These adjustment are made by individuals, by groups and by organisations and take place at all levels, from the performance of a specific task to planning and management. Furthermore, since resources (time, materials, information, etc.) almost always are limited and uncertain, the adjustments will typically be approximate rather than precise. This is rarely critical because the approximations usually are close to the mark and because people mostly know what to expect and are able to compensate appropriately. The *principle of approximate adjustment* can thus be seen as an elaboration of the principle of equivalence. It explains why things predominantly go well, but also why they occasionally go wrong.

Third Principle: Emergent Outcomes

Our understanding of the world around us, not least in relation to how we manage it, rests on what has been called a *causality credo* – a belief in cause-effect or cause-consequence reasoning. As a fundamental principle causality means that two events can be related so that the first (the cause) is solely or partly responsible for the second (the effect) and the second is solely or partly dependent on the first. An effect can in turn be a cause of, or causal factor for, many other effects, and so on. The first part of the causality credo states that outcomes happen because something has happened before, but also implies that if the outcomes are adverse or negative – such as an accident – then the cause is also negative – such as a malfunction or an error. The second part states that causes can be found and treated by a process of rational deduction, given sufficient information and time. The third parts concludes that all accidents therefore are preventable (the so-called zero harm principle). (The causality credo is however rarely stated explicitly, but is simply taken for granted.) The two first principles of the FRAM offer an alternative way of understanding how things happen by pointing to the role of performance adjustments. Resilience engineering and Safety-II, and therefore also the FRAM, further recognise that the consequent variability of individual functions is rarely large enough to serve as the only cause of an effect, in particular as the only cause of something going wrong. Neither is the variability usually large enough to be classified as a failure. The variability of two or more functions may on the other hand combine in unexpected ways that can lead to outcomes that both are unpredictable and disproportionate in magnitude – negative as well as positive. Acceptable and unacceptable outcomes can in this way be explained as *emerging* from variability due to the everyday adjustments rather than as a result of single or multiple cause-effect chains arising from a malfunction or failure of a specific components or parts.

Fourth Principle: Functional Resonance

As an alternative to linear causality and cause-effect reasoning, the FRAM proposes that the variability of two or more functions can coincide and either dampen or amplify each other. The variability of one function may in this way come to affect the variability of other functions in analogy with the phenomenon of resonance. In physical systems, classical (or mechanical) resonance means that a system can oscillate with larger amplitude at some frequencies than at others, known as the system's resonant (or resonance) frequencies. At these frequencies even small external forces that are applied repeatedly can produce large amplitude oscillations, which may seriously damage or even destroy the system. In analogy with that functional resonance is defined as the detectable (supraliminal) variability that emerges from the unintended interaction of the everyday (subliminal) variability of multiple functions. Functional resonance is more formally defined as the noticeable performance variability in a socio-technical system that can happen when multiple approximate adjustments coincide. There is some regularity in how people behave and in how they respond to unexpected situations – including those that arise from how other people behave. Since the resonance effects are a consequence of how the system functions, the phenomenon is called functional resonance rather than stochastic resonance. Functional resonance offers a systematic way to understand outcomes that are both non-causal (emergent) and non-linear (disproportionate).

How to build a FRAM model

Having thus taken care of the preliminaries, we can proceed to describe how the FRAM is used. The Functional Resonance Analysis Method consists of four steps:

- Identify and describe the important system functions and characterise each function using six basic characteristics (called aspects). Together, the functions constitute a FRAM model.
- Characterise the potential variability of the functions in the FRAM model, as well as the possible actual variability in one or more instantiations (or realisations) of the model.
- Determine the possibility of functional resonance based on dependencies / couplings among functions given their potential / actual variability.
- Develop recommendations on how to monitor and manage the variability, either by attenuating variability that can lead to undesirable results, or by enhancing variability that can lead to desired results.

This handbook will focus on the first and the second step, but also say something about the third step. The fourth step addresses important issues in system and safety management for which a FRAM model can provide a valuable contribution, but which go beyond the basics of the FRAM.

FRAM Step #1: Identifying and describing functions

The building of a FRAM model must obviously start by identifying and describing the functions that taken together make up the activity that is analysed. Once the functions, or an initial set of functions, have been identified, they are then described using the six aspects.

What is a function?

It is common in human factors – using the term broadly – to refer to tasks and task analysis, and sometimes even to activities and activity analysis. One way to start thinking about functions is to see them as corresponding to tasks and activities.

In the FRAM a function represents the means that are necessary to achieve a goal. More generally, a function represents the acts or activities – simple or composite – that are needed to produce a certain result. Unlike tasks, functions can be carried out in different ways and by different means.

- A function typically describes what people – on their own or together – have to do to perform a specific task and thus achieve a specific goal. Examples of human functions are to triage a patient or to fill a glass with water.
- A function can also refer to something that an organisation does. The function of the emergency room in a hospital, for example, is to treat incoming patients while the function of a restaurant is to serve food.
- A function can finally refer to what a technical system does either by itself, such as a washing machine or an “intelligent” assistant, or together with one or more people (an interactive or socio-technical function), such as a flight management system.

A FRAM function is thus more than a task as the term is commonly used. A function describes what needs to be done, but not necessarily how or by what means. Following a long tradition in safety studies, functions are categorised as either human functions, organisational functions, and technological functions. This is, however, done mainly for practical reasons rather than because the categories represent any deep theoretical insights.

In the description of functions an important distinction can be made between tasks and activities, corresponding to the distinction between Work-as-Imagined (WAI) and Work-as-Done (WAD). A tasks describes work as designed, or as imagined by managers. An activity describes work as it is actually performed or done. The FRAM primarily focuses on activities as they are done or WAD, but can of course also be used to model WAI.

A general rule for the use of the FRAM is that a function should be described by a verb (verb) if it is a single word, or a verb phrase – in both cases using the infinitive form. For instance “(to) diagnose a patient” rather than “diagnosing a patient” or “(to) start a pump” rather than “starting a pump” or “pumping water”.

Example: Preparing Delicious Cup Noodles

The example used to illustrate how to build a FRAM model is really simple. It is how to prepare cup noodles. Usually, the instructions are written on the lid, for instance as follows:

- Tear open half of the paper/plastic lid. ...
- Pour some boiling (100 C/212 F) water up to the inside line of the container.
- Put the lid on and weigh down the lid with a set of chopsticks or a light plate.
- Wait patiently for 2 to 3 minutes then stir the noodles well with chopsticks.
- Eat and enjoy.

Try to write the above five parts or actions of the instruction as functions, i.e., using a verb phrase <To open half of cup lid>. Consider whether each action corresponds to a single functions or to more than one function. Also consider whether some functions (actions) are missing from the description, perhaps because they are taken for granted, because it is something that everybody knows, etc.

(If you think preparing cup noddles is too simple, you can try to model how to get money from an ATM instead. A description of that can be found on wikihow https://www.wikihow.com/Use-an-ATM.)

The first function

So where should the model building begin? In principle, a FRAM model can start from any function. This is because the method itself ensures that the model will be complete, regardless of where it starts. Nonetheless, it may be a good idea to start with a function that is central to the activity being analysed.

Just describe the functions in plain text. Do not use the FMV at this point in time. You can describe the functions in the order they are presented above, which is basically the recommended sequence in preparing cup noodles. Or you could write them in alphabetical order or in any other way. When building a FRAM model the functions can be described in any order you like. Indeed, it may in many cases be an advantage not to think in a sequence from the start, but to write down the functions as they come to mind. As you progress, the method will ensure that you have all the functions that are necessary. But not necessarily in any specific order, because a FRAM model is *not* a flow model.

