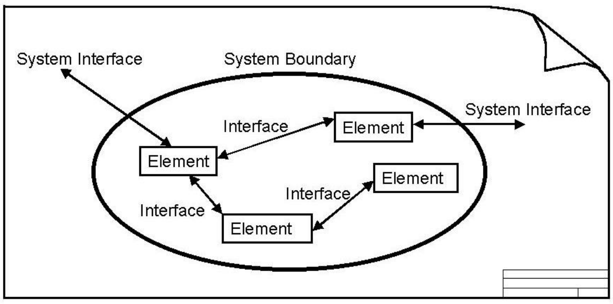
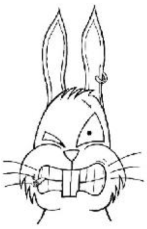
SYSTEM MODELS TO MODEL SYSTEMS

A "system model" is one that tries to capture some of the complexity of the system by being complex itself - and therefore can in itself be seen as a system. So a system model is more than simply "a model of a system", it is "a system to model a system". So it goes beyond a model of some aspect of the system (such as the mass model) and tries to capture some of the system's "emergent behaviour".

But it is still a model; that is it is still an "abstraction" of the system itself and would normally be less complex than what it is modelling.

And it is still a system; that is the model must have interconnected elements that create emerging properties. In short it fits the generic "System Model" we defined in the first lecture.

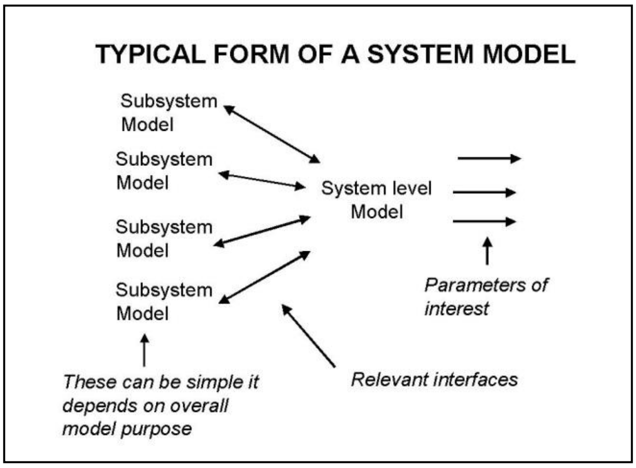




Warning
The term "System Model" is being used for two different things here

A System Model is an integrated collection of relationships that describe the behaviour of the individual sub-systems and hence the System as a whole to the various design inputs to achieve an overall design.

Parkinson



A DIFFICULT SUBJECT

The issues related to complex models of systems are not discussed in any of the text books I have seen – or any papers. It is not "literatured".

Even the text book by old boss (Parkinson) – who started the BAe Space Systems spreadsheet models - does not adequately discuss the basics of system models. (Although you will notice I have pinched stuff from it.)

I believe it to be an important area – hence "wasting" a lecture on it. However I do not have an underlying academic philosophy to impart (one day there will be one and it is emerging through Model Based System Engineering – see later) just examples showing the principles in action.

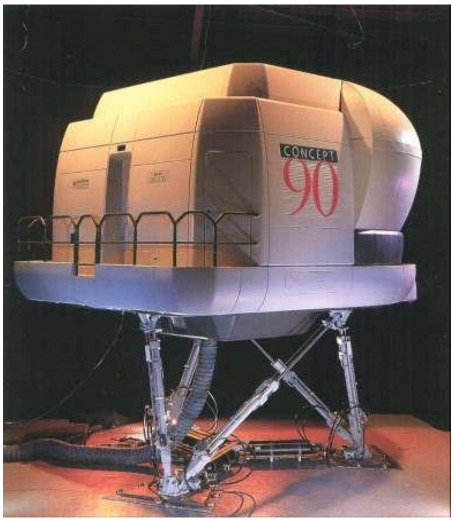
According to Parkinson an integrated System Model should capture:

· * The Mission Objectives as they affect the System Design
· * Identification of the Functionality of the System necessary to meet the Requirements
· * Characterization of External Interfaces, Environmental
* Constraints, Safety Factors, Margins etc.
· * Requirements for the component Sub-systems
· * Predictions of component Sub-system performances, and (by integration) the performance of the System itself
· * Relevant configurational issues
· * Where possible, an assessment of the System Cost and the origins of those costs.

