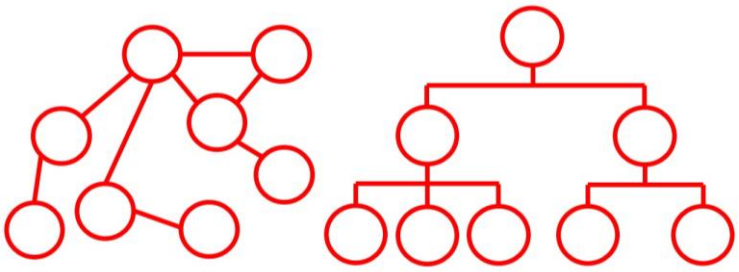
SYSTEM DIAGRAMS & NETWORKS

i.e. Pretty pictures that help

Systems engineers like pictures and generally they like pictures based on various types of network.

A network is a topography of nodes and interconnecting paths. In systems engineering the nodes represent elements and the paths interactions. We will be looking at

- Block Diagrams
- Flow Diagrams
- Family Trees
- Function Breakdowns
- Planning Networks



Hierarchical - has levels and the position of an element in the diagram matters
Non-Hierarchical - only the interconnections matter

BLOCK DIAGRAMS

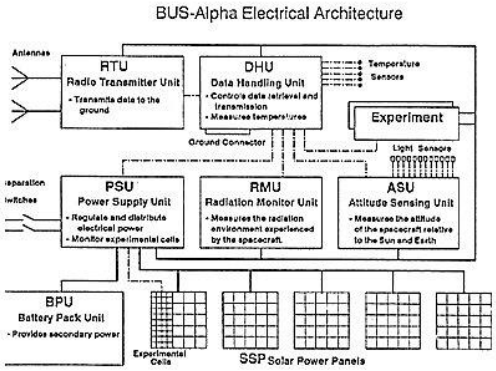
Block Diagrams simply show the elements of the system and the interconnections. They are not generally of much analytical use. Although preparing them to explain the system to other people often highlights a drop-off.

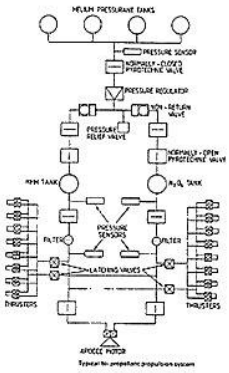
It is good practice to show the system boundary and system interfaces crossing it.

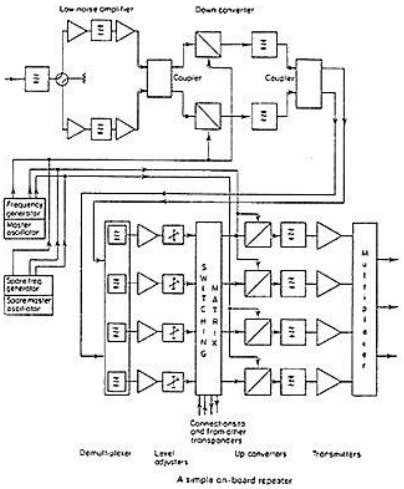
Basically these are used for illustrating clearly how the system has been divided into elements and what the main interconnects between the elements are. A lot easier and more digestible than a written description or an illustration of actual elements in their actual physical relationship.

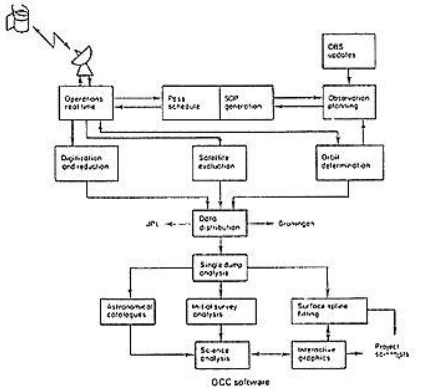
Block diagrams are static (i.e. no time dependence or flow component), and normally non-hierarchical.

