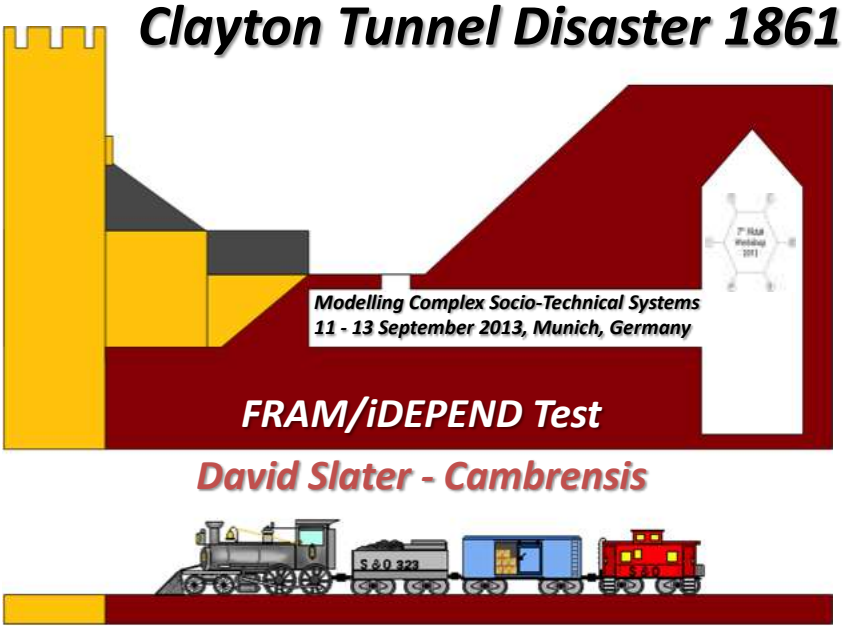
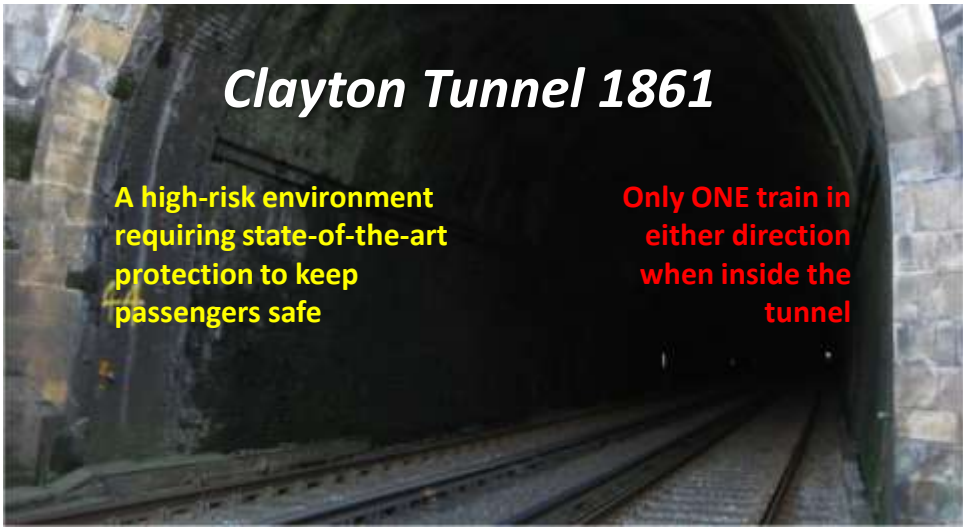
Clayton Tunnel Disaster 1861

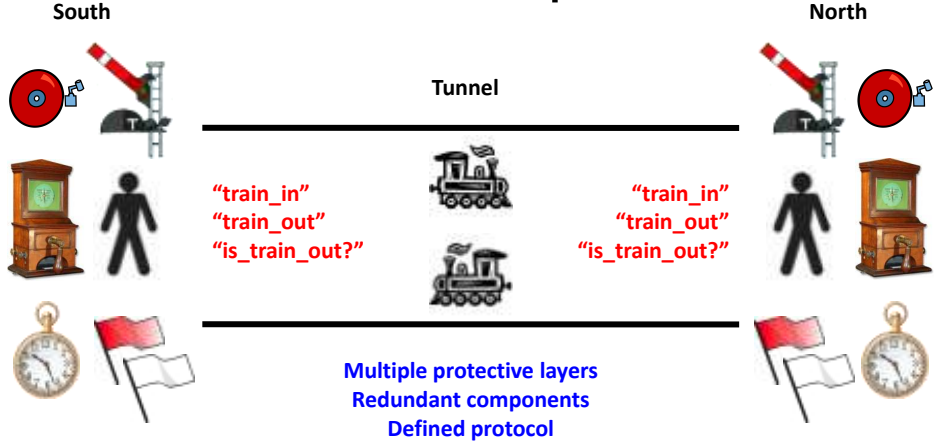


David Slater - Cambrensis

A "Modelling" Exercise



State-of-the-art protection



Multiple protective layers
Redundant components
Defined protocol
State-of-the-art technology
What could possibly go wrong?

