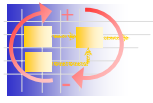


ESD.36J System & Project Management



Integrated Concurrent Engineering

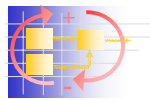
Instructor(s)

Prof. Olivier de Weck

Lecture 12

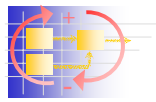
Oct. 14, 2003

Massachusetts Institute of Technology 



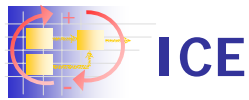
Today's Outline

- Introduction to Integrated Concurrent Engineering 15 min
 - What is Integrated Concurrent Engineering?
 - Why is it important?
 - How can Information Technology support product/system development?
- Presentation: FIPER by Engineous Inc. 30 min
 - Justin Vianese
- Presentation: CO by Oculus Technologies Corp. 30 min
 - Matthew Wall
- Open Q&A and discussion session 15 min
 - What is the relationship between System Project Management and ICE?

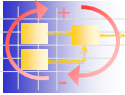


ICE versus Traditional Design

- Traditional Approach
 - Sequential, “toss it over the wall” design
 - Aircraft Design: Aerodynamics --> Propulsion --> Structures --> Flight Controls
 - Advantage: decoupled, traditionally accepted
 - Key problems
 - Iterations are not planned
 - If iterations are needed, they are very time consuming
 - Downstream teams have reduced design freedom and have to live with the “consequences” of upstream decisions
 - Sub-optimal designs will generally result.




Integrated Concurrent Engineering includes a set of principles, methods and tools whose aim it is to design systems and products with all relevant technical and non-technical disciplines in mind such that the system is optimized as a whole, rather than as a collection of sequentially designed subsystems.




Engineering Design Disciplines

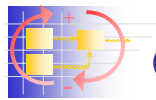
<p><u>Aircraft:</u> Aerodynamics Propulsion Structures Controls Avionics/Software Manufacturing others</p>	<p><u>Spacecraft:</u> Astrodynamics Thermodynamics Communications Payload & Sensor Structures Optics Guidance & Control others</p>	<p><u>Automobiles:</u> Engines Body/chassis Aerodynamics Electronics Hydraulics Industrial design others</p>
---	--	---


 Fairly mature individually, but concurrent design offers significant potential benefits.

4-Sept 03 - ESD.36J SPM


5

Here are 3 applications areas where MDO has been used extensively. Listed are the traditional engineering disciplines for each. Each one of these disciplines can be thought of as being fairly mature, with its own well-developed set of tools. For example CFD has become a commonplace tool for aerodynamic design, while structural dynamicists typically use high-fidelity FEMs. These disciplinary tools continue to grow and develop, enabling many complex analyses that previously were not possible. However the real challenge for MDO is how to properly integrate disciplinary tools and consider the entire system simultaneously. We will talk more about this issue later in the lecture. It is important to note however, that many engineering systems have a large amount of coupling between disciplines, and thus consideration of the multidisciplinary nature of the system is critical. Shown here is the Boeing BWB which is an advanced aircraft concept in the preliminary stages of design. The engineering disciplines on the BWB are highly coupled – even more so than for a conventional tube and wing aircraft. MDO has been invaluable in the design of this aircraft.



Concurrent Engineering Disciplines

Must also include the broader set of concurrent engineering (CE) disciplines.

- Manufacturing:** Model manufacturing tools and processes as a function of part geometry, materials, and assemblies
- “Illities”:** Model parts reliability and failure rates, estimated down-time due to repairs etc...
- Cost:** Estimate development, manufacturing and operations costs.