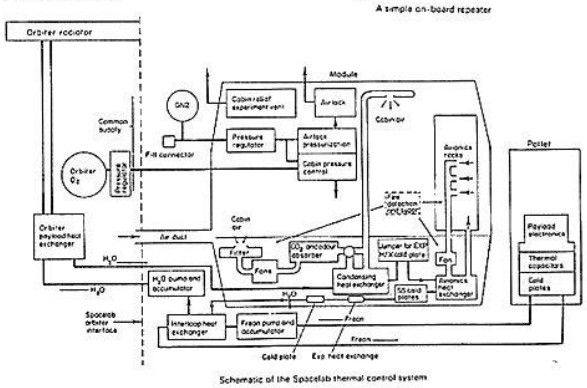
SPACECRAFT EXAMPLES OF BLOCK DIAGRAMS



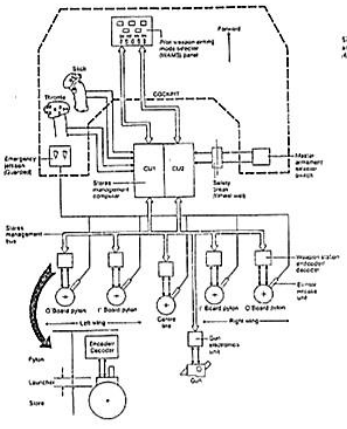


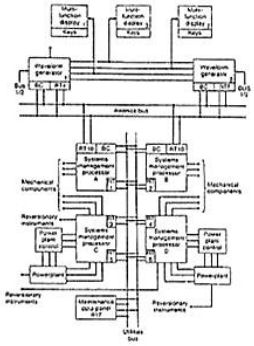


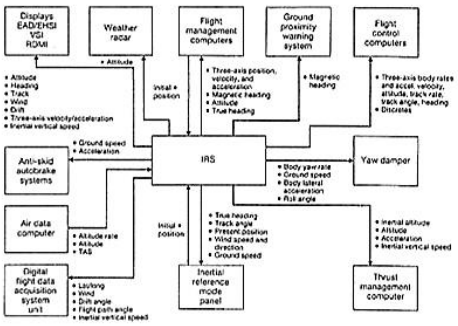


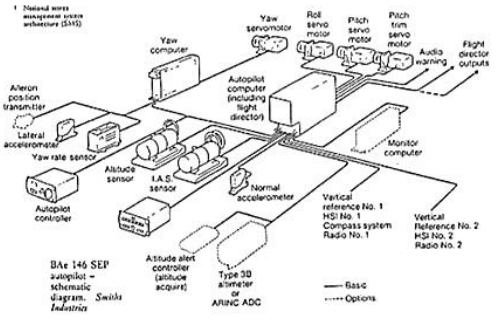


AIRCRAFT EXAMPLES OF BLOCK DIAGRAMS



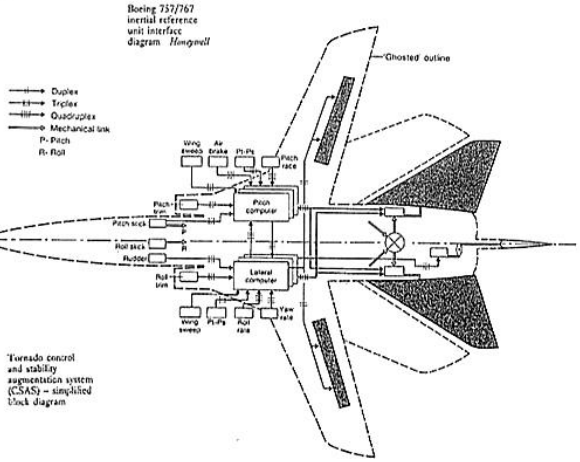






BAe 146 SEP autopilot – schematic diagram. *Smiths Industries*

Boeing 757/767 inertial reference unit interface diagram *Honeywell*



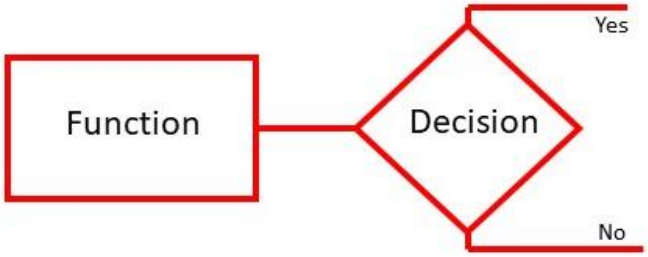
Tornado control and stability augmentation system (CSAS) – simplified block diagram