Telegraphic protocol

- The-needle telegraph allows three signals:
  - "train_in"
  - "train_out"
  - ("is_train_out?")
- Process:
  - train passes green signal
  - train enters tunnel
  - signal trips to red
  - signalman A telegraphs "train_in"
  - train traverses tunnel…
  - …train exits tunnel
  - signalman B telegraphs "train_out"
  - signalman A resets signal to green

Brighton station, 25 August 1861, 08:28

Assistant Station Master Legg

Driver Gregory

Brighton Parliamentary

08:35 (08:30)

Driver Scott

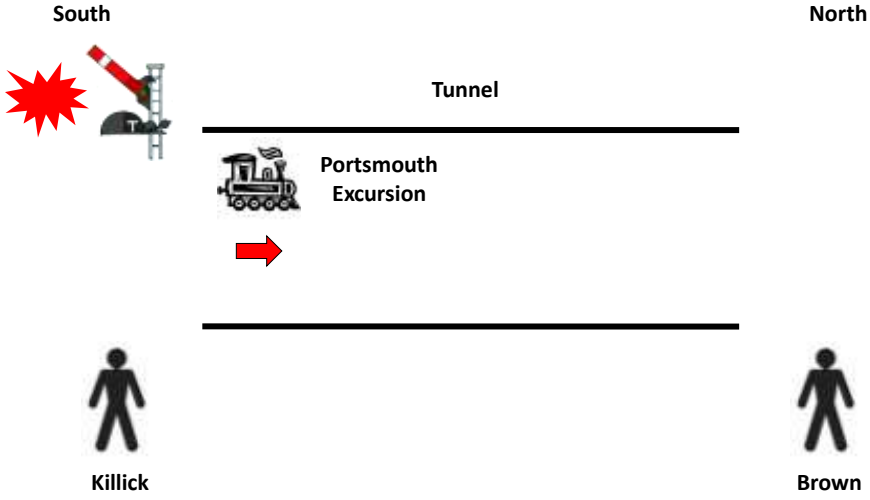
Brighton Excursion

08:31 (08:15)

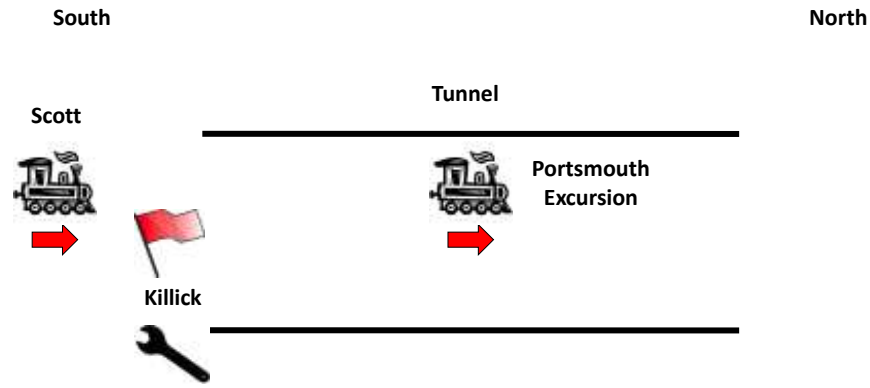
Portsmouth Excursion

08:28 (08:05)