Prerequisites:

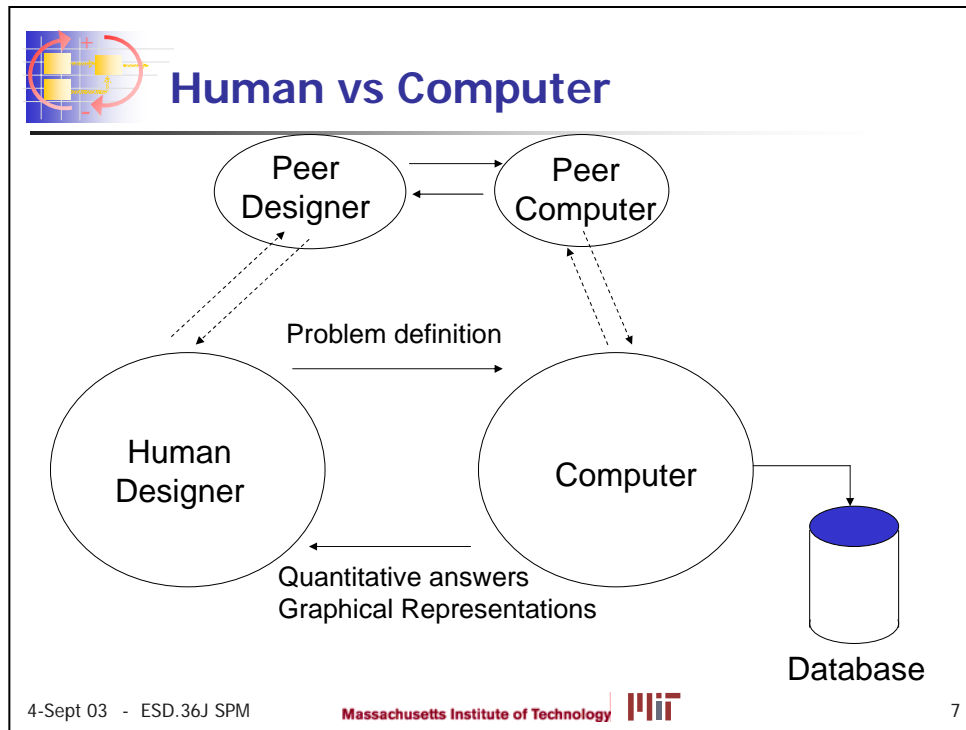
1. Development of realistic, easy to use models
2. Integration of these models

4-Sept 03 - ESD.36J SPM

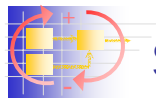
Massachusetts Institute of Technology 

6

CE is a systematic approach to the integrated, concurrent design of products and related processes, including manufacturing and supportability. From the outset, consideration must be given to all elements of the product lifecycle from concept through design, manufacturing, operation and disposal – including quality, cost and schedule. In order to truly optimize a system, MDO must encompass these aspects. For example, along with the aerodynamic and structural properties of a given wing design, we should consider the lifecycle cost of the wing. This includes development costs, manufacturing costs, operating costs and disposal costs. Within each of these, there are several aspects to be modeled. For example, manufacturing cost should account for specialized tooling and process which are a function of the wing design. Within operating cost, one cannot only consider the wing weight, but also its reliability and maintenance costs. It is clear that a truly optimal system cannot be designed by only considering performance issues. Unfortunately this broader set of disciplines do not satisfy a closed, clean set of governing equations and are much harder to model. For example cost models are almost exclusively based on empirical data. For this reason, MDO has to date focused almost exclusively on performance. In aircraft MDO, weight has been used as a surrogate for cost. The broader set of CE disciplines are considered only after the “optimal” design has been obtained. This is an area of current research interest.



So one can imagine the design process as an ongoing interaction between a human designer and a computer. Another important part of the design process is the interaction between human beings. These could be people within the same disciplinary team, or interaction between people from different teams. This interaction is an important part of the qualitative design stream. The other human designers will potentially also be using computational models – perhaps in a different discipline or for a different aspect of the system – and will also be experiencing the question/answer stream with their own quantitative tools. It is also possible to imagine extending this network of interactions to consider the interaction between computational designers. This idea is becoming popular with the use of distributed computing, and has been implemented in an MDO context in the program Oculus.