FLOW DIAGRAMS

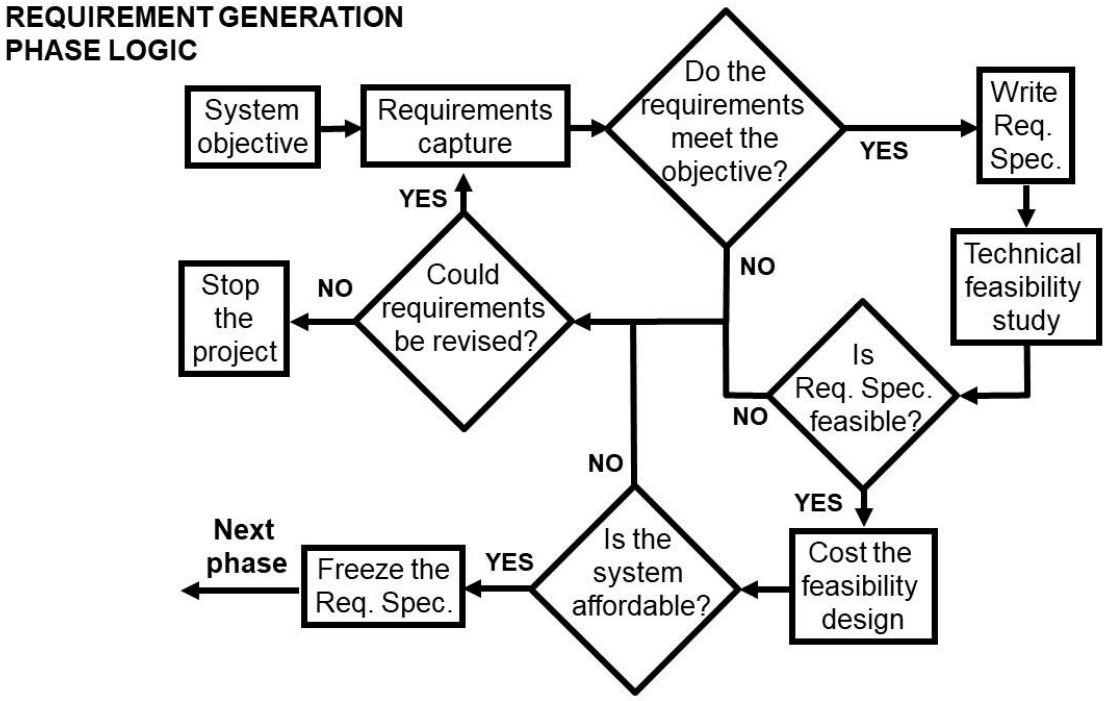
Functional flow diagrams are probably familiar to most people. They are much admired (but in practice not so often used) by the computer software industry.

These are networks where elements are functions that are time dependent and logically dependent upon each other in a serial manner - i.e. one follows after another - i.e. they flow.

Functions and processes (in rectangles) are placed in sequence and decision points (Diamonds) lead to branching. When used in computing decisions are normally limited to two options (as computers work in binary) for more general use more than two splits from a decision is possible.



Main uses is in systems where the functions are serial and this enables exploration of the logic of the system. This is why it works well on computer programs which can only do one process at a time. But it can also be used for work and process flows as in this example we have already used in an earlier lecture.

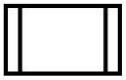


OTHER SHAPES

The ISO standards for flow diagrams also define other shapes such as

Terminal
(start and end)

Input or Output



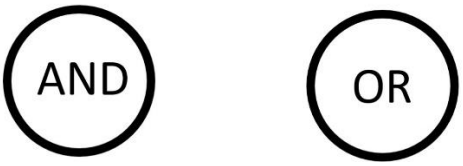
Pre-defined Function
or process

FLOW DIAGRAMS FOR PROCESSES

When using Flow diagrams for computer program design they should be confined to Functions and Decisions as these are all computers can do. When using flow diagrams for process analysis other functions can be incorporated.

Examples are AND and OR functions in circles. This introduces parallel processes into the diagram. Warning be careful parallel processes can frazzle your brain.

I do not like OR functions as they imply a decision without specifying the criteria to make that decision - stick to decision points in my view.