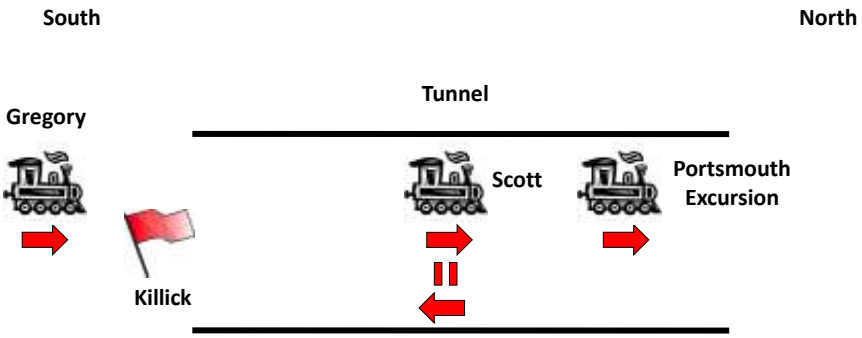
Disaster strikes



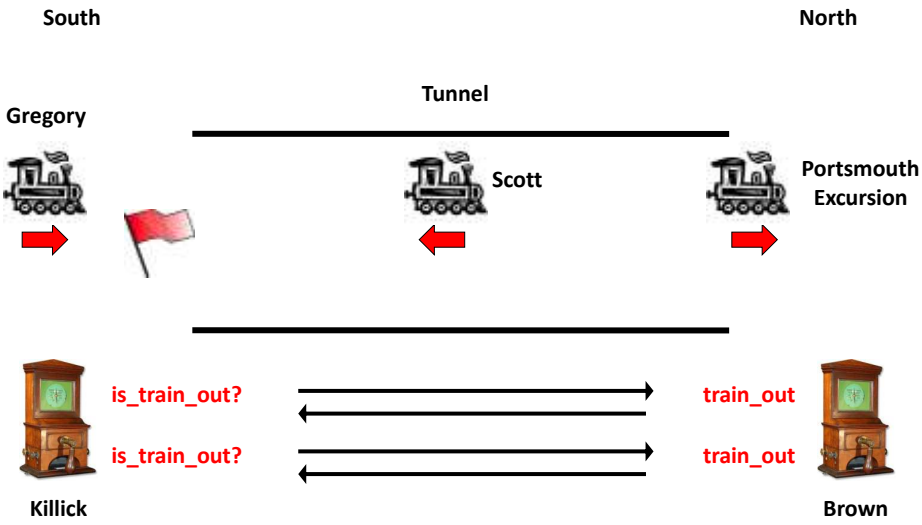
Disaster strikes



Disaster strikes

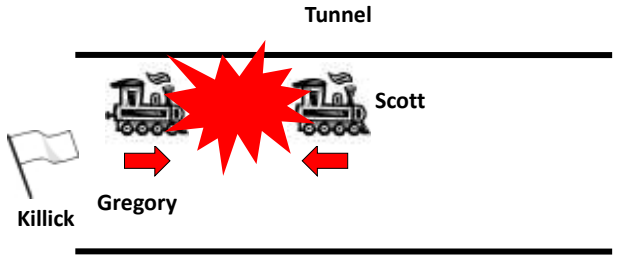


Disaster strikes



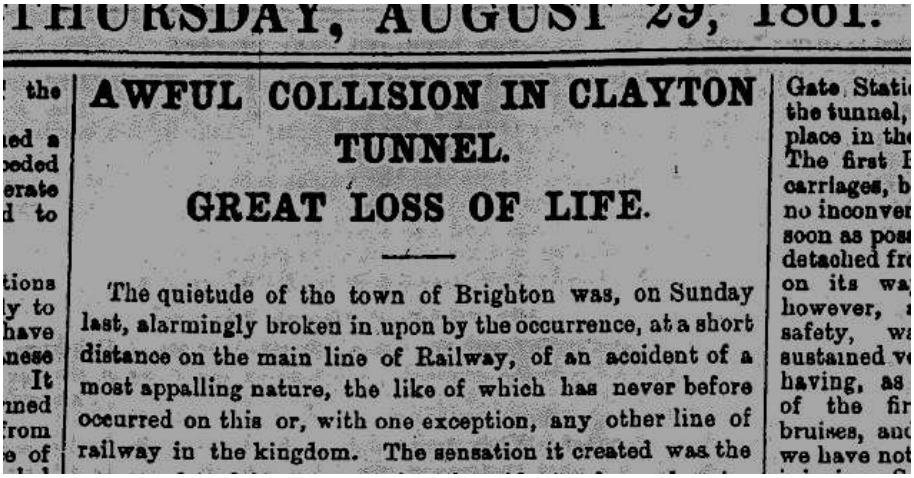
Disaster strikes

South North



23 killed, 176 seriously injured

23 killed, 176 seriously injured



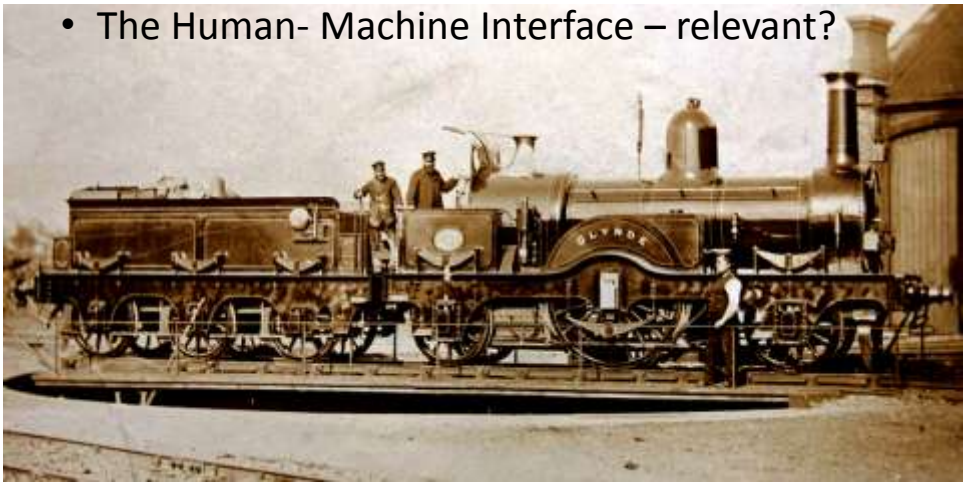
THURSDAY, AUGUST 29, 1861.

AWFUL COLLISION IN CLAYTON TUNNEL.

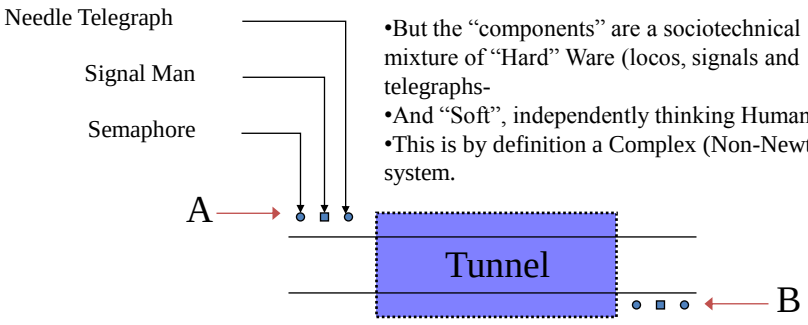
GREAT LOSS OF LIFE.

The quietude of the town of Brighton was, on Sunday last, alarmingly broken in upon by the occurrence, at a short distance on the main line of Railway, of an accident of a most appalling nature, the like of which has never before occurred on this or, with one exception, any other line of railway in the kingdom. The sensation it created was the

A London to Brighton Passenger Loco circa 1865



Clayton Tunnel as a Control Loop?



•But the "components" are a sociotechnical mixture of "Hard" Ware (locos, signals and telegraphs-
•And "Soft", independently thinking Humans.
•This is by definition a Complex (Non-Newtonian) system.

Signals
In! Train in tunnel.
Clear! Tunnel is free.
Ok? Has the train left the tunnel?