Breadth before depth

When a FRAM model is developed it is advisable to describe all the important functions before beginning to consider the aspects of the functions. In the cup noodle example that would probably mean all the functions that are included in the instructions – perhaps leaving out the final consumption. For more complicated activities it is advisable to start by a handful, say four to six, main functions rather than to try to be exhaustive. This can be seen as applying the principle of breadth-before-depth. In other words, try to characterise the activity as a whole before you being to develop the details.

Many analysis methods implicitly support a depth-before-breadth approach. (A characteristic example of that is the root cause analysis.) In a depth-before-breadth approach the items or parts are analysed in detail one by one before their possible relations are considered. Examples of that are FMEA/FMECA and HAZOP. Using a breadth-before-depth approach prevents the analysis team from becoming distracted by the details of the investigation at hand, especially if the analysis is of an event that has happened.

Describing functions using the FMV

At this point in time you should begin to use the FMV. Depending on how many functions you defined, the outcome may look something like Figure 1.

In this example six functions have been defined. At this stage of development they are not connected or coupled, and their position on the FMV canvas is not significant. At the moment they are only identified by their name, which is a verb phrase as recommended. It may be useful even for this simple example also to write something in the Description field, for instance who or what carries out the function. The Description field serves as a scratch-pad for things it may be useful to remember for later, things that were discussed but not included, etc.

Assuming that you are satisfied with this initial model in the sense that there are no other functions that need to be included, the model development continues by characterising the functions using the six aspects.

What are the FRAM aspects?

The FRAM characterises functions by means of six aspects named Input, Output, Preconditions, Resources, Control, and Time, respectively.[1] The general rule of the FRAM is that an aspect of a function should be described when it is seen as necessary or appropriate

1 The aspects can be thought of as features or dimensions of a function. In the FRAM, the six aspects are written with capital initials when they refer to aspects as part of a functional description (e.g., Time or Control), but without the capital initial when used in other contexts (e.g., timely or effective control).

by the analysis team, provided there is sufficient information or experience to do so. It is thus not necessary to describe all six aspects of every function, and it can indeed sometimes be either impossible or unreasonable to do so.

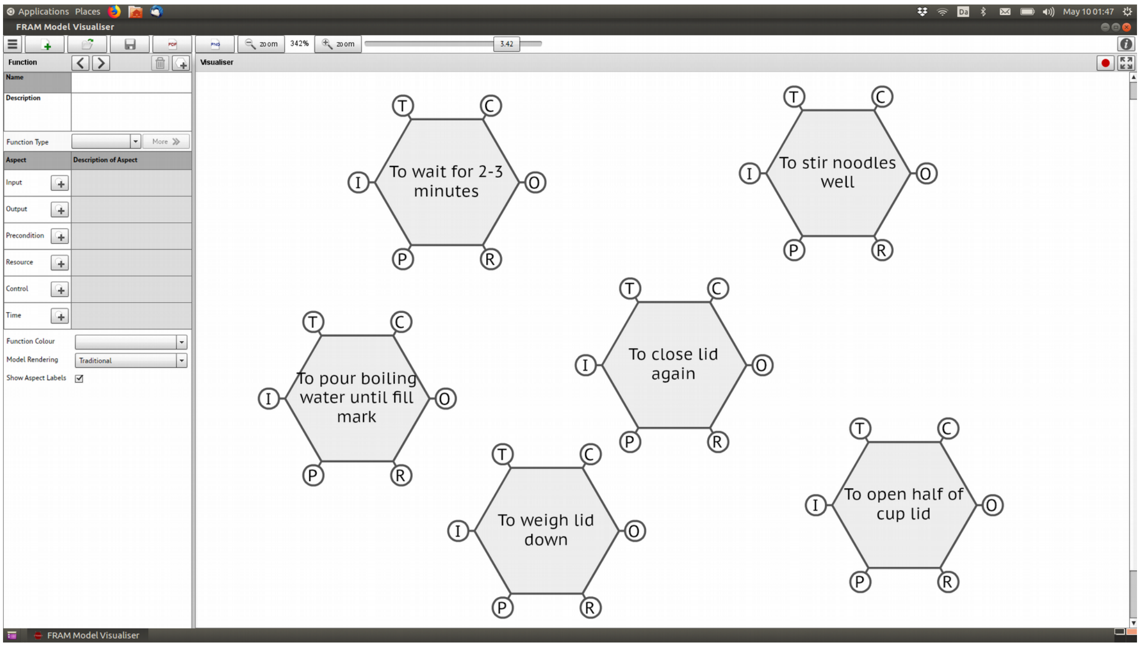


Figure 1: Six functions in the preparation of cup noodles

The guidelines for which aspects to describe and when, are given in the following. As a minimum, at least one Input and one Output must be described for all foreground functions. (The meaning of a foreground function is also explained in the following.) Note, however, that if only the Input and Output aspects are described then the FRAM model is reduced to an ordinary flow chart or network diagram. A general rule in the use of the FRAM is that an aspect is described using a noun (noun), or a noun phrase. In other words, an aspect is described as a state or as a result of something – but not as an activity.

A brief description of the six aspects

- **Input**. The Input to a function is traditionally defined as that which is used or transformed by the function to produce the Output. The Input can represent matter, energy, or information. There is, however, another meaning that is just as important for the FRAM, namely the Input as that which activates or starts a function. The Input in this sense may be a clearance, an instruction, or even a command to begin to do something. Input can be seen as a form of data or information, or more generally as a state change that is recognised by a function as a signal to begin. It is for that reason that the description of the Input is always a noun or a noun phrase.

  Something that is defined as an Input to one function must clearly be defined also as an Output of another function – or functions. An Input cannot come out of nowhere.

  In the FRAM, designated foreground functions must have defined Inputs, while designated background functions need not have. The difference between foreground and background functions will be explained later.
- **Output**. The Output of a function describes the result of what the function does, for example, the result of processing the Input. In the cup noodle example an Output from the function <To wait for 2-3 minutes> could be [Tender noodles]. The Output can

represent material, energy, or information – an example of the latter would be a permission or clearance, or the result of a decision. The Output describes a change of state – of the system or of one or more output parameters. The Output may, for example, be the signal to start a function. The description of the Output should be a noun or a noun phrase.

Something that is defined as an Output from one function must clearly also be defined as either an Input, Precondition, Resource, Control, or Time of another function – or functions. An Output cannot be left dangling but must end somewhere.

- **Precondition**. In many cases it may not be possible to begin a function before one or more Preconditions have been established. These Preconditions can be understood as system states that must be [True], or as conditions that ought to be verified before a function is carried out. A Precondition does, however, not itself constitute the signal that starts the function. An Input, on the other hand, can activate a function. This simple rule can be used to determine whether something should be described as an Input or as a Precondition. It is however not necessarily critical for a FRAM analysis whether something is labelled Input or Precondition, as long as it is included in the model in one way or another. A Precondition can, of course, not come out of nowhere but must always be defined as an Output from another function – or functions. The description of a Precondition should be a noun or a noun phrase.
- **Resources or Execution Conditions**. A Resource is something that is needed or consumed while a function is carried out. A Resource can represent matter, energy, information, competence, software, tools, manpower, etc. Time could, in principle, also be considered as a Resource, but since Time has a special status it will be treated as a separate aspect.

