Why do MBSE?

Models are created to deal with complexity. In doing so they allow us to understand an area of interest or concern and provide unambiguous communication amongst interested parties.

MBSE goals:

- Improved communications
  - With stakeholders
  - Within the engineering project teams
  - Across spoken language barriers
- Improved quality
  - Early identification of requirements issues
  - Enhanced system design integrity
  - Improved specification of allocated requirements to hardware and software
  - Fewer errors during integration and testing
  - More rigorous requirements traceability
  - Consistent documentation
- Increased productivity
  - Improved impact analysis of requirements changes
  - Improved interaction across a multi discipline team
  - Reuse of existing models to support design and technology evolution
  - Auto-generation of documentation
- Reduced risk
  - Improved cost estimates
  - Early, and on-going, requirements validation and design verification

…and why not?

MBSE is not the answer in isolation from other practices.

- It is hard to model non-functional requirements.
- The model can be a barrier to understanding for some stakeholders
- Effective MBSE requires a disciplined and well trained project team and a mature process approach

Model Based systems engineering has the model, or models as the primary data source.

Model Driven Development uses the activities associated with modelling to drive the whole development process.

Dos and Don'ts for MBSE:

Do define an approach to MBSE which applies to your particular project
Do plan and manage the modelling process
Do define the system of interest and keep the models as simple as possible
Do add extra content as required to solve a particular need
Do understand the assumptions made within the models – all models are incomplete by definition
Do verify your models as you develop them
Do question simulated results
Don't model in isolation from the rest of the design team
Don't model for the sake of it
Don't assume a standard MBSE methodology will simply give you answers – it still requires engineering know-how
Don't simply believe what a tool vendor tells you – verify that the tool gives you what you need in your overall approach
Don't model without understanding the inputs and outputs of the modelling exercise
Don't use the same data to develop and test the model

This leaflet

This leaflet is intended as a brief introduction to challenges of a Model Based Systems Engineering approach

For further information, advice and links to helpful websites go to: www.incoseonline.org.uk

Download copies of this leaflet and other Systems Engineering resources online at: www.incoseonline.org.uk

For more information about the worldwide Systems Engineering professional community, go to: www.incose.org

Series editor: hazel.woodcock@uk.ibm.com
Lead author: robbie.forder@hiqsigma.com

Z9

Issue 1.0
January 2012

What is Model Based Systems Engineering

MBSE is:

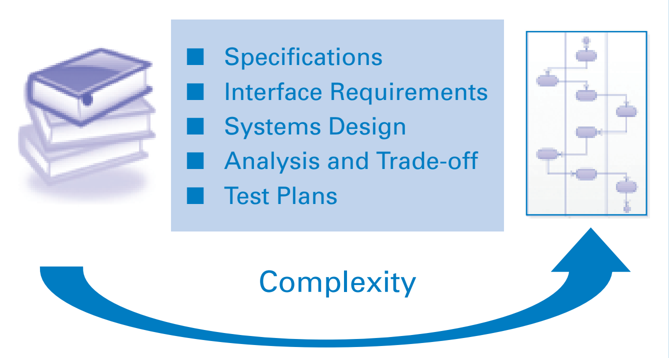
The formalised application of modelling to support:

- System requirements
- Analysis
- Design
- V&V activities

Beginning in the conceptual design phase and continuing throughout development and later lifecycle phases.

MBSE is a term that predicates the use of modelling to analyse and document key aspects of the Systems Engineering Lifecycle. It is broad in scope, spanning the SE lifecycle and covering levels from System of Systems to individual components.

MBSE is a model-centric approach providing a single point of truth which is reflected in a set of living artefacts.



- Specifications
- Interface Requirements
- Systems Design
- Analysis and Trade-off
- Test Plans

Defining and Implementing Effective Process

The use of standardised modelling languages, such as the Unified Modelling Language (UML) or Systems Modelling Language (SysML), is desirable as they are understood across a significant proportion of the SE community. However, since it is important to model to the level of the intended audience, it may sometimes be necessary to use an alternative notation.

Increasingly technology is moving into the virtual/conceptual world of dynamic simulation. These models can often be run in real time to give a virtual response close to the actual system, dynamically adapting to meet changes in tolerance or responses to events.