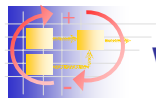
System Level Optimization

Why system-level, multidisciplinary optimization ?

- Disciplinary specialists tend to strive towards improvement of objectives and satisfaction of constraints in terms of the variables of their own discipline
- In doing so they generate side effects - often unknowingly- that other disciplines have to absorb, usually to the detriment of the overall system performance
- Multidisciplinary Design Optimization (MDO)

Example: High wing aspect ratio aircraft designs

We have already begun to see some of the problems that can arise if uncoupled disciplinary models are used to quantify design options. The tendency in any discipline is to strive towards the best possible design from their narrow viewpoint while just satisfying constraints imposed from other disciplines. The trades between competing designs are then often made by considering disciplinary extrema, rather than a systematic exploration of the design space. This more often than not leads to a sub-optimal system design as shown previously for the case of the aircraft wing design.



What MDO really does

MDO mathematically traces a path in the design space from some initial design \mathbf{x}_0 towards improved designs (with respect to the objective \mathbf{J}).

It does this by operating on a large number of variables and functions simultaneously - a feat beyond the power of the human mind.

The path is not biased by intuition or experience.

This path instead of being invisible inside a "black box" becomes more visible by various MDO techniques such as sensitivity analysis



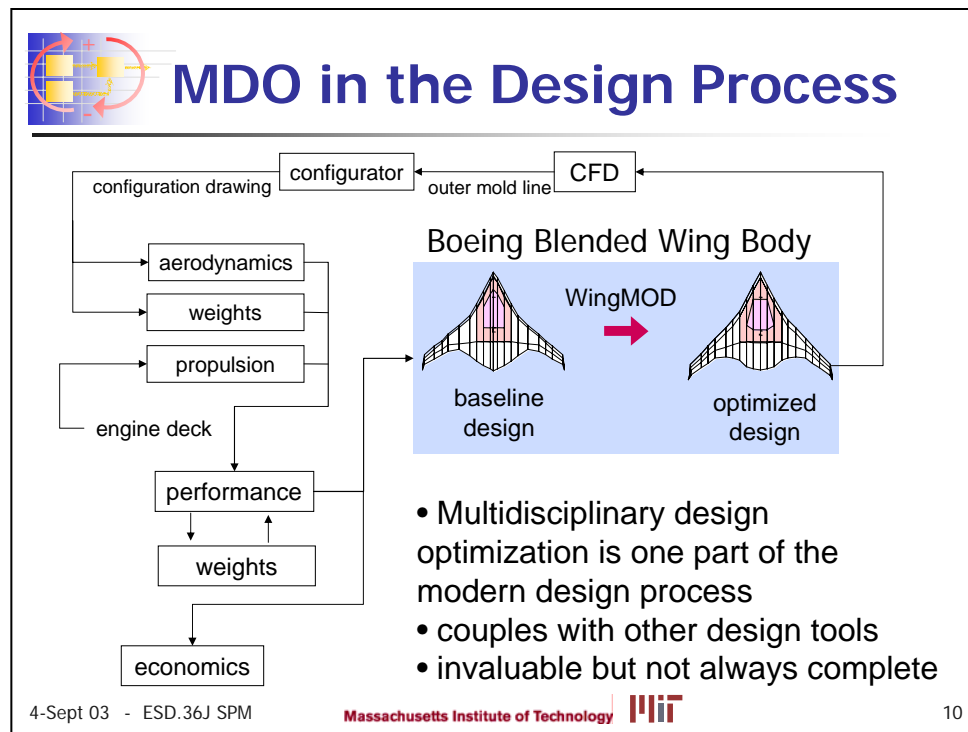
Optimization does not remove the designer from the loop, but it helps conduct trade studies