  Since some Resources are consumed while the function is carried out and others are not, it is useful to distinguish between (proper) Resources on the one hand and Execution Conditions on the other. A (proper) Resource is consumed by a function; it will be reduced over time and must therefore be renewed or replenished. An Execution Condition only needs to be available or exist while a function is active but is not consumed in the same way that a (proper) Resource is. (The difference between a Precondition and an Execution Condition is that the former is only required before the function starts, but not while it is carried out.) The description of a Resource (an Execution Condition) should be a noun or a noun phrase. Something that is defined as a Resource for a function (or for two or more functions) must also be defined as an Output from a function, or functions. Resources do not come out of nowhere. This is especially the case for (proper) Resources.

  An example of a (proper) Resource could be blood plasma used by a transfusion, e.g., during surgery. An example of an Execution Condition could be the competence to disassemble a machine before it is repaired. After the surgery has been completed, some or all of the blood plasma will have been consumed; after taking a machine apart the competence will be the same – if not actually increased.
- **Control**. Control is that which supervises or regulates a function so that it produces the desired Output. Control can be a plan, a schedule, a procedure, a set of guidelines or instructions, a program (an algorithm), a ‘measure and correct’ functionality, etc. Another, less formal type of control is social control or expectations to how the work should be done. Social control can be external, such as the expectations of others (management, organisation, co-workers) and is sometimes expressed explicitly. Social control can also be internal, for example, when we plan a job and mentally go through when and how to do it, or when we imagine what others expect of us. The description of Controls should be a noun or a noun phrase. Something that is defined as a Control for a function (or for two or more functions) must also be defined as an Output from another function, or functions.
- **Time**. This aspect represents the various ways in which time can affect how a function is carried out. Time, or rather temporal relations, could be seen as a form of Control, as

when Time represents the order of two or more actions (sequencing conditions). A function may, for instance, have to be carried out (or be completed) before another function, after another function, or overlapping with – parallel to – another function. Time may also relate to a function alone, seen in relation to either clock time or elapsed time.

Time could also be seen interpreted as a Resource, such as when something must be completed before a certain point in time, or within a certain duration. Time could, of course, also be interpreted as a Precondition, e.g., that a function must not begin before a certain time or that it must not begin before another functions has been completed. Yet rather than having Time as a part of either of the other three aspects of a function it seems reasonable to acknowledge its special status by having it as an aspect in its own right. The description of Time should be a noun or a noun phrase. Something that is defined as a Time aspect for one or more functions must also be defined as an Output from another function, or functions.

Please remember that 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. It is always possible to update a model either by describing additional aspects of functions or to remove definitions of aspects for some functions.

Keep in mind that there can be more than one entry for each type of aspect. A function may, for instance have a single Output that used by several downstream functions. Or it may have several Outputs that are used by several downstream functions. Similarly, a function may have multiple Inputs coming from more than one other function, multiple Preconditions, etc. A function may in general have two – or more – upstream couplings to its Input, Precondition, Resource, Control, or Time aspects. The decision of which aspects to describe for which functions is taken by the analysis team based on their understanding of and experience with the activity or process that is being analysed and modelled.

A short definition of the six aspects can be seen in Figure 2 below.

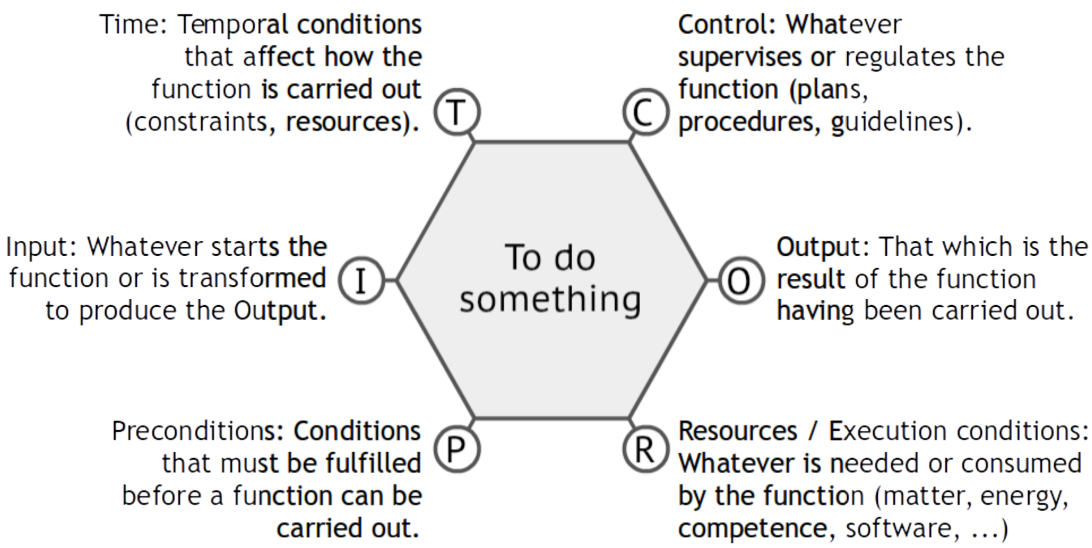


Figure 2: The six aspects used to characterise functions

Aspects in the cup noodle example

We can illustrate how aspects are defined by adding some details to the initial model of how to make cup noodles. Before you read any further, please try to do it on your own, following the guidance given above.

… …

We have already seen that one possible Output from the function <To wait for 2-3 minutes> is [Tender noodles]. (Beginning with this function also demonstrates that there is no need to consider them in any specific order.) If we stay with that function, one necessary question is when it starts, which simply means when does a person begin to wait for the noodles to become tender. In this example the answer is that the waiting begins when the noodles have been covered with boiling water. So we can now define an Input for the function as [Noodles covered with boiling water]. We can then continue to consider the four remaining aspects (P, R, C, and T) and ask whether there is a need to describe them. Waiting for 2-3 minutes does not seem to require any Preconditions, nor any specific Controls. There could arguably be the need of a Resource, namely a clock or a timer. But for the time being we shall disregard that. But it is necessary to define the T aspect, since the 2-3 minutes are important for the preparation of the noodles. We therefore describe the Time as [2-3 minutes]. Having done that, the model may now look as in Figure 3.

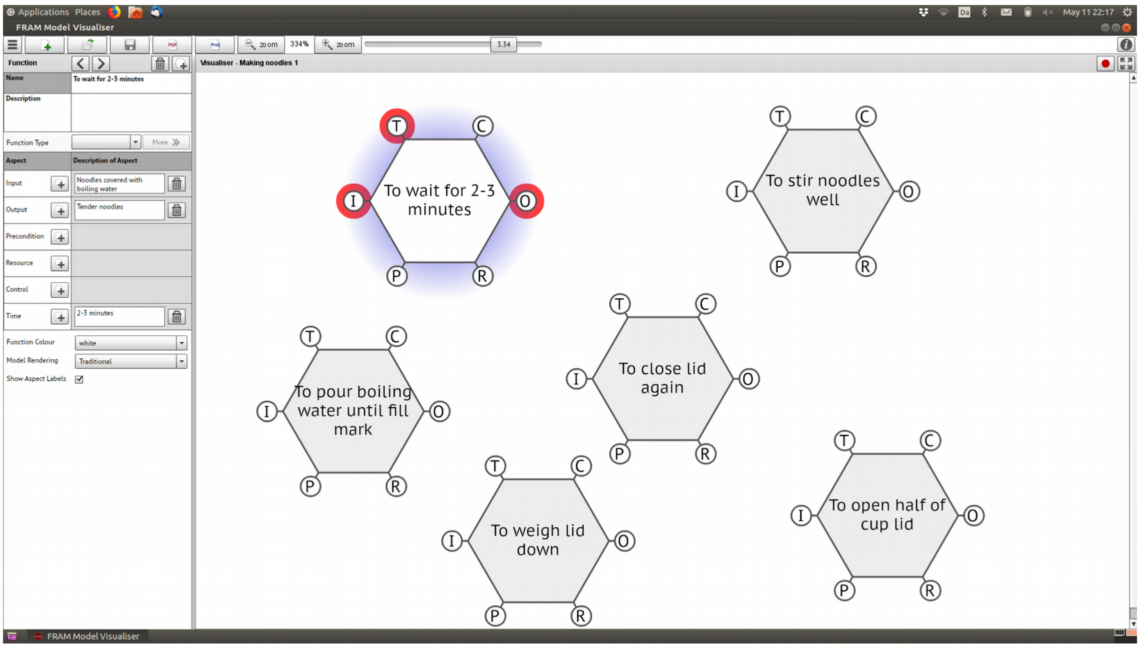


Figure 3: Cup noodles - the first function

The development of the model can continue in several ways – none of them being preferable over the others. One way is to look at the other functions in the same way and try to define as many of their aspects as seems reasonable and possible. Another way is to try to define aspects that are incompletely described in the current version of the model.