•We can't treat it as a simple equipment reliability risk analysis (Fault/Event Trees etc.)
•Or even as a Computer Protocol?

Gerard Holtzman
Design and Validation of Computer Protocols

The FRAM Methodology?

- If it was a computer protocol?
- Structured Analysis and Design Technique (SADT) looks at software as an "Orchestra" of interacting "Functions" for playing "scores" of music
- These exchange inputs and outputs with a range of other "Functions"
- Erik Hollnagel has proposed to utilise and extend this approach to systems in general.
- They are envisaged as a collection of these "Functional" units , **FRAM's**
- (Hexagonal Functional Resonance Accident Modules)

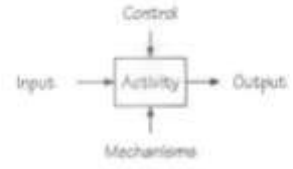


Figure 5.3 An SADT activity

Figure 5.4 A hexagon representing a function

"Human" Variability of Inputs and Outputs

- In the real world these required interchanges will almost always not be exactly what is required
- The success of the system is the how well it can cope with these necessary adjustments (Resilience?)
- But if the variability is outside the range it can cope with we may have a problem
- The analysis systematically identifies possible deviations and predicted resulting system behaviours, rather like a HAZOP
- It also looks at unexpected outcomes from unintended resonance between these individual deviations.
- This is the Functional Resonance Analysis Method - FRAM

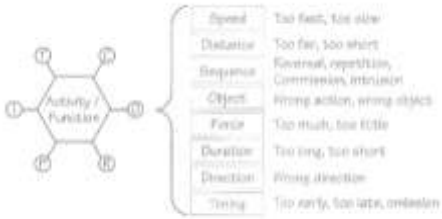


Figure 6.1 Dimensions of failure modes

FRAM Template

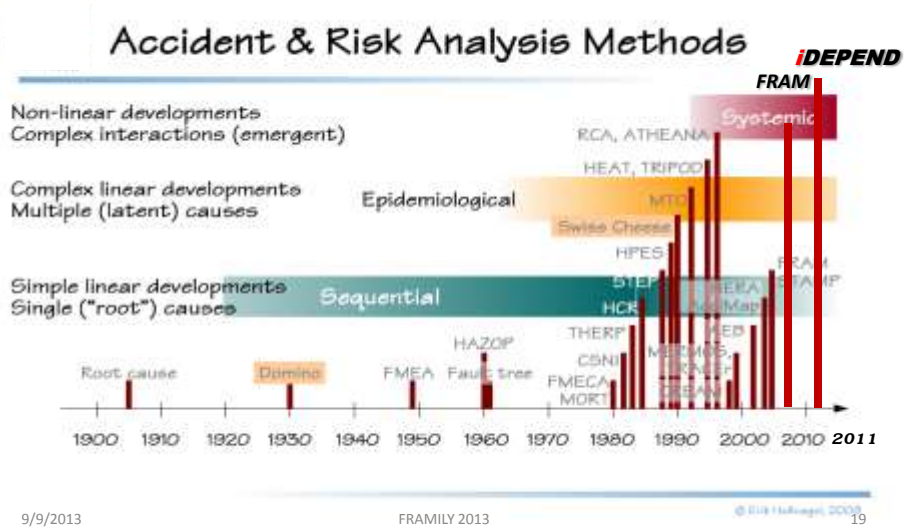
FUNCTION NAME	
FUNCTION DESCRIPTION	
INPUT (That which activates the function and/or is used or transformed to produce the output. **Constitutes t**he link to upstream functions.)	
OUTPUT (That which is the result of the function. Constitutes the links to downstream functions.)	
PRE-CONDITIONS (System conditions that must be fulfilled before a function can be carried out.)	
RESOURCES / EXECUTION CONDITIONS (That which is needed or consumed by the function when it is active (matter, energy, competence, software, manpower)	
CONTROL (That which supervises or regulates the function. E.g., plans, procedures, guidelines or other functions.)	
TIME (Temporal aspects that affect how the function is carried out (constraint, resource).	

Background function			
Foreground function			
	Timing	Precision	Elaborated
Potential variability			
Actual variability			
	Human	Tech.	Org.
Function type			

So why isn't everybody using FRAM?

- **Resource intensive!**
- **Unfortunately resources only tend to be made available in response to problems or high profile accidents – "Black Swans"**
- **It currently gives qualitative insights as "static" snap shots in time.**
- **But these insights are valuable for both Design and Incident Analysis**
- **So what if it was easier to build and use?**
- **Could we produce quantitative outputs? (Risk Numbers?)**
- **Could we make it Dynamic and interactively visual?**
- **Make and break the functional links automatically and reassign/change them at will?**
- **Use real (time) inputs to the FRAM's?**
- **Can we look at handling variability systematically, quantitatively?**
- **Can we retain the record/ Models- without having to start from scratch every time.**
- **We could then use and reuse them more routinely as management tools, predictively – before the accident?**
- **i.e. Checking out the "White Swans" as well– they're much more common?**

The FRAM methodology has evolved from an increasing awareness of "Complexity" issues (after Hollnagel)

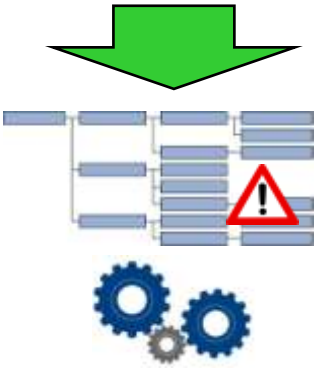


What is Dependency Modelling?