WHAT ARE THEY USED FOR?

Parkinson view of the objectives is rather orientated to the role in preliminary system design – in fact there are three roles for system models.

1 – Early Systems Design Optimisation
Impact of options and variables can be rapidly assessed moving towards a more optimised design.



2 Change Control
Evaluation late changes during the detailed design and build phases.

2 - Training Simulation
In order to simulate the complete range of behaviour of a complex system the model must be complex.

(in the future we can expect an expansion of the roles system models can perform)

EXAMPLE 1 - BAE SPACE SYSTEMS SPREADSHEET MODELS

SYSTEM MODELS FOR SATELLITE DESIGN

BAe Space Division - (later Astrium) has used Excel Spreadsheet based system models to maintain control of configuration during preliminary design.

Each subsystem has a page in the spreadsheet where interfaces are either calculated using parametrics and other subsystem design tools, or are directly inputted. The subsystem lead engineer controls this sheet.

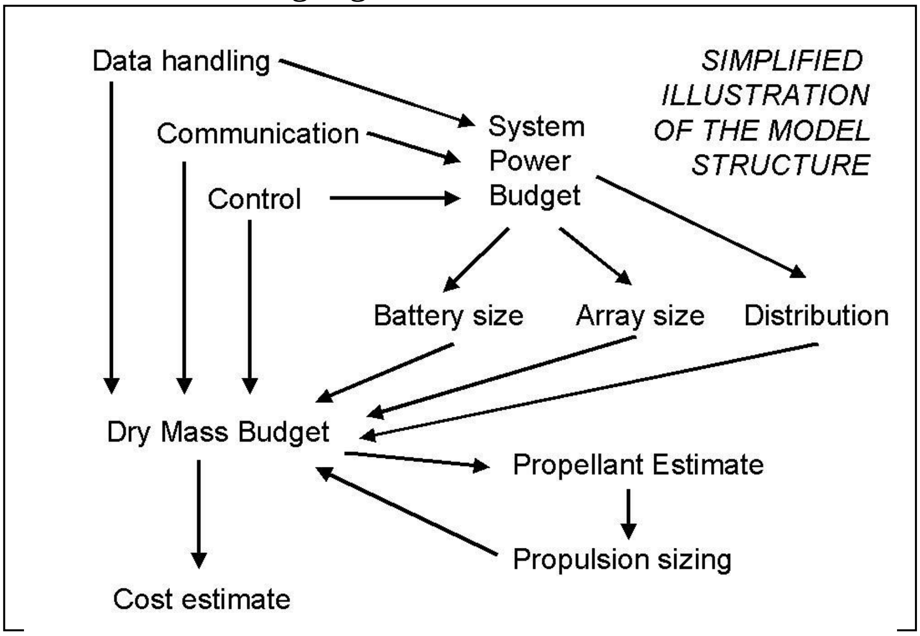
The system pages then read the interfaces from the subsystems pages and creates systems budgets and other system level data automatically.

WHY USE A SPREADSHEET?

- It is easy to set up and use
- Engineers are familiar with spreadsheets
- Easy to verify
- No outside software costs

Some guidelines when setting the model up (from Parkinson)

- Make sure it is readable to other engineers
- Structure the model like the system it models
- Do not hide inputs
- Use colour code to highlight status of data



WHY DO IT?

Once set up and validated (and it had to be reset up for each project) it would instantly assess the system impact (cost, mass, power) of changes to the design.

A version of this model was used for the Eurostar Communications Satellite model. This was a very standardised product and it could be used by a small bid team to input a customer's requirement specification and produce a system definition for bids - almost automatically.

Saved around £1 million (1980s) on bids compared with a full design team producing a full system design.

SO IT'S PERFECT?

The main weakness was inputting the physical position of components - if this was altered it could take some time to alter the model to reflect this.

In theory the model could be set up to "goal seek" optimum design solution - in practice iterations were done "by hand" but it still was a quick way to settle on design values.