As Figure 3 shows, the three aspects that have been defined for <To wait for 2-3 minutes> are all marked by a red ring. This is done automatically by the FMV to point out that an aspect only has been defined for one function, hence is incompletely described. In the specific case two of the aspects (I and T) must come from somewhere and therefore be defined as Outputs from other functions, while one aspect (O) must end somewhere and therefore be defined as an (I, P, R, C, or T) for another function.

If we first look at the Input [Noodles covered with boiling water] then that seems to be an obvious Output from the function <To pour boiling water until fill mark>. If we then look at the Output [Tender noodles], then that would reasonably serve as an Input to <To stir noodles well>, since the stirring clearly should not begin until the noodles are tender. But the Time aspect [2-3 minutes] does not seem to be an Output from any of the six functions in the current version of the model. It therefore becomes necessary to introduce a new function that provides this Output. That function could be that the user reads the instructions on the lid or, perhaps less straightforward, the writing of the instructions at some earlier time. Although it is tempting, the function cannot simply be, e.g., <Instructions> since that is not an action or a verb phrase. In the example we will choose to introduce a new function called <To read the instructions> which has the Output [2-3 minutes]. (As a consequence of that it also seems reasonable to change the name of <To wait for 2-3 minutes> to be simply <To wait until tender>, since the time in which to wait is specified by the Time aspect.) The result is a new version of the model which may look as in Figure 4.

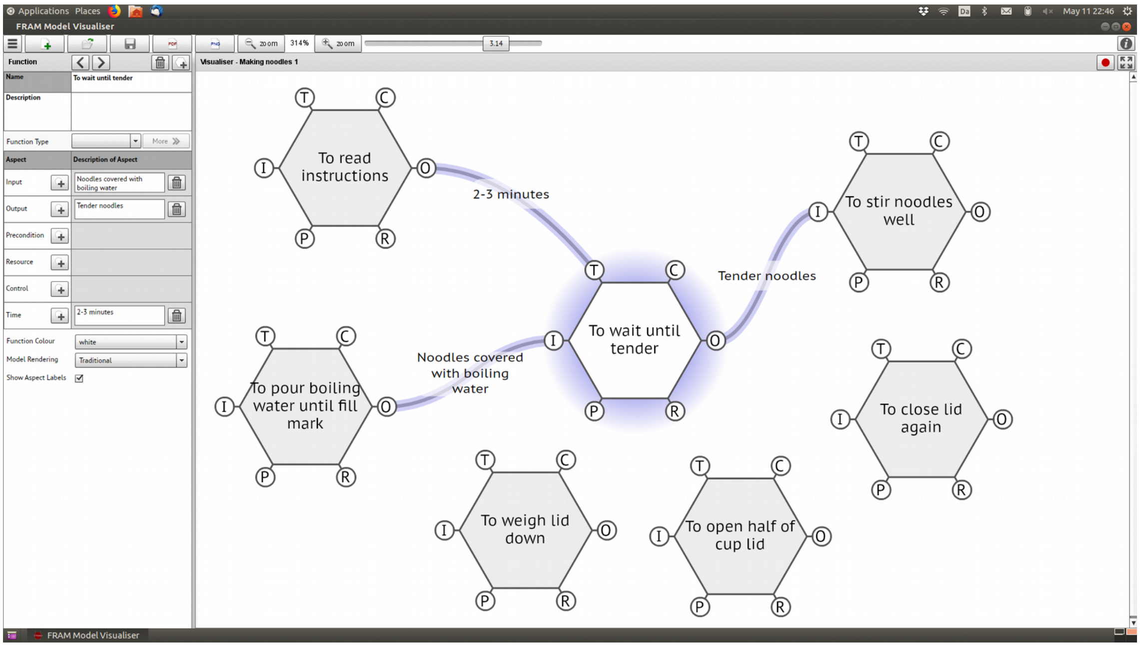


Figure 4: Cup noodles – the second version of the model

Couplings

As explained above and as shown by the development of the “cup noodles” model, every function in a FRAM model is characterised by using some or all of the six aspects. If the same values (names) are defined for aspects of different functions – for instance the Output of one function and the Time of another – then there is a potential coupling between the functions. This is shown by the FMV as a line that connects the two aspects. The line is not shown as having a direction (for instance by an arrowhead), but the logic is clearly that the Output goes from one function to other functions.

The significance of the couplings is perhaps easier to see if the model is elaborated a bit further. An example of what that could look like is shown in Figure 5. Apart from the definition of further aspects of the functions, the most conspicuous differences between Figures 4 and 5 are that the functions have been rearranged on the FMV canvas, that two functions have been collapsed into one, and that three additional functions have been introduced, namely <To boil water>, <To make preparations>, and <To enjoy cup noodles>, respectively. (The reader may have produced a different expansion, but that is no concern for worry.)