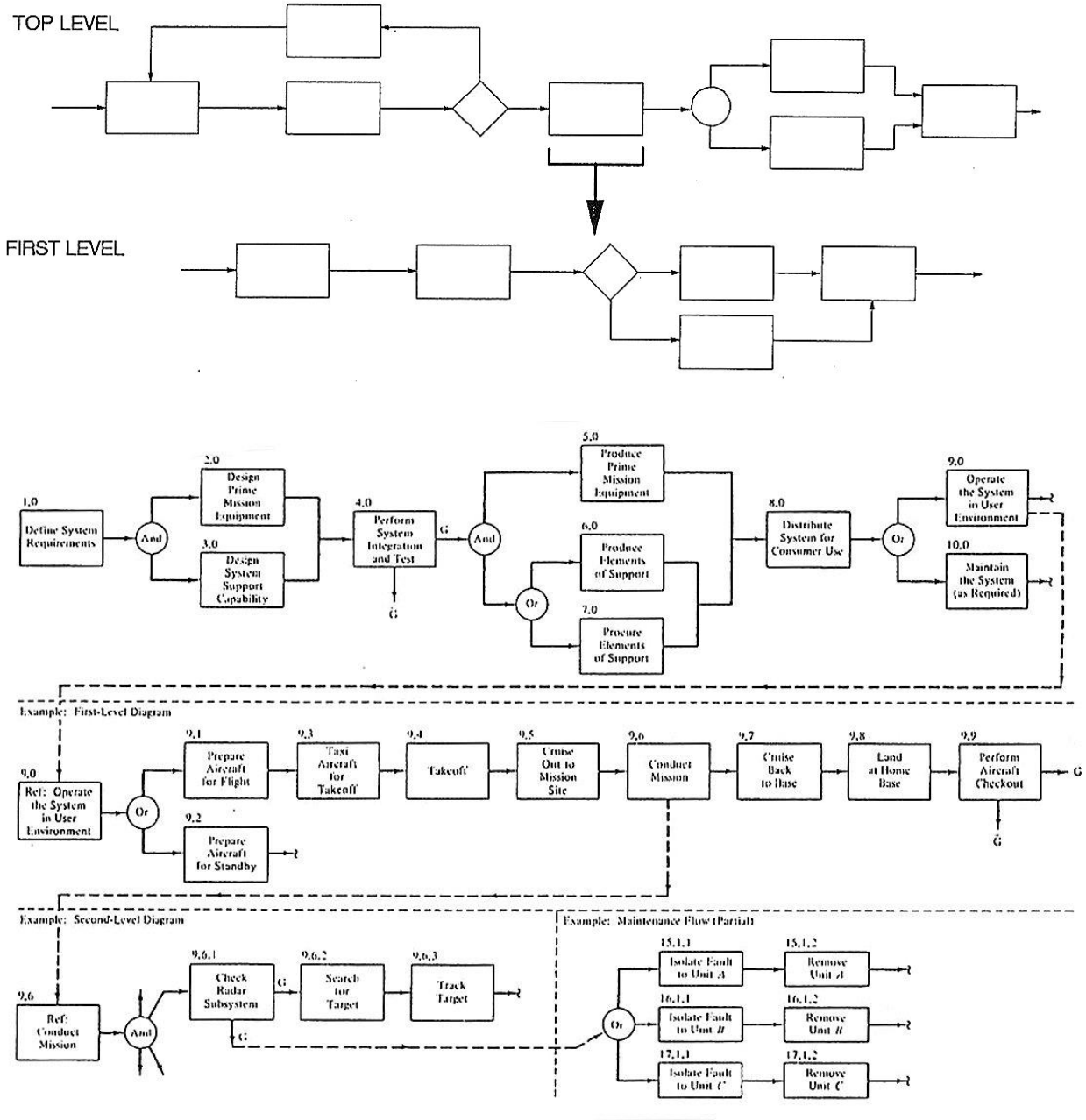


HIERARCHICAL FLOW DIAGRAMS

A flow diagram is normally non-hierarchical. However for complex functional flows, diagrams can be prepared at various levels.

A functional box in the top level can itself be represented as a flow diagram at the next level.

As only Functions can be split out, the lower level flow diagrams should have only one in and one out. All decisions must loop internally.



FAMILY TREES

A special type of network used for systems which have many levels. Used to illustrate where elements fit relative to each other in a hierarchy. These are of little analytical value but are useful for illustration purposes such as communicating organisational processes.

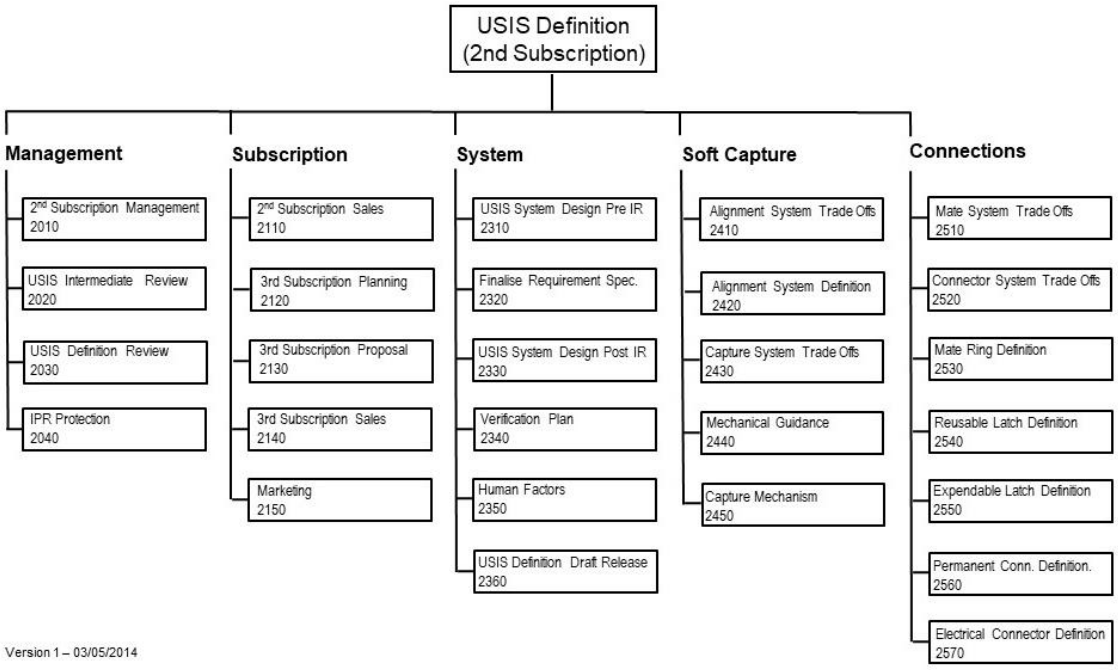
It is purely "Hierarchical" in structure connections only go up/down between levels not across to other functions on the same level. Hardly a surprise, as their main role is to illustrate