There are a multitude of modelling techniques and approaches that fall within MBSE. These include:

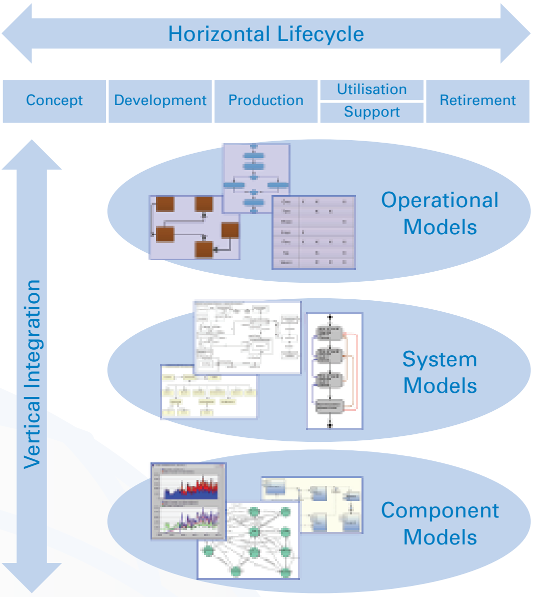
- Structured Analysis and Design
- Data Flow Diagramming
- State Transition Diagramming
- Behavioural Modelling
- Entity Relationship Modelling
- Finite Element Modelling
- Environment Virtualisation
- Computer Aided Design (CAD)
- Analytical Modelling
- Process Modelling

There is also a range of methodologies that support the MBSE approach. One methodology that has been supported by INCOSE specifically is Object-Oriented Systems Engineering Methodology (OOSEM). There are however many others, often produced by tool vendors; examples include the Rational Unified Process (RUP), Harmony SE, and Vitech's Model-Based Systems Engineering Methodology. It is often a good idea to start with a predefined methodology, or even a combination of methodologies, and then tailor this to suit the needs of a specific SE problem.

Modelling needs to be managed; without control or planning it is likely to result in rubbish in/rubbish out. It is important to record objectives and assumptions in a similar manner as when defining the SE approach. This is particularly true where the modelling and SE process become one and the same thing.

Fundamental Concepts and Enablers

Models can be either abstractions or representations of reality that facilitate the understanding of complexity. MBSE commonly uses multi-user repository-based modelling tools which provide an environment where a precise and unambiguous world view of the parts of the system and their behaviours and interactions can be defined and managed. This can be applied horizontally to support to the SE lifecycle process and vertically from integration into the implementation disciplines.



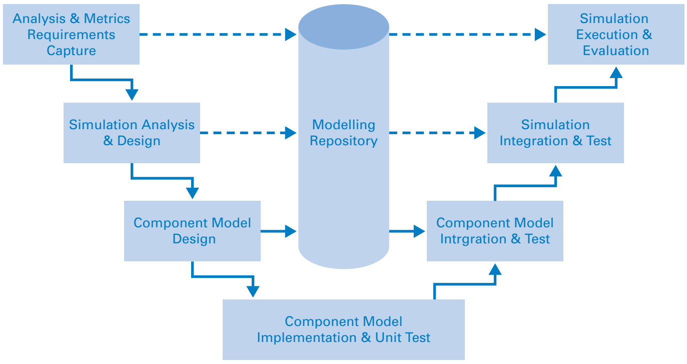
A move to MBSE requires a cultural change and a different mindset. It is one where modelling is used to capture much of the required data.

With MBSE the system solution and design, as described in the modelling environment, are allowed to evolve, with increasing detail being added as required. Documents detailing particular design baselines are usually automatically generated.

This provides several advantages:

- Systems Engineers focus on the technicalities of the problem rather than document structure
- Diagrammatic descriptions are often less ambiguous than textual descriptions
- Greater consistency across related documents
- Dependencies are explicitly captured across stovepipes resulting in less duplication and inconsistency

Scoping the modelling to meet its objectives is essential to managing the evolution of the solution and its implementation through the SE lifecycle.


Analysis & Metrics Requirements Capture
Simulation Analysis & Design
Component Model Design
Component Model Implementation & Unit Test
Modelling Repository
Component Model Intrgration & Test
Simulation Integration & Test
Simulation Execution & Evaluation

Architecting processes capture elements, relationships, and attributes that are needed to describe a system architecture which forms a variety of viewpoints that address stakeholder concerns [see Z8 System Architecture]. It is the Systems Engineer's job to assimilate all these into one coherent representation, or model, of reality.

Models are commonly used to describe or capture the architecture. Equally the data contained within architectural descriptions feeds into models that aid the understanding of system structure or performance, such as simulation or a decomposition of functions.

A high level system model can be built, which allows the collaborating team to visualize the entire system and the surrounding environment. Following decomposition into sub-functions or components, models can then be used to map onto physical architectures. This allows validation of the resulting system to take place and for the customers / users to more clearly see their vision earlier than would otherwise be possible.

The models can be used as integration test benches to support Integration, Verification and Validation testing throughout the SE lifecycle.

The Modelling Language is just the language, and must be combined with a methodology to be useful.