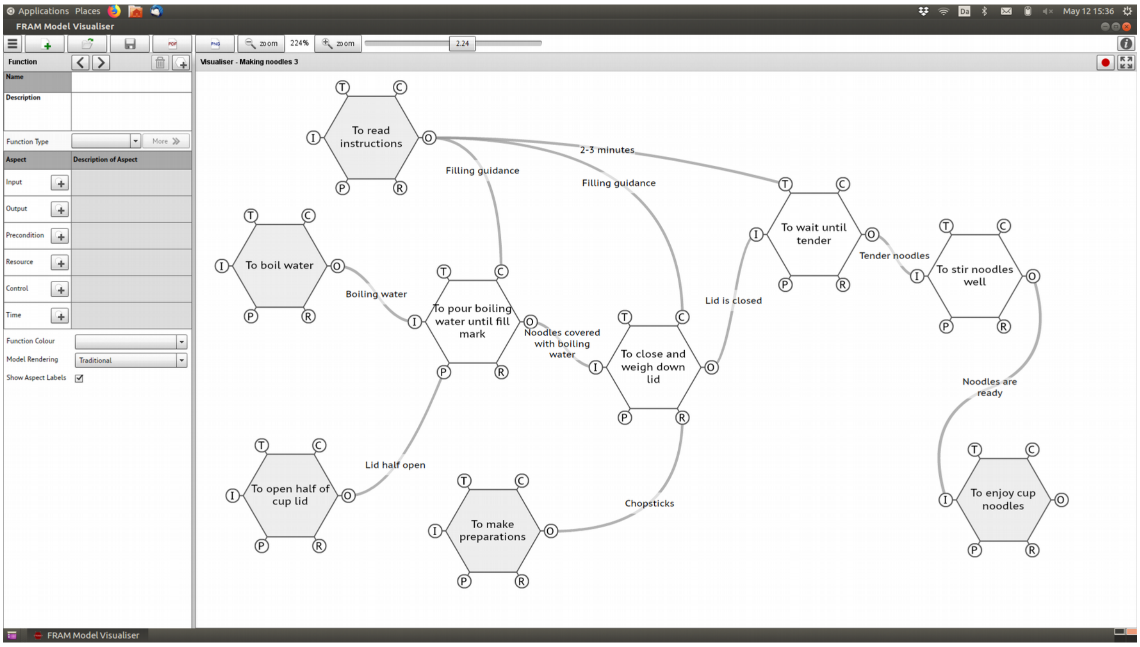


Figure 5: Cup noodles - the third version of the model

The basis of the FRAM is the description of the functions that make up an activity or a process. The description starts by the functions themselves, and not by how they are ordered or related. The relationships are not specified nor described directly and the FMV in fact does not allow lines or connectors to be drawn between functions. Relationships are instead specified indirectly via the descriptions of the aspects of functions. The common technical term for such relations is couplings. In Figure 5, for example, the Output [Lid half open] from <To open half of cup lid> is also defined as a Precondition of the function <To pour boiling water until fill mark> – reflecting the simple fact that it is impossible to pour water into the cup unless the lid is open or half open. Because the two functions share the same aspect, they are potentially coupled.

The couplings in a FRAM model are generally n-to-n (or many-to-many) rather than 1-to-1. For example, the function <To read instructions> has two different Outputs which serve as Control for three other functions. The same Output, [Filling guidance], is defined as a Control aspect of both <To pour boiling water until fill mark> and <To close and weigh down lid>.

The couplings that in this way are described in a FRAM model, i.e., the dependencies that are the consequence of shared aspect attributes, are called *potential couplings*. This is because a FRAM model describes the potential or possible relationships or dependencies between functions without referring to any particular situation. In an instantiation of a FRAM model, only a subset of the potential couplings can be realised; these represent the *actual couplings* or dependencies that have occurred or are expected to occur in a particular situation or a particular scenario. An instantiation of a FRAM model thus represents how a subset of functions can become mutually coupled under given conditions or within a given time frame. The couplings realised for a specific instantiation do not vary but are assumed to be are 'fixed' or 'frozen' as long as the conditions exist. For an event analysis the instantiation will typically cover the entire event and the couplings that existed at the time. For a prospective analysis, the duration is the time required for the activity under consideration to be completed. This can vary significantly, from the few minutes it takes to prepare cup noodles to the weeks or months required for a large-scale industrial or business operation.

Foreground and background functions

The cup noodle example has already shown how the description of functions can be expanded and how new functions become necessary. This does in principle raise the perspective that any given model can be expanded forever, since there always will be details that conceivably could be added to the model. The FRAM, however, includes a practical stop rule that can be used to limit the size of the models developed.

Functions in the FRAM can be characterised either as *foreground functions* or *background functions*. The terms have nothing to do with the type of functions that are involved, but with the role of a function in a particular model – and of course also in the instantiations of the model. A function is considered as a foreground function if it is part of the study focus, which in practice means if the variability of the function may have consequences for the outcome of the event or process being examined. A background function is similarly a function which can be assumed not to vary – i.e., which can be assumed to be constant – during the duration of the process or activity being analysed.

In Figure 5, there are four foreground functions and five background functions. Foreground functions are white while background functions are shaded grey. The determination of whether a function is a foreground or a background function is made automatically by the FMV using the following rules:

- A function that only has an Output – or Outputs – defined, is designated a background function.
- A function that only has an Input – or Inputs – defined is designated a background function.
- All other functions are designated foreground functions.

Because background functions are assumed not to be variable while the activity being analysed takes place, there is no need to expand them further by describing their Preconditions, Controls, etc. A background function can thus be seen as being part of the boundary of the system being analysed. Background functions typically represent something that is used by foreground functions, but which is assumed to be stable during the situation under consideration. It could, for example, be a Resource (the right level of staffing or the competence of the staff) or an instruction (Control). A person's competence must generally be assumed to be stable (not varying) during the execution of a task, just as an instruction also must be assumed to be stable. This does not mean that the competence is sufficient or that the instruction is correct, but only that they are assumed not to vary during the time it takes to perform the task. While the execution of an instruction may vary, the instruction itself only changes in case it is corrected or modified. The instruction is therefore only variable when considered over a longer time span, which is typically many times longer than the duration of the event. In that case the focus would change to be the writing and maintenance of the instructions, which means that this became the function.

Background functions may be used deliberately to stop the expansion of functions and thereby limit the scope of a model. But a background function can at any time be changed to a foreground function, for instance by defining one of its other aspects (P, R, C, or T) which in turn means that the model must be expanded. The terms foreground – background function thus refer to the relative importance of a function in the model and not to the "nature" of a function as such. If the study focus changes, a function may change from being a designated foreground function to become a designated background mode, and vice versa.

In most cases the background functions serve as the source of an Output that has been defined as an (I, P, R, C, or T) aspect of one more other functions. Figure 5, however, also includes the background function <To enjoy cup noodles> for which the Input rather than the Output has been defined. This function is needed because the Output [Noodles are ready] from the function <To stir noodles well> must be defined for (at least) one other function, i.e., the Output must end somewhere. In this case it seems natural to introduce the function <To enjoy cup noodles> not least because that presumably is the purpose of preparing the noodles in the first place. But since the model represents the activity of *preparing* the cup noodles, it is not necessary to go into the details of how the noodles are consumed. The background function in this case is said to serve as a drain or sink, i.e., as a destination for the Output. A FRAM model will typically have several background functions that serve as sources as well as one or more background functions that serve as drains or sinks.

Upstream and downstream functions

While the terms foreground and background represent a function's role in a model, the terms *upstream* and *downstream* are used to describe the temporal relationship between a function that currently is in focus and the other functions. The analysis of the FRAM model takes place by tracing the potential couplings as they lead from function to function. This means that there will always be one or more functions that are in focus, i.e., whose variability is currently being considered. The functions that have been in focus before, which means functions that already have been carried out, are referred to as upstream functions. Similarly, the functions that follow the function (or functions) that is in focus, are called downstream functions. During an analysis, any function can change status from being downstream, to come into focus, and to become an upstream function.

A FRAM model describes the functions and their potential couplings for a typical situation, but not for a specific situation. It is therefore not possible to say with certainty whether a function always will be performed before or after another function. That can only be determined when the model is instantiated. A FRAM model is therefore not a flow model or a network where links or transitions are defined once and for all. By contrast, the labels foreground function and background function are valid both for the FRAM model as its instantiations. An instantiation of the model uses detailed information about a particular situation or scenario to create an instance or a specific example of the model. This corresponds to a temporal organisation of functions that reflects the order in which they will take place in the scenario, depending on how much variability there is. An upstream function is a function that for a given instantiation is performed **before** others, and therefore may affect them. Functions that – in the instantiation – take place **after** other functions and therefore can be affected by them, are called downstream functions. The terms upstream and downstream function are thus relative and not absolute.

Continuing the development of a model

The model in Figure 5 shows how an initial FRAM model of an activity can be developed. Even in this simple example there are several things that need to be developed further, for instance how the boiling water is produced. The continued model development can be guided by the following questions:

- Is there a need of additional functions in the model, i.e., are there parts of the activity that have not been described because they are taken for granted?
- Are there any foreground functions where the Input has not been defined? If the Input to a function is missing it means that the function will never be carried out, in which case there is no need to include it in the first place. Similarly, are there any foreground functions where the Output has not been defined? If the Output is missing from a foreground function it means that the Output is never used for anything, in which case there is no need to include the function in the model. Note, however, that this only applies to foreground functions, but not to background functions, cf., above.