EXAMPLE 2 - BAe SYSTEMS (Warton) CAPS

BAe CAPS Model

CAPS was a configuration model used by BAe design team at Warton for fighter aircraft studies.

The programme contained many modules handling different aspects of the aircraft. Engine performance, aerodynamic qualities, control surfaces sizing etc. This combine with overall models for aspects such as mass, flight performance etc. These interact and the programme can be run to establish optimal configuration for a given set of parameters.

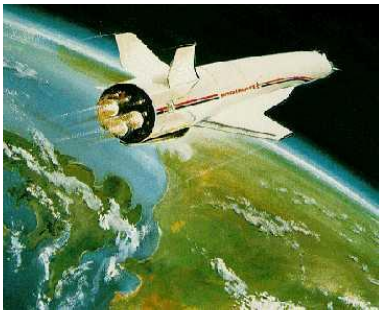
This was used on the Eurofighter Typhoon development programme.

CAPS was based on a mainframe (1980s vintage) and was written by BAe engineers in Fortran code.

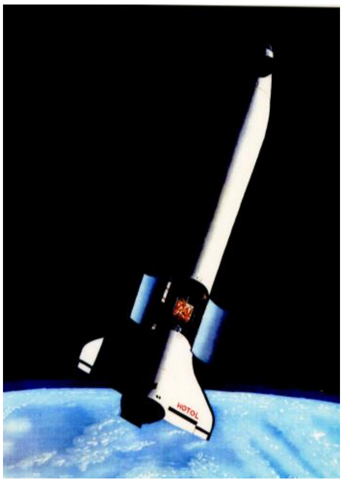
This could be used quickly for the advanced study group to explore different approaches using standard modules. If a study went further the standard modules could be replaced with specialist more accurate modules. As design decisions were made the model became more detailed and refined guiding design team to the optimum system configuration.

The CAPS programme was used during the 1980s HOTOL single stage to orbit launch system design study. This might have been a step too far for a programme that was designed for fighters.

The special modules to replace the standard were added very quickly as the inappropriate nature of the standard modules became apparent. But no-one on the team (or anywhere in the world) had the knowledge to validate the new modules.



Early Design (B or C)



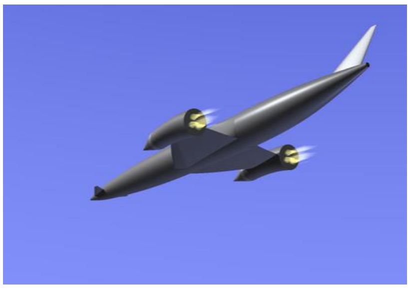
Mid configuation (F)

Last Configuration (J/K)

Although no-one outside the study seemed to notice the shape did keep altering noticeably. CAPS was being used to determine the length to balance between drag and structure weight, and it was only towards the end of the study this had been validated enough to give confidence in the results.

Right at the start the engines were placed in the back - after all it was obvious that is where rockets have their engines. This caused trim problems that were never really solved – and it was too much work to alter CAPS to explore other locations. Note HOTOL was a feasibility study so the team was proving a vehicle could be built and not trying to find what the best design was.

When the concept was taken over by Reaction Engines Ltd to become Skylon – and thus released from the tie to the CAPS programme they put the engines on the wings and solved the trim problem.



The lesson is CAPS, like the satellite spreadsheet model, and indeed any System models that work from the subsystem models up have a problem handling the physical configuration (the layout).

SYSTEMS MODELING LANGUAGE (SYSML)
A more recent development.