Often used to match various engineering activities with work-packages (activities that are managed and costed as an element in the management system) in something called a work-package breakdown

Typical breakdown levels might be:

- System,
- Discipline (mechanical, electrical, payload, habitability)
- Subsystem (structure, power, communications, navigation, control)
- Equipments or Units (Power regulator, Amplifier, GPS unit)
- Components (Nuts Bolts, resistors, circuit boards)

EXAMPLE Family Tree - A work package breakdown for a study programm



FUNCTION BREAKDOWN & REQUIREMENT BREAKDOWN

The Systems engineering top down approach means that requirements and functions are "unpacked" from the starting purpose. That is analysis - "Analysis" means decomposing into simpler elements, as opposed to "Synthesis" which means the combining of elements.

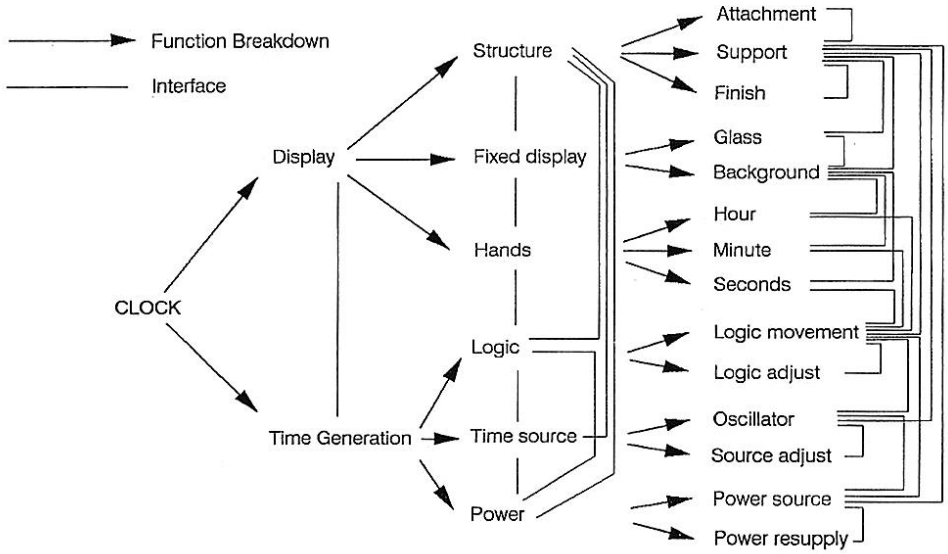
Much of this analysis activity is informal and involves common sense approaches but there is a formal technique of breaking down functions called (surprisingly) "Function Breakdown". Almost the same thing is the Requirement Breakdown, but in this case the goal is to be completely comprehensive, whereas a Function Breakdown can focus down on a particular area of interest.

This uses hierarchical networks and therefore has something in common with a family tree but there are some additional rules.

- The network should be comprehensive and deal with functional ore requirement elements
- Care should be taken that each level has logically equivalent elements (this can be difficult)
- To be successful "Mother" functions should have a maximum of 4 or 5 daughters, less is OK, but more can cause difficulties - computer based function breakdown packages normally limit you to 7 maximum.
- At each level interfaces are tracked between the elements. This greatly enhances the value of the diagram as it enables interfaces to be tracked down from the top to the lowest level.

FUNCTION BREAKDOWN OF A CLOCK

Showing Hierarchical Function breakdown and Interfaces



Note how complexity builds up with each level even on this simple system.

THE USE OF FUNCTION BREAKDOWN

Function breakdown is used in formal systems engineering analysis.

Where it is most successful is where the system elements are very similar in type and real life interactions are minimised - mostly computer software development. An early adopter was British Aerospace military aircraft who in the 1980s were using a computer based function breakdown tool for the detailed system design of avionic software.

In recent times it has become a more frequently used as a formal analysis tool for complex systems generally. The attraction being the computer ensures all requirements are tracked and the modern programs also write the requirement specifications for you.

Personally I am little sceptical on these as the effort need to make it comprehensive is so much larger than less formal, common sense, approaches. But I have seen them successfully used and they do ensure rigour. However I think you should not lose sight of their usefulness for informal thinking when trying to break the system down. They can also be used as illustrative material to show what you are doing.