- Are there any functions where the FMV has marked any of the aspects with a red ring?
- Are there any background functions from which the Output could be variable? In that case they need to be considered in further detail or "expanded".

It is important throughout the model development to look critically at the model and make sure that it "makes sense" vis-a-vis the activity in focus. This can for instance be done by showing it to people who are familiar with the activity in question but who are not part of the analysis team.

An important issue is what information is needed to develop a model and where it can be found. The example of preparing cup noodles has been chosen because it describes an activity that most people either have sufficient knowledge about or easily can imagine doing. But that cannot be assumed to be the case for a FRAM model of a "real" activity. In these cases information must come from extensive formal or informal knowledge and experiences with the activity. (See Appendix A for a summary guidance on how to do so.) In many cases the analysis team includes people who have at least part of that experience; in other cases the identification and elicitation / recording of that requires supplementary work such as interviews, site visits, case studies, etc.

FRAM Step #2: Characterising the Variability of Functions

The FRAM is a tool to describe or represent how an activity usually is carried out. The activity is described in terms of the functions necessary for carrying it out, the potential couplings among the functions, and the typical variability. A FRAM model can be used to understand how the variability and adjustments of one function can affect other functions and thereby the activity as a whole. Functions can mutually dampen each other (absorb variability), so that a situation is stabilised. Functions can also mutually reinforce each other (amplify variability) so that a situation becomes unstable and leads to unexpected and often unwanted results. The description of the potential couplings can be used to understand emergent outcomes and also how the building up of these can be monitored and managed.

After having described the initial set of functions, the model building continues by making sure that all aspects are completely specified, using the following rules:

- Every aspect of a foreground function has to be an Output of another function.
- Every Output from a foreground function must be a *non-Output aspect* of another function (Input, Precondition, Resource, Control, Time).

It may seem daunting to describe so many functions especially because some of the new functions will have aspects that in turn require more functions. This is where the importance of distinguishing between foreground and background functions becomes useful. Since background functions only need their Output to be defined, the expansion of the model stops there.

The process of checking whether all aspects have been defined is built into the FMV. In the graphical rendering, aspects are marked with a red circle if they have been incompletely defined. A FRAM model is syntactically complete when there are no 'loose' aspects, as shown by the FMV. As explained above, a FRAM model can in principle start from any function because the method itself ensures that all necessary functions will be identified and are described.

How to describe the variability?

The characterisation of variability in a FRAM model is the starting point for understanding how the coupling among functions can lead to unexpected results. The analysis focuses more on the variability of the Output of functions than the variability of functions as such. The reason is simply that if the performance of a function is variable without this showing up in Output, then the variability is in principle not important. However, if the Output of a function is variable, then the variability of the function is of interest because it is that which determines the characteristics hence the quality of the Output.

There can be three different reasons why the Output of a function is variable.

- The variability of the Output can be a result of variability of the function itself, i.e., a consequence of the function's uniqueness or character. This is called **internal** or **endogenous variability**.
- The variability of Output may be due to variability of the work environment, i.e., the conditions under which the function is performed. This is called **external** or **exogenous variability**.
- The variability of Output may finally be due to variability of the Output from upstream functions that provides the Input, Precondition, Resource, Control, or Time for downstream functions. This type of coupling is the basis for functional resonance and is called **functional upstream-downstream coupling**.

The variability of a function can, of course, also be due to a combination of the three conditions, i.e., internal variability, external variability and upstream-downstream couplings.

The variability of the different types of functions

The FRAM assumes that there are characteristic differences in the variability of technological functions, of human functions, and of organisational functions. (This three-way split corresponds to a traditional distinction between T, M, and O – Technology, huMan, and Organisation.)

- Technological functions are performed by different types of 'machines', which in most cases include information technology. As technological functions are designed to be both predictable and reliable, the default assumption in the FRAM is that they have no significant variability. This does not exclude that technological functions can be variable; it only means that the FRAM considers them to be stable unless there is reason to believe otherwise.
- Human functions are carried out either by individuals or small groups (formal or informal). A FRAM analysis assumes that human functions are variable with high frequency and large amplitude. The high frequency means that performance can change quickly or even abruptly. People react promptly to changes, especially in response to other people. The large amplitude means that differences in performance can be large, sometimes dramatically so – for better or worse. The variability depends on many different things, including the working conditions. One purpose of FRAM is to provide a clear and comprehensive description of such dependencies.
- Organisational functions are performed by a group of people, sometimes very large groups, where activities are explicitly organised in some way. Although organisations clearly consists of people, organisational functions differ from human functions and are usually described – and defined – on another level. They are thus functions of the system itself, rather than of the people working in the system. A FRAM analysis assumes that the frequency of organisational variability typically is low, but that the amplitude is large. The low frequency means that organisational performance changes slowly, but that the differences in the results, i.e., amplitude, can be very large.

Endogenous variability

The endogenous variability can influence technological, human, and organisational functions alike.

- Technological functions can vary because the 'inner workings' often are so complicated that it is not really known how the technology functions – it is intractable or underspecified. This may be true for pure 'mechanical' systems, and is clearly the case for software systems. Variability can also be caused by the inevitable degradation of the physical components. Apart from these, there are no other significant sources of internal variability of the technological functions.

- Human functions can vary due to physiological and psychological factors. (Social factors are treated as an external source of variability in the FRAM.) Fatigue and stress (workload) are probably the most studied of the several physiological factors. Other factors are diurnal rhythm, well-being (or illness), various physiological needs, temporary disabilities, etc. There are also many different psychological factors that may affect the performance of a task, such as personality traits, cognitive style, bias in assessment and decision-making, etc.
- Organisational functions can vary for several reasons, such as how effective communication is, the authority gradient, confidence, organisational culture, organisational memory, etc.

Exogenous variability

The exogenous variability can also influence technological, human, and organisational functions alike.

- Technological functions can vary due to improper maintenance, inappropriate operating conditions – especially if they exceed the design specifications, faults in measuring instruments and sensors, overloading, misuse, etc.
- Human functions can vary due to social factors (peer pressure, unspoken norms and expectations, and so on), and because of organisational factors such as expectations, standards, requirements, commercial considerations, political considerations, etc.
- For organisations, the greatest external influence comes from the environment, the physical as well as the legal (regulations) and the commercial. The environment includes customer requirements or expectations, the availability of resources and spare parts, the regulatory environment, commercial pressures, supervisors, as well as weather and other forces of nature.

Even this brief discussion makes clear that there can be many reasons why functions vary when they are carried out, and that no type of function is immune. For simplicity, the default assumption of the FRAM is that technological functions are relatively stable, that human functions vary with high frequency and high amplitude, and that organisational functions vary with low frequency but high amplitude. As a result of this, the variability of human and organisational functions is of most interest, whether it is the potential or actual variability.

Knowledge of both endogenous and exogenous variability should be considered when the model is analysed by the team. This knowledge is used to propose what the potential and / or actual variability of the Outputs from functions will be. Neither the endogenous nor the exogenous variability is represented directly in the model and cannot be captured by the FMV except as part of the Description field.