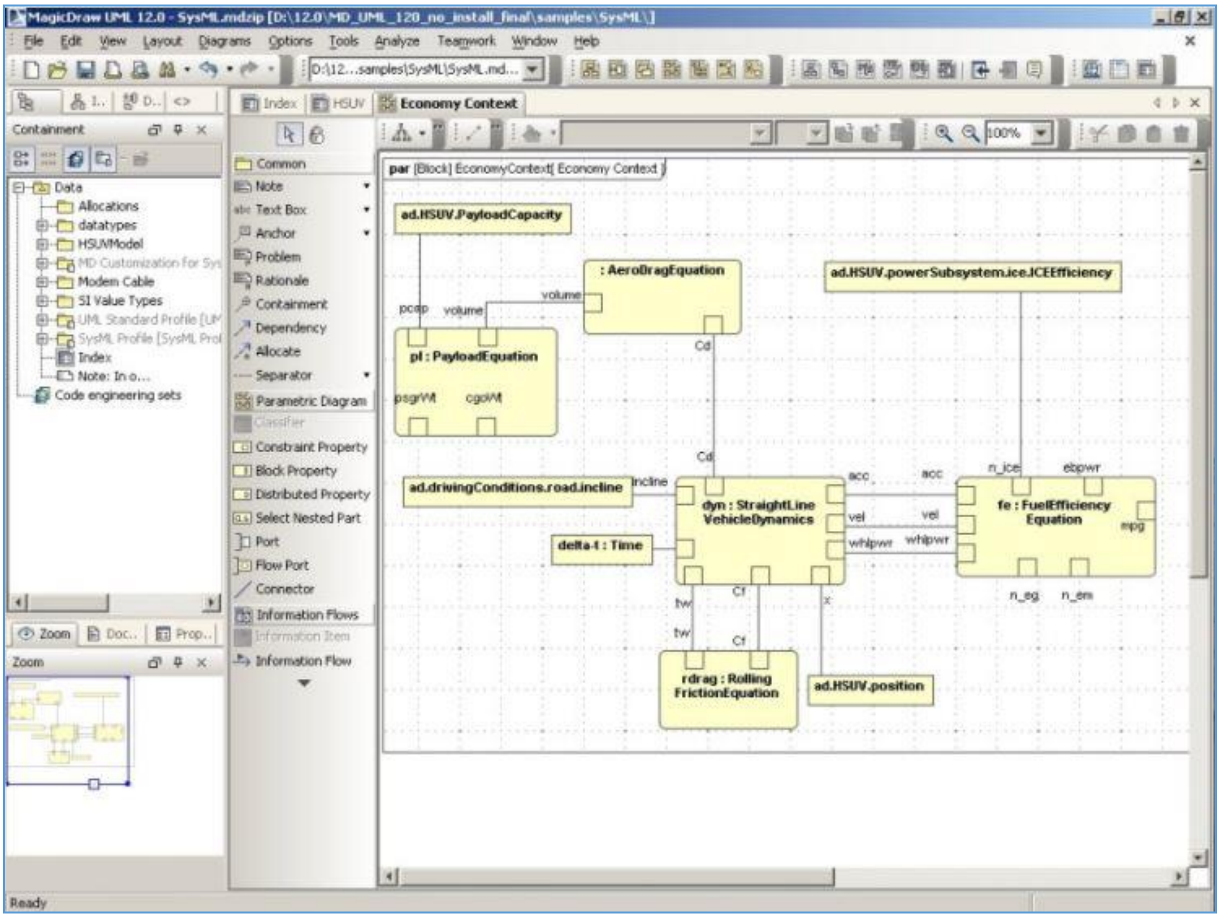
In the 1990s a group created a "modelling language" called The **Unified Modeling Language** (**UML**) as a tool for software development. A modelling language is a general term for artificial languages that use a defined rule set to create an analysable model either through graphics (e.g. a flow chart) or words.

Between 2010 and 2005 a variant was created that was more appropriate to general systems engineering which was called the Systems Modeling Language (SysML). It is primarily a graphics based language using interlinked diagrams. So for example a requirement in the requirement breakdown diagram would be looking for a function in the function breakdown and an element in the system diagram which would have properties derived from direct input or a parametric diagram etc.

This was supported by INCOSE and is an open source standard.

To quote the sales blurb:
The Systems Modeling Language (SysML) is a general-purpose architecture modeling language for Systems Engineering applications.
SysML supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. These systems may include hardware, software, information, processes, personnel, and facilities.

An example of a commercial product using it is Cameo Systems Modeler from No Magic Inc, which is now part of Dassault. The SysML and programs like Cameo have the same relation as English and MSWord.



Such programs do what the Parkinson BAe space systems spreadsheet and the BAe CAPS do; only better and a way that can be used in the general world (as opposed to within a specialist team.)