Its gets very difficult on paper but can be handled by computer programs. Many commercial packages are available to do this job.

IBM DOORS (Dynamic Object Oriented Requirements System)

MICRO FOCUS Caliber

jama software Connect

pearls Pearls

bosseno ReqSuite

Accompa Accompa

MODERN Requirements Modern Requirements

ORCANOS Orcanos

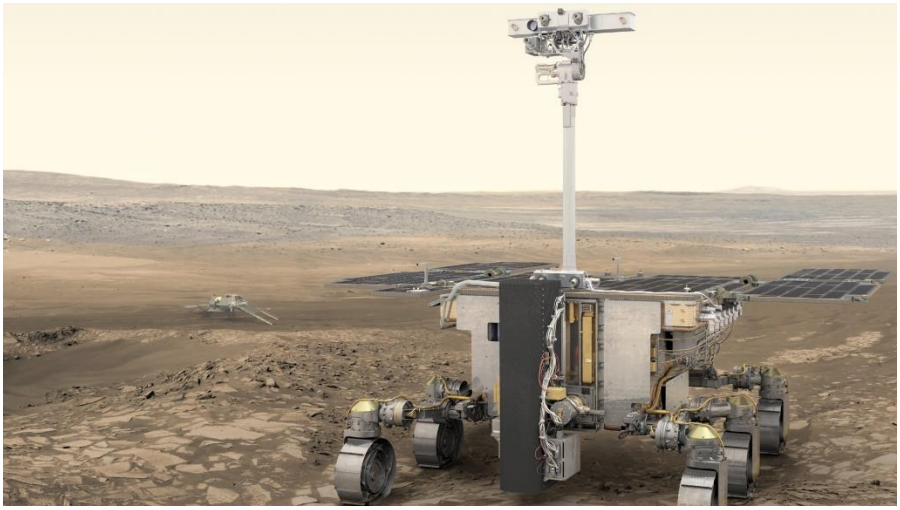
PERFORCE Helix RM

These programs do requirement tracking (function breakdown) and intended to automatically produce all the Requirement Specifications.

Requirements decomposition & management using DOORS*
(Dynamic Object Oriented Requirements System)

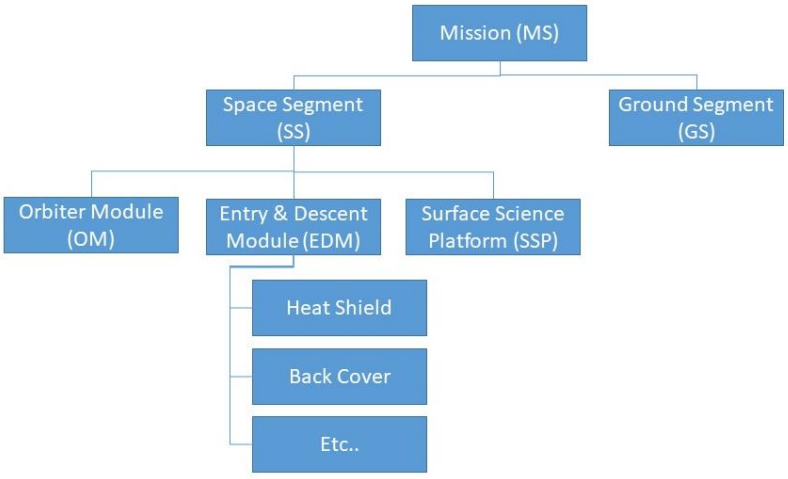
Some notes from Matt Darley

Matt works at Reaction Engines, but cannot show a real application of DOORS for commercial reason and so has kindly produced this generic example from his experience before Reaction Engines on Mars landers (such as ExoMars) which does not have commercial confidentiality constraints.



The ExoMars Rover – Rosalind Franklin

System decomposition



This is not from DOORS but note Matt uses a Hierarchical family tree to show the system breakdown so you can follow the DOORS output. He has produced a series of slides which I have put on the Blackboard site separately because of the detail in the screen shots they do not reproduce well in these notes.

TWO WARNINGS WHEN USING FUNCTION BREAKDOWNS

WARNING 1 - One trap to beware of is that some service functions (structure and power being the obvious) tend not to fall out early in the breakdown if at all. When they do fall out as functions as a support requirement there is an implication of isolation within a low level, for example, you could draw the conclusion that each electrical subsystem should have its own power supply. Thus missing that at a higher level you should have a common power generation and distribution system. You often have to force these utility functions in to the breakdown early to get the answer you want (but of course in doing that you are not following the analysis approach in its purest form).

A subtler version of this problem is less obvious system simplifications where a synthesis of functions into a single system which would be beneficial tend to be obscured by the analysis technique.