The manifestations of variability

Once the possible internal and external sources of variability have been identified, it is necessary to describe how the variability will appear in a function's output – how it will manifest itself. This is important both because it provides the basis for observing or detecting variability, and because it gives an idea of how variability can affect downstream functions. The manifestations of variability can in principle be described in two different ways, one simple and one more detailed. The simple way is efficient, but not as thorough as the detailed – which in turn is more thorough, but not as efficient. In practice, it is recommended to start by the simple way, and then later go into more detail if needed.

The simple description characterises the variability of a function's Output in terms of **time** and **precision**.

- In terms of **time**, an Output can occur *too early*, *on time*, *too late*, or *not at all*. (The category 'not at all', equivalent to the traditional 'omission', can be seen as an extreme version of 'too late'. The consequence may be that Output either never occurred or that it occurred so late that it was useless.) An Output that is not available on time can affect the variability of downstream functions in several different ways.

- In terms of **precision**, an Output can be *precise*, *acceptable*, or *imprecise*. (More detailed characterisations can of course also be used.) Because the Output provides the coupling between upstream and downstream functions, the meaning of precision is relative rather than absolute.

  An Output can be described as *precise* if it meets the needs of a downstream function. A precise Output will therefore not increase the variability of downstream functions, and may potentially even reduce it.

  An *acceptable* Output may be used by a downstream function, but will require some adjustment or variability of the receiving function. An acceptable Output may therefore increase the variability of downstream functions.

  An *imprecise* Output is incomplete, incorrect, ambiguous or otherwise misleading. An imprecise Output cannot be used as it is, but requires interpretation, verification, comparison with other data or with the situation as such. These are all things that can increase the variability in the receiving function, typically by consuming resources and time that could and should have been used for other purposes.

The detailed description of Output variability can be with respect to time (too early, too late) and duration (short, long), strength (weak to strong), distance (too long, too short) and direction (wrong direction), object or target (wrong item, wrong recipient), and finally with regard to sequence or order (of two or more sub-activities).

Potential and actual variability

The potential variability describes what might happen under different conditions. The actual variability describes what should reasonably be expected to happen under given conditions (more specific assumptions about demands, opportunities, and resources), i.e., for an instantiation of the model. For technological functions the FRAM assumes that the potential variability is not realised as long as the operating conditions corresponding to the nominal conditions. For both human and organisational functions it is assumed that the potential variability will become realised as actual variability, unless working conditions are absolutely ideal. How the actual variability will express itself depends to a very high degree on the level of detail of the instantiation, i.e., the situation for which the model is analysed. For an event analysis there will often be quite detailed information, corresponding to the facts of the event that took place. For a risk analysis or design, it depends on the specificity of the assumptions made for that scenario.

It is important to distinguish between the potential and actual variability in the description of how variability – or the consequences of variability – can propagate, specifically how variability can affect downstream functions. An analysis of coupling and resonance should only be carried out for an instantiation of a model, and thus for the actual variability – which always will be a subset of the potential variability. For this reason it is wise to begin by describing the potential variability, in order to avoid being unduly restricted by having a specific scenario in mind.

Representing potential variability in a FRAM model

The FMV offers the possibility of indicating the (potential) variability of functions as explained in the instructions on how to use the FMV. For practical reasons it is only possible to indicate the variability of an Output – or Outputs – in terms of time and precisions. More elaborate descriptions can be given in the Description field for a function.

Dependence between the functions

Work involves multiple tasks and sub-tasks, here called functions, a collaboration between different people, and a coordination of their work. Each activity, i.e., the functions a person performs, must be adjusted to the conditions as described previously. But each function also forms part of the conditions of other functions.

This means that the adjustments a person makes at a certain time becomes part of the variability (of the environment) for other (downstream) functions, whether performed by the same person or someone else. For a downstream function, the adjustments of the upstream functions is not known with certainty, although it is rarely completely unknown or unexpected. It can usually be assumed that the upstream (previous) functions have been performed in accordance with established practice. The adjustment to upstream functions thus constitutes a variability that affects the adjustments of subsequent (downstream) functions. In a stable working environment with limited organisational variability (such as changes in demands, resources, personnel, etc.), the variability and adjustments will after a time match each other and thus provide the basis for effective everyday performance (Work-as-Done). In a working environment that is not stable, unexpected and unwanted situations may easily arise.

The previous sections have briefly described the internal (endogenous) and external (exogenous) variability, both how they occur and how this can affect downstream functions – and thus either enhance or suppress other variability. The main reason for variability is, however, the consequence of upstream-downstream couplings, particularly for the actual contexts described in an instantiation of the model. Given that the variability is described for the Output of a function, there can in principle be five different upstream-downstream couplings: between Output and Preconditions, Output and Resources, Output and Control, Output and Time, and finally between Output and Input.

The couplings between upstream and downstream functions make it possible to describe how the variability of the Output of a function can affect other functions without the need of linear cause-effect relationships. This reflects the fact that the way in which an event takes place depends on how the situation develops. A FRAM model can be used to explain how everyday approximate adjustments lead to unexpected results, and thereby how non-linear outcomes occur. The method is presently more qualitative than quantitative, but that is mostly because there are no generally accepted methods of expressing variability numerically.

FRAM Step #3: Looking for functional resonance

A system model is typically used either to consider what happens if an element or component fails or malfunctions or to follow the flow of something, typically energy or information. A fault tree, for instance, is used to deduce how combinations of failures of individual components may lead to the undesired state represented by the top event (root). A hierarchical task analysis is used to understand how activities or tasks are organised in terms of levels and how different activities can be structured, as well as how they can fail.

A FRAM model is, however, not a flow model and neither is it a graph or network model. A FRAM represents the functions that are seen as necessary (and sufficient) to carry out the target activity as well as the (defined) relations or couplings among the functions. The graphical rendering of a FRAM model produced by the FMV does, of course, show the lines between the aspects of the functions. But the lines should be seen as representing potential rather than actual couplings or connection. Upstream functions may affect downstream functions but not in the usual sense of cause-effect relations. An Input to a function therefore does not “cause” the Output. An Input can start a function, but how the function is carried out and what the variability of the performance may be, hence what the quality of the Output will be, depends on the other aspects as well as the endogenous and exogenous variability.

A FRAM model can be used for several types of analysis, retrospective as well as prospective. In some sense this makes the analysis of a FRAM model less straightforward than, e.g., the analysis of a Fault Tree or a Bow Tie. Consider, for instance, a case where the FRAM is used to understand an event that has happened – and which most likely has been noticed because it has gone wrong. The FRAM model is not intended to be a model of the accident scenario but rather a model of the activity when it goes well. By understanding how it goes well, by identifying how the functions are coupled, and by describing the characteristic or typical variability of the functions, it becomes possible to understand why the outcome in the specific case differed from the many other cases where everything went well. Or consider a case where the FRAM is used to represent a future scenario, for instance a new proposal for a guideline or procedure. In this case the FRAM model can be used to look for potentially critical upstream-downstream couplings.

To illustrate that look at the function <Check identity of patient> in Figure 6 below. It has five couplings to upstream functions and three couplings to downstream functions. The multiple couplings suggests that there could be variability in the way the function is carried out and that this variability may affect downstream functions. Note that this is based on a simple visual inspection. Only a more detailed analysis of the couplings will show whether there can be actual variability. But looking for functions with multiple couplings can be a first step in determining whether functional resonance could occur. If so, further steps need to be taken to assess the likelihood and magnitude of the variability, as well as how it may affect other (downstream) functions.