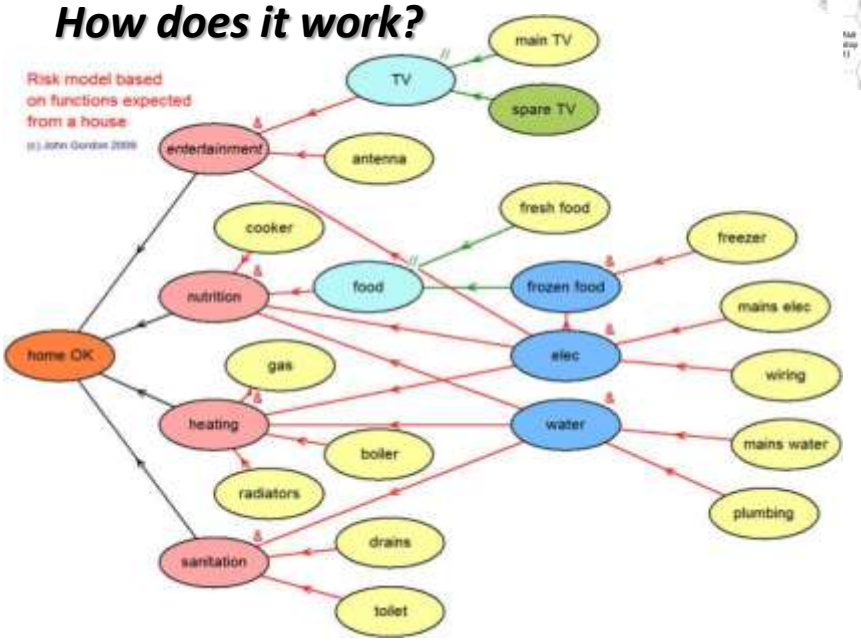
- Intuitive , goal-driven approach
- It starts by clearly defining our Objectives (in a logical and consistent way)
- Then "Maps" the critical dependencies onto those Objectives
- Builds a pragmatic, hierarchical model of our actual working environment
- Smarter, more focussed and "just enough"
- A Living model rather than static shelf ware
- (Now an Open Group Standard)

Dependency modelling

"What do we need to succeed?"

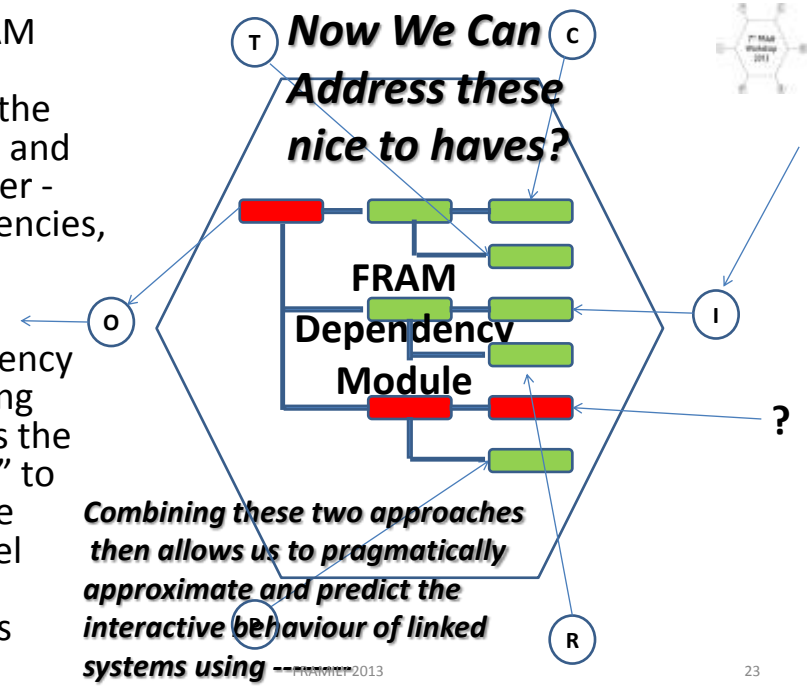


How does it work?



The FRAM analysis defines the Systems and their inter - dependencies, and

Dependency Modelling provides the "Engine" to drive the Hollnagel FRAM Modules



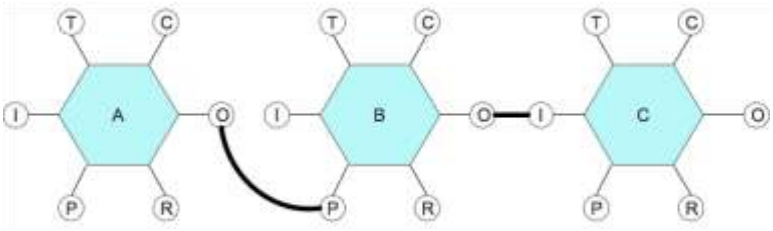
Combining these two approaches then allows us to pragmatically approximate and predict the interactive behaviour of linked systems using ---------