Examples of what might be missed are:

Common services - e.g. active cooling loop, power supply, data bus

Multiple functions - that is using the same component for more than one purposes e.g. using the structure as an electrical ground return.

WARNING 2 - One of the main problems with function breakdowns is they are good at internal interfaces but very bad at system interfaces that cross the system boundary.

A way round this is to add the outside world as a fake daughter function at the start and link system interfaces in as if it were another element.

Overall message

Function breakdowns do not work well if you are purest or apply the technique brainlessly

NETWORKS

The system model has the characteristic of a network; so where the system itself has a network characteristic, somewhat obviously the use of network diagrams can be useful in modelling the system and analysing its characteristics.

This is where the interconnections (of a single type) are of a distributed (as opposed to radiating from a central source) and the nodes act solely as a source/sink.

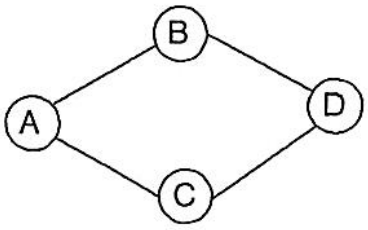
- Examples:
- Traffic systems
- Distributed Data Networks

Then network theory and mathematics can be applied.

(Covered in Aslaksen and Belcher - Chapter 3).

Perhaps the most common use of networks is for project planning. Project management can itself be treated as a system where the elements are the tasks to be performed.

NETWORKS CAN SHOW

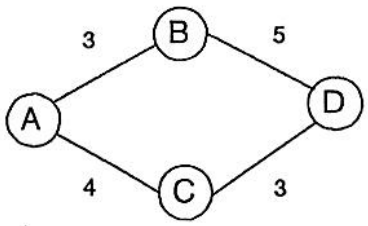


	A	B	C	D
A	0	1	1	0
B	1	0	0	1
C	1	0	0	1
D	0	1	1	0

CONNECTIVITY

e.g. Digital systems

Matrix representaion enables Boolean algebra to be used.

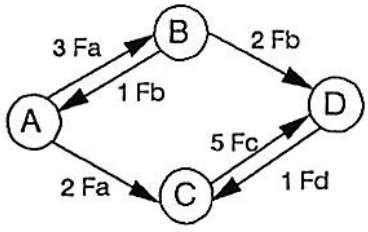


	A	B	C	D
A	0	3	4	0
B	3	0	0	5
C	4	0	0	3
D	0	5	3	0

NUMERCIAL VALUES

e.g Distances in transport system

Most common use is finding the shortest path.



	A	B	C	D	
A	0	1	0	0	Fa
B	3	0	0	0	Fb
C	2	0	0	1	Fc
D	0	2	5	0	Fd

FUNCTIONS

e.g Flows through a system

CRITICAL PATH & PERT NETWORKS

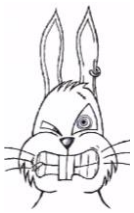
(Covered in Blanchard and Fabrycky Chapter 12)

Both techniques are used for project planning. Interconnections are the tasks required to complete the project.

Critical path analysis lays out a network of these tasks. Where a task needs a previous task to be done first, it originates at the node where that prior task is completed. Along each path between nodes the time required for the task is shown.

The longest route through the network from start to finish is the Critical Path; that is the route through the network where delays will affect the completion date.

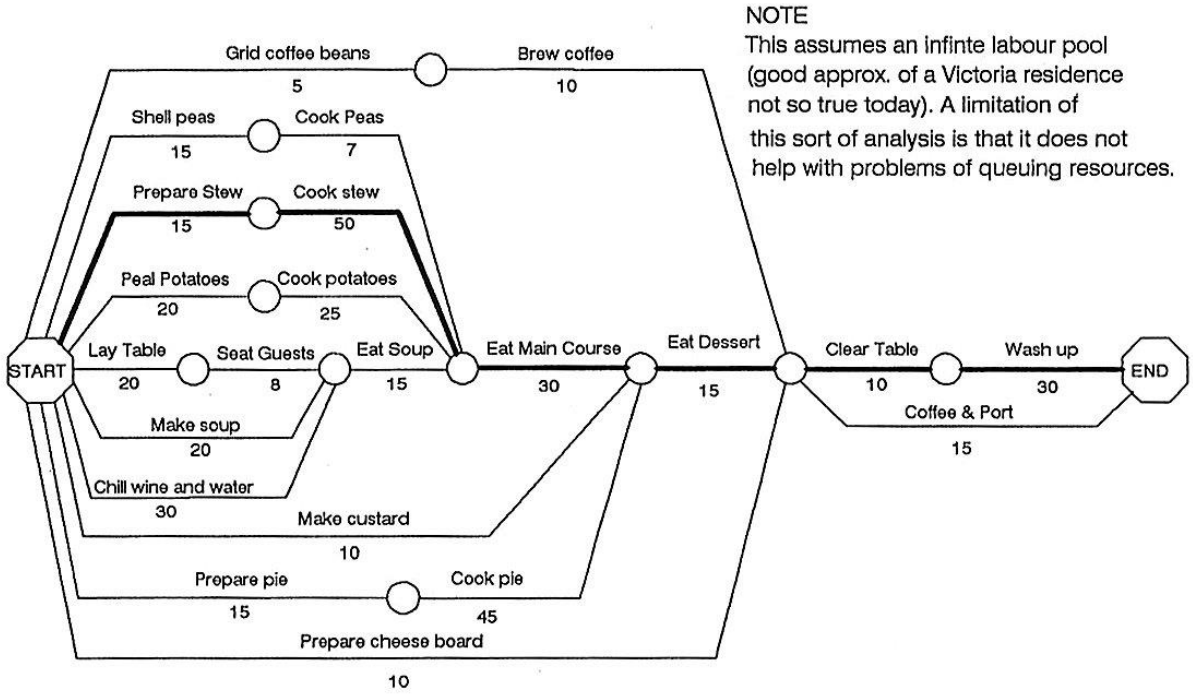
PERT (Program Evaluation and Review Technique) is similar to, and an extension of, to critical path analysis. The term is now frequently misused to describe normal critical path networks. Actually it a probabilistic variant of critical path analysis invented by Lockheed for the Polaris missile programme. Now often used in major project planning.