4-Sept 03 - ESD.36J SPM

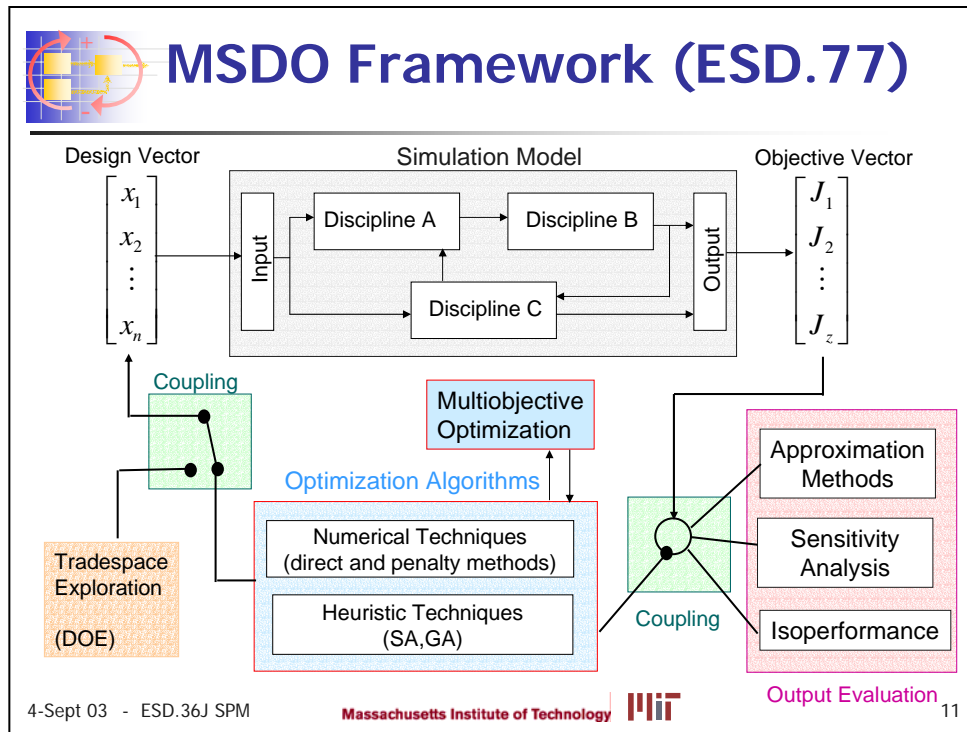
Massachusetts Institute of Technology 

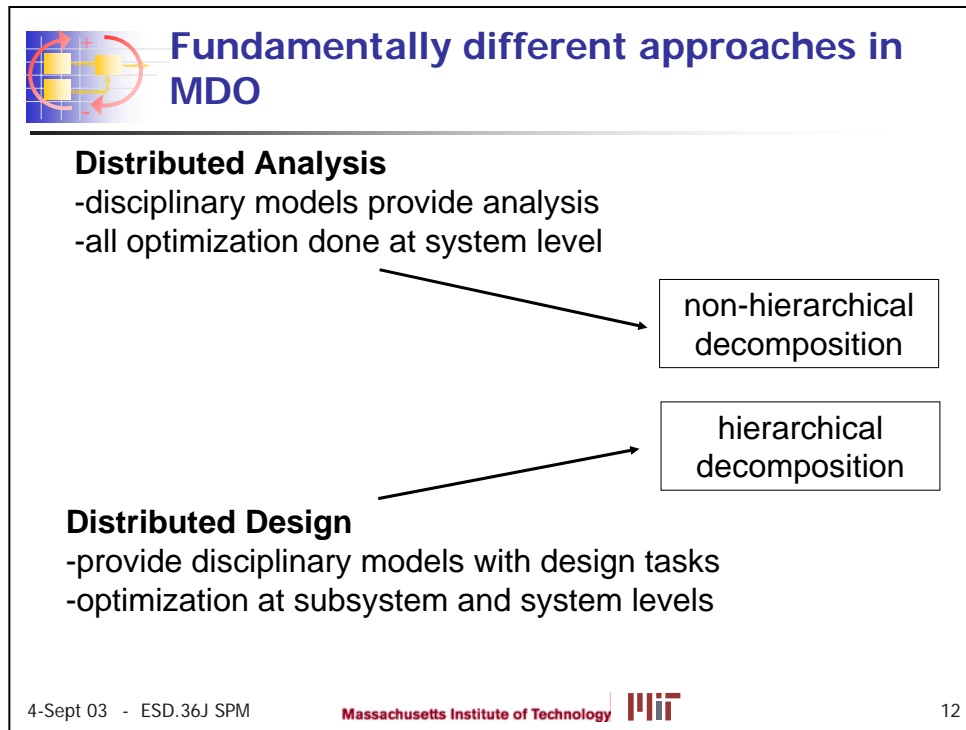
9

So we have seen the mathematical formulation of the optimization problem. What MDO does is to start from some initial design \mathbf{x}_0 and mathematically determine a sequence of designs to step through. This sequence should eventually lead to an improved design with respect to the objective \mathbf{J} . In later lectures, we will see how various optimization algorithms choose this path, but for now we will just comment that for a large design problem, the mathematical operations are very complicated and cannot be performed without a computer. Also, the path is determined strictly by mathematical criteria, and is not biased by designers intuition. Moreover, more insight to the techniques and the path they choose can be gained from using techniques such as sensitivity analysis. We will see more about this in the last third of the class.



So far we have tried to emphasize the fact that MDO is not a push-button tool that replaces other design practices. Instead, you should think of it as one item in your designers' toolbox. This diagram shows how MDO might fit into the design process for the BWB. The blue box represents the MDO code: it takes some baseline design and runs the optimizer to try and improve this design, resulting in the so-called "optimized design". However, as we will see in a few slides, the MDO tool must sacrifice some disciplinary fidelity in order to encompass multiple disciplines simultaneously. For example, WingMOD uses a vortex-lattice method for the aerodynamics. So we know that the aerodynamic properties predicted by the MDO code for the "optimized