Common Fractal Structures

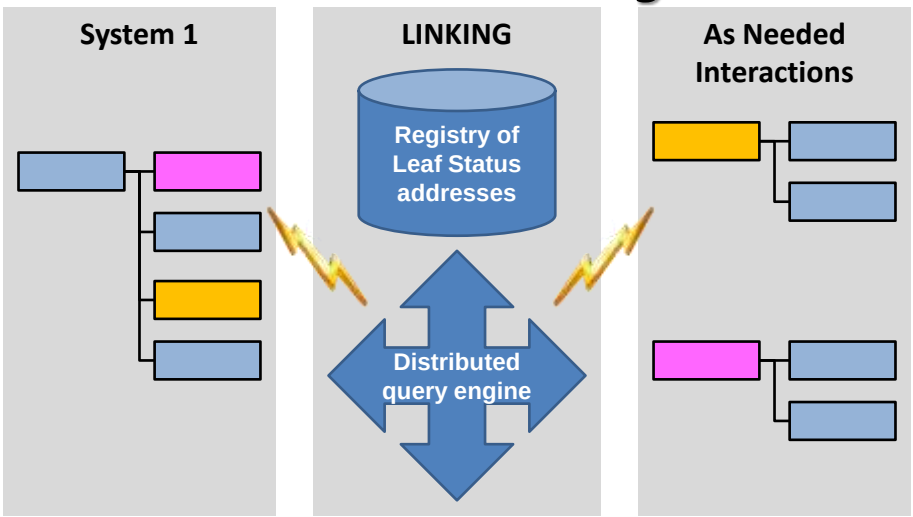
- In essence the **FRAM** notation provides a consistent structure (template) for that system's **Dependency Models**.
- But the fractal development of dependency models is mirrored in **"Functions within Functions"!** (Russian Dolls!)
- Each **FRAM** is a Function (and is in turn made up of functions?).
- (It need not be 6 "receptors" for example there could be multiple inputs, preconditions, resource requirements , etc.)
- If you label the entities (dependencies) correctly, they don't care which level they're on (or which Doll?), they'll link up automatically.
- As will Geofences, Crowdsourcing, etc.
- **Time?** –also includes correct **SEQUENCE**.
- This can now be an automatic update –time step, time out – elapsed time etc)

How do they Link up?

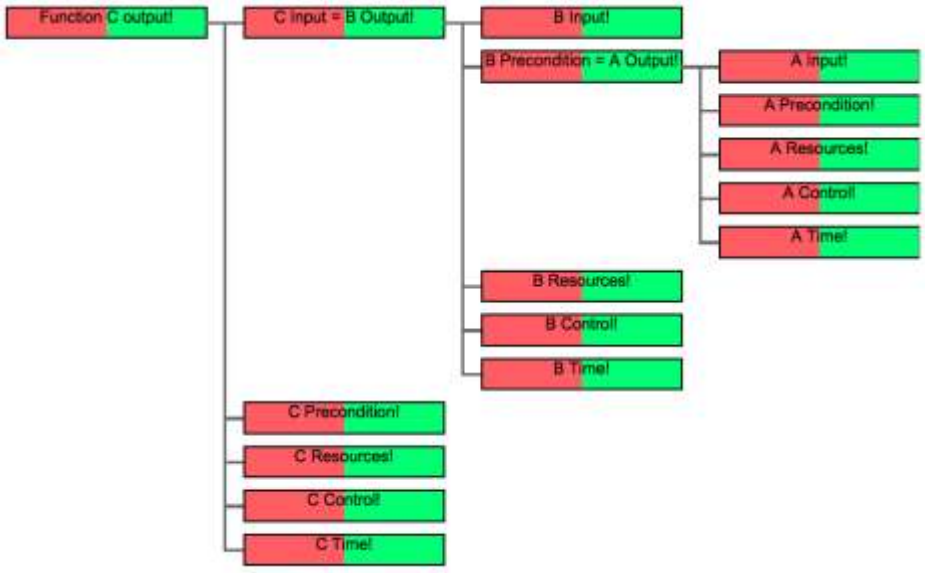
- Say we have three (FRAM) functions, A, B and C.
- The **Output** of A feeds into B as a **Precondition**
- and the **Output** of B feeds into the **Input** of C.



The iDEPEND "Engine"



This is then how the dependencies link automatically!



FUNCTION NAME	Station Master Legg
FUNCTION DESCRIPTION	Release Trains
INPUT (That which activates the function and/or is used or transformed to produce the output. **Constitutes t**he link to upstream functions.)	Timetable Prompt
OUTPUT (That which is the result of the function. Constitutes the links to downstream functions.)	Release Train
RE-CONDITIONS (System conditions that must be fulfilled before a function can be carried out.)	All Clear from Signal Men
RESOURCES / EXECUTION CONDITIONS (That which is needed or consumed by the function when it is active (matter, energy, competence, software, manpower)	Competence
CONTROL (That which supervises or regulates the function. E.g., plans, procedures, guidelines or other functions.)	Region?
TIME (Temporal aspects that affect how the function is carried out (constraint, resource).	Timetable

So let's try it!?

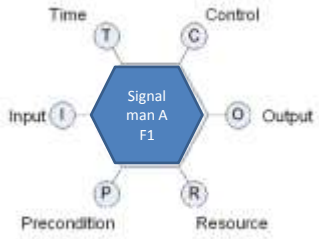
- **First define the functions**
- **Take the initiating Function "Train Release"**
- **complete the FRAM template**

Background function			
Foreground function			
	Timing	Precision	Elaborated
Potential variability			
Actual variability			