MODEL BASED SYSTEMS ENGINEERING (MBSE)

There is a movement towards trying to incorporate the systems model as the main design tool at the heart of the system development rather than a support analysis tool. This is called Model Based Systems Engineering (MBSE)

The vision is that everything from the requirements breakdown to the system design with all the elements and interfaces is included in one giant model that would produce the requirements and other documentation, through to the design instructions (drawings, manufacturing procedures, 3D printing file etc.) and then the test procedures.

The objective being that such a model would ensure that everything would be consistent and the complete system wide impact of any changes could be instantly assessed.

A second advantage is the level of analysis that can take place at any stage can be more detailed and comprehensive, or looked at another way the various analyses can be done earlier in the development cycle.

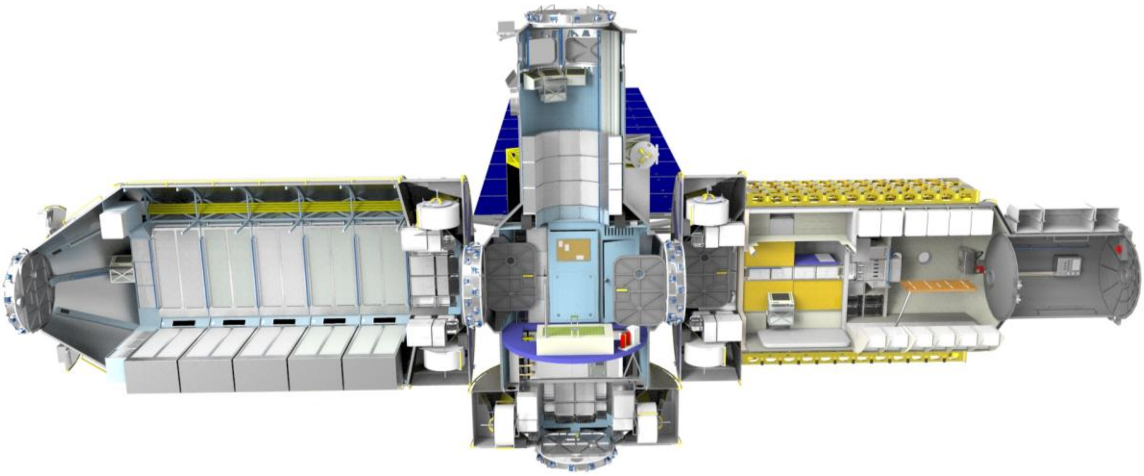
GEOMETRY MODELS

There is another set of systems models that have grown out of 3D CAD packages. Of which in the aerospace industry the standard is CATIA. This programme is marketed by IBM and Dassault.

A CATIA model of a car

Such 3D CAD packages are now available for personal computers such as

- AutoCad Inventor
- Solidworks (Dassault again)
- IronCAD



An IronCAD model of a space station.

These packages can produce the manufacturing drawings, or these days more likely the inputs to the computer driven manufacturing machines such as 3D printers.

They soon added packages to the 3D drawing package to calculate mass properties, tank volumes and other geometry related items from the 3D model. And in more recent times it has moved closer to a system model with additions to examine human factors, integral structure modelling, and other analysis tools related to the geometry.

So this package solves all the weaknesses of a “subsystem properties” System Model where the link with geometry is the problem area. However there are a whole load of different problems as the subsystems properties are either very difficult or not possible to incorporate. Therefore by themselves these 3D packages do not cover everything that is needed to make design decisions.

THE FUTURE

Of course the way round this is to run both sorts of model. The issue is making sure the System Architecture model’s status matches that of the 3D geometry model. This must essentially be done “by hand” and in large systems this can be time consuming and prone to error. However it is still current best practice to get the fastest most detailed guidance for the design of complex systems.

However when the 3D geometry models are fully incorporated into the Model Based Systems Engineering we have the potential to explore very detailed and accurate simulations of the systems being developed early in the program even at the system design phase – so real validation activities could take place before the real money is spent. And evaluating and incorporating any changes identified as desirable could be realistically incorporated from a time and cost point of view – everything from a revised requirement specification to revised system design and the subsystem specification all consistently changed.