design" may not be the most accurate we can achieve. The MDO solution gets sent to the aero disciplinary specialists who run CFD codes on it. They will have to make slight modifications to the outer mold line and then pass those on to the configurator who creates a configuration drawing and makes sure that all the parts of the aircraft come together properly. This modified design then goes through a detailed disciplinary analysis and we arrive back with a new baseline design. At this point we could think of another pass through the system, or perhaps adjusting some of the system parameters or constraints. MDO enables large improvements in the solution – one could imagine the loop without the MDO box taking a lot more time and perhaps not approaching as good a design – but it is also important to understand just how MDO fits into the overall design process. Obviously the level to which MDO is a player depends on the problem at hand. For some problems, the MDO analysis may be sufficient.

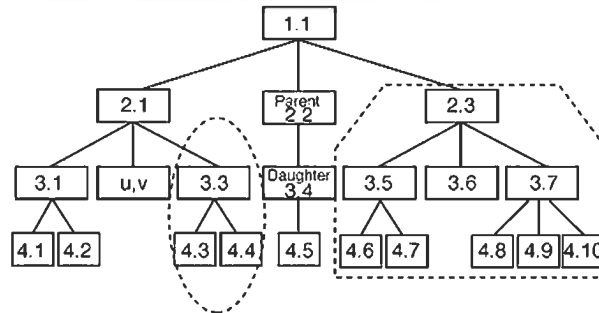




We are now going to talk a little about the details of how to set up an MDO framework. There are two fundamentally different approaches. They go by various names, but we will divide them into Distributed Analysis and Distributed Design. In distributed analysis, the disciplinary models only provide analysis. All the optimization and design variable control is done at a system level. This can also be thought of as a non-hierarchical decomposition. In distributed design, the disciplinary models are provided with design tasks and actually do some optimization themselves. There is a system-level optimization which controls these design tasks.



Hierarchic Decomposition



A hierarchic system.