Rather than a simple time allocation; a range of possible times is placed along each path. This allows

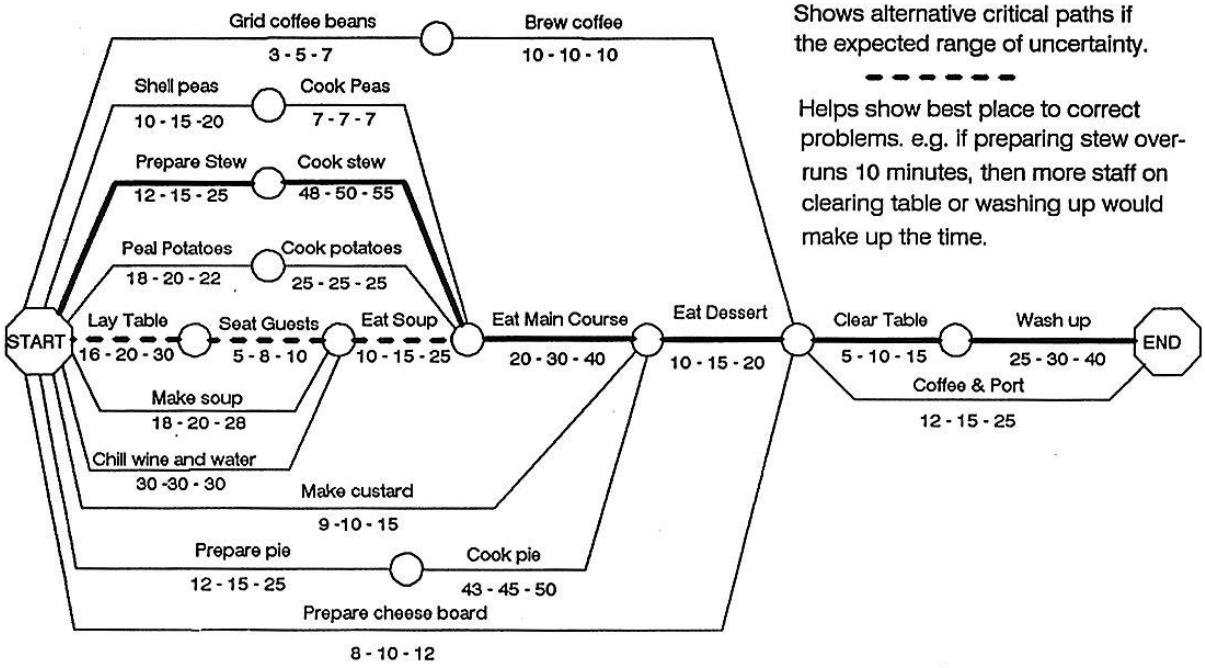
- analysis of uncertainty in project completion,
- identification of alternative possible critical paths with errors to be found, and
- finding best approaches to resolve scheduling problems as they arise.

CRITICAL PATH ANALYSIS OF DINNER PARTY



NOTE
This assumes an infinte labour pool (good approx. of a Victoria residence not so true today). A limitation of this sort of analysis is that it does not help with problems of queuing resources.

PERT ANALYSIS OF DINNER PARTY



NOTE
Shows alternative critical paths if the expected range of uncertainty.

Helps show best place to correct problems. e.g. if preparing stew over-runs 10 minutes, then more staff on clearing table or washing up would make up the time.