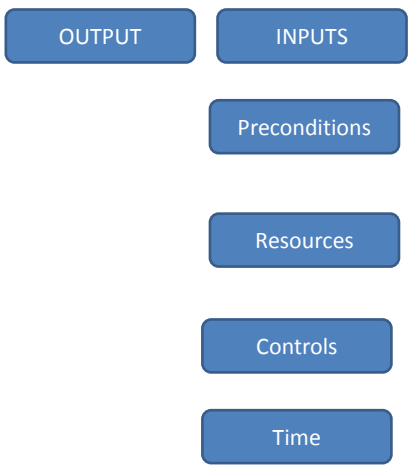
FUNCTION NAME	Signal Man A Function 1
FUNCTION DESCRIPTION	Control Access to Tunnel
INPUT (That which activates the function and/or is used or transformed to produce the output. **Constitutes t**he link to upstream functions.)	Train Out
OUTPUT (That which is the result of the function. Constitutes the links to downstream functions.)	Set Signal/ Instruct Flagman to Green/white
RE-CONDITIONS (System conditions that must be fulfilled before a function can be carried out.)	Communications working and message repeated
RESOURCES / EXECUTION CONDITIONS (That which is needed or consumed by the function when it is active (matter, energy, competence, software, manpower)	Working Signal or available Flagman Telegraph
CONTROL (That which supervises or regulates the function. E.g., plans, procedures, guidelines or other functions.)	Station Master
TIME (Temporal aspects that affect how the function is carried out (constraint, resource).	Timetable

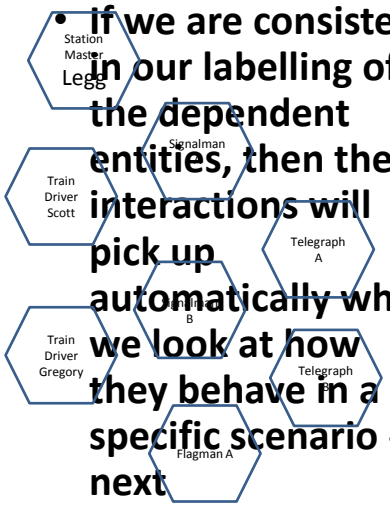
Background function
Foreground function

And the first control function



Now model each of these FRAM modules in an iDEPEND sequence

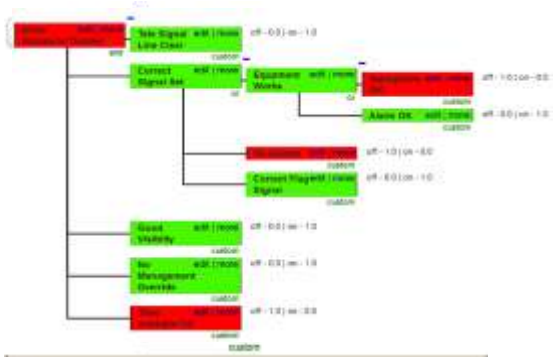




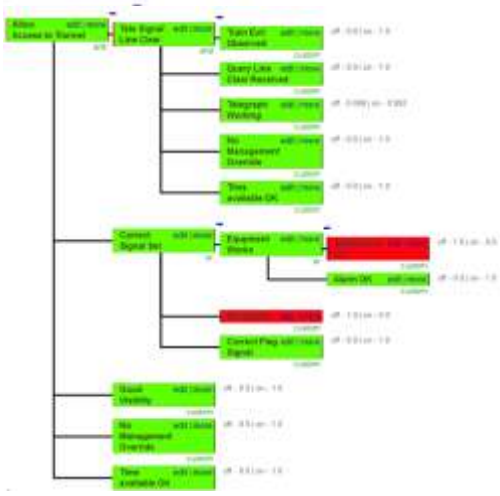
- **if we are consistent in our labelling of the dependent entities, then the interactions will pick up automatically when we look at how they behave in a specific scenario - next**

Signalman A (Killick).

- This function controls access to the Tunnel by communicating by signals, the state of the line ahead as confirmed by signalman B using the telegraph. An attempt to model this function on the FRAM template might look like this:-



System with two models, "Signalmen A and B".



If we set the time available to sufficient for both signalmen to perform their functions, , the system works as designed and access is successfully controlled, in spite of a semaphore signal failure.

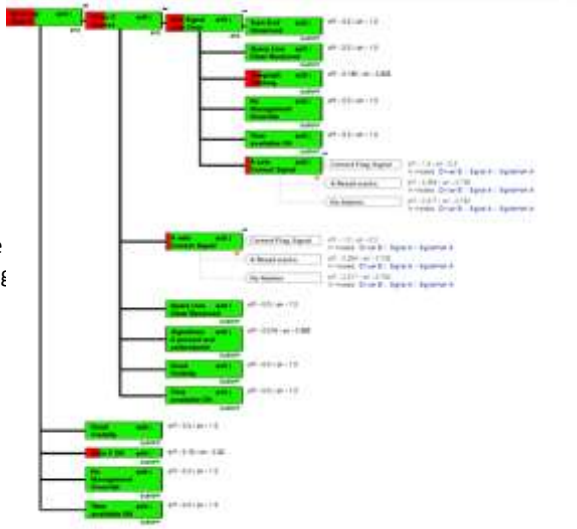
Model all the "actors" and see how they interact

- The system now includes:-
- Dispatcher
- Drivers A, B and C
- Signalmen A and B
- Telegraphs A and B
- Alarm A

The effect of the telegraph failure on the ability of Driver A executing his function to advance safely.

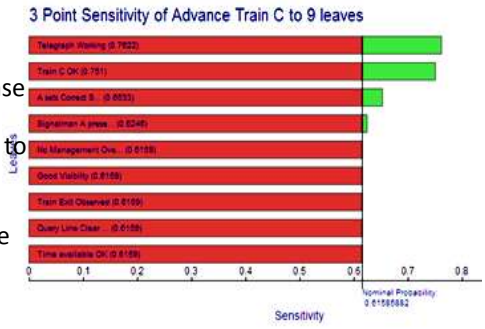
The effect of Killick (Signalman A) showing the wrong Flag signal.