- analysis information flows top down from “parent” to “daughter”
- top-down flow ends when it reaches bottom level of the tree
- then each “black-box” solution is available and the optimizations begin progressing from the bottom level up
- the top parent represents the system level objectives and constraints

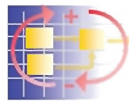
4-Sept 03 - ESD.36J SPM

Massachusetts Institute of Technology 

13

Hierarchic decomposition. The concept of a hierarchic decomposition for engineering design was introduced in [24] using the algorithm from [25] as means for efficient calculation of the optimum sensitivity derivatives. Examples of this type of decomposition applied to structures may be found in [26], and a demonstration of its usefulness in multidisciplinary optimization to aircraft configuration was given in [27]. The hierarchic decomposition method exploits a special way in which the computational and decision making operations may be arranged in the design process of an engineering system. The arrangement is illustrated in Figure 19. Each box represents analysis and optimization of a subset of the entire system problem. The analysis information flow is topdown from the "Parent" black-box to the "Daughter" black-box. For example, a finite element analysis of the entire airframe may be a Parent that transmits the boundary forces to a Daughter wing substructure and the natural vibration frequencies and modes to another Daughter representing aeroelastic behavior. The topdown flow ends when it reaches the bottom level of the black-box pyramid. Then, each black box solution is available and the optimizations begin progressing from the bottom level up.

Inputs received by a Daughter from a Parent are frozen as constant parameters for the duration of optimization performed inside of the Daughter black-box. Moving up to the Parent, one transmits the results of the Daughter optimization augmented with the derivatives of these results with respect to the parameters that the Parent has sent to the Daughter. These derivatives enable the Parent optimization to account by linear extrapolation on the effect of the Parent design variables on each Daughter constraints. The procedure continues to the top of the pyramid. The top Parent represents the system level objectives and constraints and is controlled by the system level design variables. The effects of these variables on all the black-boxes in the pyramid below are accounted for by the optimum sensitivity derivatives transmitted from below. Since the procedure is based on first derivatives, it takes a few iterations to converge. Each iteration consists of the analysis sweep top-down and the optimization sweep bottom-up. With careful implementation the optimization on successive iterations becomes more efficient if warm/hot start capabilities are used. Since the Daughters do not communicate at the same level (no information transmission among sisters), the individual black box analyses and optimizations at each level may be performed in parallel.



Non-Hierarchic Decomposition

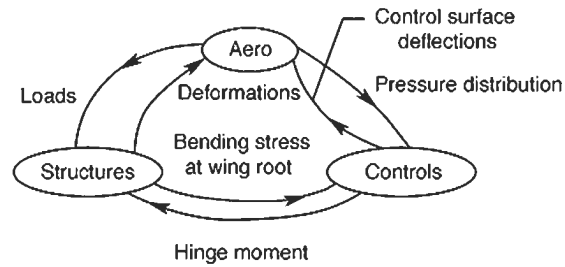
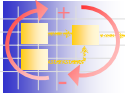



Fig.20 A nonhierarchic system.

- allows information transmission between “black boxes” on same level
- some highly coupled systems cannot be arranged as parent-daughter
- optimization is executed as a single operation for the entire system
- guided by system sensitivity, i.e. derivatives of the system behavior (response) with respect to the system design variables



The bottom line for ...IT

- Need some kind of database to store design variables, constraints, objectives ...
- Need to synchronize various design processes:
 - variable/parameter updates
 - performance, cost computations
 - system level and subsystem optimizations
- Would like to keep interface general and user friendly
 - don't "hard-code" problem specific details
- Can be a serious problem for large systems
 - coordinate various design teams, suppliers ...
- **Let us now consider the industry/IT perspective**
 - **Engineous/FIPER**
 - **Occulus/CO**

4-Sept 03 - ESD.36J SPM Massachusetts Institute of Technology  15

One thing that has not been addressed in detail but which is very important for MDO is data management. It is easy to see that complex systems generate a large number of design variables and constraints and that it can quickly become uncontrollable. We would really like to keep the interface general and user friendly, however this remains an issue for many MDO codes.