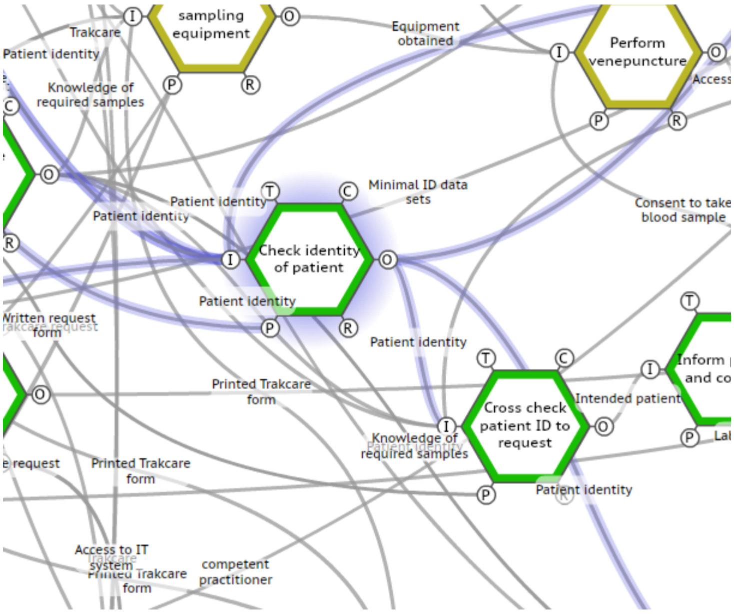


Figure 6: Functions with multiple upstream-downstream couplings

At the moment, all such analyses have to be done manually. There is some help to be found in the “record – playback” facility of the FMV, which is described in the FMV instructions. This facility makes it possible to create multiple instances of a model as you reason through them and to record them as the upstream-downstream couplings become active. Any such recoding can then be played back repeatedly as a means of presentation and/or as a possible support of extended discussions. (When using this option, it is a good idea to save each instance of the model and name it so that it is easy to retrieve.)

Summary

Here are a few important points to keep in mind when building a model using the FRAM and the FMV. The FRAM is a *method* that helps you to develop a model or a representation of how something happens or could happen. This model or representation can in turn be used to understand a phenomenon. The FRAM does not provide an analysis of what happens or could happen, nor an explanation of it and certainly not an explanation in terms of causes.

- **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 the six aspects.
- **Level of description**: There is no single, correct level of description. A FRAM model will typically include functions described on different levels.
- **Foreground - background**: Functions are pragmatically labelled as being either foreground or background functions. Background functions can be seen as constituting the boundaries of a model.
- **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 sub-functions.
- **System boundary – stop rule**: The analysis may go beyond the boundaries of the system as initially defined. (Indeed, most systems have boundaries that are based on how they are structured rather than on how they function.) The expansion of a FRAM model stops when it reaches a background function. If some background function can vary and thereby affect foreground functions "inside" the system, then it should be considered a foreground function.

Document history

A first version of this handbook was published by the Centre for Quality in September 2013. The handbook was written in Danish as support for the courses held by the Centre. The handbook contained extensive examples from health care.

It soon turned out that there was a demand for an English version of the handbook, which was prepared with the invaluable assistance of Jeanette Hounsgaard and Lacey Colligan. This version was published in June 2014 and is available for downloading (pdf-format) from www.centerforkvalitet.dk/framhandbook.

The English version has been used extensively in Denmark and other countries as a part of FRAM courses. While some courses focused on issues in the management of health care, others considered a broader range of industries. There has therefore for some years been a need for a version of the handbook that was in English but which did not focus specifically on health care. The current version of the handbook is an attempt to answer that need.

Appendix A: How to get the information needed to describe functions

The best sources of information about the activities being analysed are the people who actually carry them out. They can either be the people at the workplace under consideration, people who work in a similar workplace, or people who have had extensive experience with the work. Although interviews are the primary tool of investigation, other methods such as focus group discussions may be considered, just as the interviews may be supplemented by field observations.

Preparing for Interviews

Before the interviews it is important to think through the purpose of the study: how much information is needed and how will that information will be helpful? It is essential to prepare as as well as possible before going into to the field, for instance by consulting available information sources such as rules and regulations, documents, protocols, job descriptions, etc. Data on turnover of personnel, equipment, procedures and organisation and major events or changes to the function can also be valuable. This information will be the basis for the set of questions that should guide the interview. It is important to find out as much as possible about the physical and environmental conditions of the workplace. This may require examination of the architectural drawings, photos or videos etc.

The interview questions should focus on daily activities, established practices, and their characteristic variabilities. So, instead of asking about successes or failures, questions should focus on the daily routines and habits – things that might be taken for granted or passed over – or even suppressed if the focus was an adverse event.

Examples of Possible Questions

The following questions may by their content and form give some ideas about how an interview could be conducted. The questions are, however, meant as a source of inspiration rather than as a checklist.

- When do you start this activity? What ‘signals’ that you can begin?
- How do you adjust the activity to different conditions? How do you determine how and when to adjust?
- How do you respond if something unexpected happens? For example, an interruption, a pause required by a more urgent task that takes priority, a missing resource, missing or surprising information, etc.
- How stable is staffing? Is staff allocation permanently assigned or adjusted daily? What happens if staffing is short?
- How stable is the environment? Supplies? Resources? Demands?
- Are there often undesirable conditions that you have to tolerate or get used to?
- How do you prepare for your work (documents, instructions, colleagues, etc.)? What do you do if these resources are not available?
- What preconditions and/or other factors are normally taken for granted?
- What information do you need (equipment, services, etc.)? What do you do if this is not available?
- How does time pressure affect your work?
- What skills and competence do you need? Does everyone performing this work have the required skills and competence? What happens – how do people compensate – when that is not the case?

- What is the optimal way to perform this work? *Is* there an optimal way?
- How often do you have to make changes or adjustments to the ways you work?

The Interview

The interviews should if possible take place at the actual place of work. A tour of the workplace is often useful to get a feeling for the local environment. The interviewer may bring a valuable set of 'new' eyes to things that workers may have become 'blind' to. It is also important to prepare the interviewees for the process. They must first and foremost agree to participate in the interview and understand the purpose of the interview.

It can be useful if two interviewers conduct the interviews together: one can then concentrate on the dialogue, and the other on taking notes. One of the interviewers may well be recruited from the work domain, but they must be aware of and try control their own biases and preconceptions. It can be helpful to record the interview if the interviewees explicitly agree.

How to document the Interview

The first step is to type or transcribe the notes from the interview and consider these together with information previously obtained during the preparations. The team needs to identify the important functions and arrange the material so that the information is sorted by functions. If possible, some foreground and background functions can be identified already at this stage.

For each foreground function, one should try to identify as many of the six aspects as reasonable. Information about Input and Output represent the bare minimum required.

The Output, with its expected variability – potential and actual – should be characterised in detail with respect to *time* and *precision*. For time, one must determine whether the Output varies by coming too early, too late, on time, or not at all. For precision, one must determine if the Output is likely to be imprecise, acceptable, or precise.

For each function, you should provide the following:

- **Function Name**: it is important to find a short and clear name that describes the activity. This should be written as a verb or verb phrase.
- **Description of the function**: try to describe the function in as much details as necessary (free text description). The description should include who performs the function – not necessarily a specific individual but the organisational role. This description can be as long or as short as you wish. It is also useful to enter any points that came up during either the data collection or the discussions in the analysis team, lest they should be forgotten
- **Characterisation of some or all of the six aspects**. Each relevant aspect should be described to the extent possible with the information available and to the extent necessary to best describe the function. Remember that a function can have more than one Input, Output, Precondition, etc.

The description of the functions and aspects should be made by means of the FMV. This is an uncomplicated but powerful software tool that helps structure the information and also provides some useful functions to check the completeness of the model.