But the tool allows us to also so apply it in predictive mode and assign probabilities to the various dependencies.



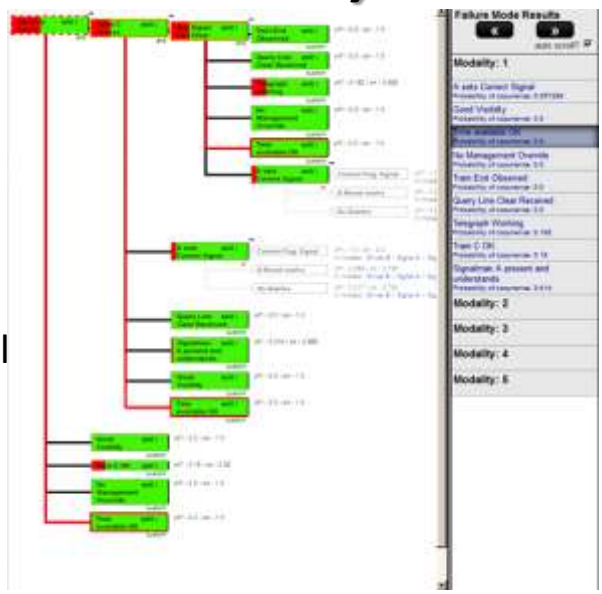
Running a Sensitivity Analysis

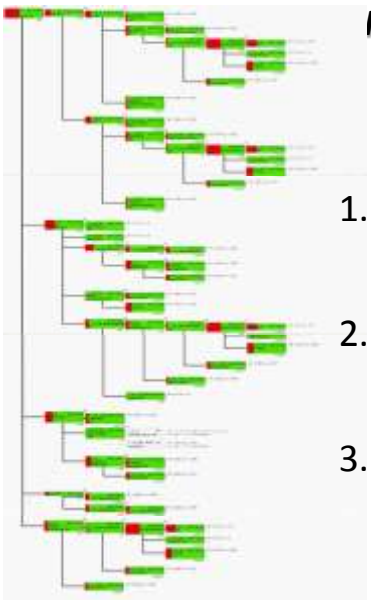
- The tool produces this 3 point sensitivity plot
- This indicates a nominal probability of successful operation of some 61%.
- The green bar indicates that if we increase the reliability of the Telegraph to 100%,
- we can increase the overall success rate to about 80% (Pareto) at best.
- Of more concern are the red bars,
- on this model failure of any of 9 separate leaves can cause complete failure of the system.
- Most of these are "Human",
- This illustrates well that assessing a system performance on purely mechanical reliability is at least incomplete?



The failure mode analysis

- This indicates more clearly that “Time” is important.
- Its involved in multiple critical dependencies, as shown.





An iDEPEND Model of the Interacting Systems would look like this – AND -

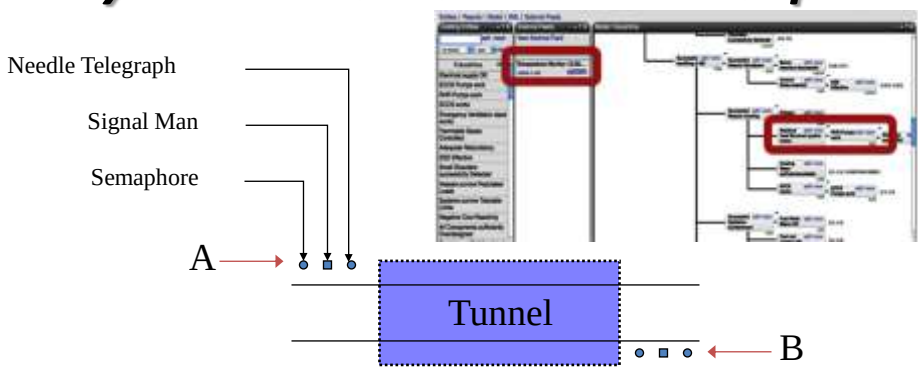
WE can now do three types of Quantitative Analysis

1. What if? Vary the inputs ad look at the way the predicted behaviour changes, or
2. Hook it up to real external feeds and monitor/ Manage behaviour, or
3. Run Monte Carlo simulations with different input probability distributions and look for the Resonances – (Finally - ***Black Swans***!)

But that Depends on ---?

- What do I do about "Mains" electricity?
- Water Supply?
- Supermarket Stocks?
- The Weather?
- These are what we call "Leaf" dependencies
- So Monitor it, measure it and input it
- Requires the Open Group, accepted Data Transfer Protocol to ensure secure availability and connectivity.

Clayton Tunnel as a Control Loop?



Signals

In! Train in tunnel.

Clear! Tunnel is free.

Ok? Has the train left the tunnel?

Gerard Holtzman
Design and Validation of Computer Protocols

Conclusions?

- **We can now build a complete "System" containing all these interacting, interdependent functions.**
- **To this we can add spatial and environmental inputs again as "External Feeds" - dependencies common to many of the functions.**
- **But the major attraction for the FRAM process is that now we have a permanent, accessible "model" to test interactions, variabilities, "Barriers", redesign "work arounds", or ensure responsible and aware Management of Change.**
- **This Test Case has illustrated the potential power and usefulness to be derived from the synergy possible from combining these two "Systemic" approaches and is surely worth pursuing further.**