Struct Multidisc Optim (2010) 40:17–33
DOI 10.1007/s00158-009-0381-5

FORUM ARTICLE

MDO: assessment and direction for advancement—an opinion of one international group

Jeremy Agte · Olivier de Weck · Jaroslaw Sobieszczanski-Sobieski · Paul Arendsen · Alan Morris · Martin Spieck

Received: 10 October 2008 / Revised: 17 February 2009 / Accepted: 12 March 2009 / Published online: 17 April 2009

Abstract This paper is a summary of topics presented and discussed at the 2006 European–U.S. Multidisciplinary Optimization (MDO) Colloquium in Goettingen, Germany, attended by nearly seventy professionals from academia, industry, and government. An attempt is made to accurately reflect the issues discussed by this diverse group, qualified by interest, experience, and accomplishment to present an opinion about the state-of-the-art, trends, and developments in Multidisciplinary Design Optimization. As such, its main purpose is to provide suggestions and stimulus for future research in the field. The predominant content of the colloquium was centered on aerospace, with a few contributions from the automotive industry, and this is reflected in the article. Due to the timeframe that has passed since the conclusion of the workshop, the authors have updated topics where appropriate to reflect observed developments over the past 3 years. Finally, rather than dwelling extensively on past accomplishments and current capabilities in MDO we focus on the needs and identified shortcomings from the colloquium which lead to potential future research directions. A brief MDO background is provided to set the discussion in its proper context.

Keywords Multidisciplinary design optimization · MDO · Review · Trends · Current

A previous version of this article was published as AIAA-2007-1905 and presented at the 3rd AIAA Multidisciplinary Design Optimization Specialist Conference, April 2007

J. Agte (✉) · O. de Weck
Department of Aeronautics & Astronautics, MIT,
Cambridge, MA, USA
e-mail: jagte@mit.edu

O. de Weck
e-mail: deweck@mit.edu

J. Sobieszczanski-Sobieski
Research & Technology Directorate,
NASA Langley Research Center,
Langley, VA, USA
e-mail: jaroslaw.sobieski-1@nasa.gov

P. Arendsen
Collaborative Engineering Systems Department,
NLR, Amsterdam, The Netherlands
e-mail: arends@nlr.nl

A. Morris
School of Engineering,
Cranfield University, Cranfield, Great Britain
e-mail: a.j.morris@cranfield.ac.uk

M. Spieck
Institute of Aeroelasticity,
German Aerospace Center,
Goettingen, Germany
e-mail: martin.spieck@dlr.de

Abbreviations

F	Functional behavior (“performance”) of a system;
$\mathbf{g}, \mathbf{h}$	Inequality constraints, equality constraints;
i, j	Variable, module or subsystem index;
m	Number of fixed parameters;
n	Number of design variables;
$\mathbf{p}$	Vector of (fixed) parameters;
$\mathbf{x}$	Vector of design variables;
N	Number of modules or subsystems;

Y Module or subsystem inputs;
Z Module or subsystem outputs

1 Introduction

This paper is not a formal survey of the field of Multidisciplinary Design Optimization and the authors are not a recognized technical committee empowered to report on MDO. They do, however, represent a fairly broad-based group of individuals who are actively involved in MDO, both theory and applications. The group acted as the technical organizing committee for the 2006 European–U.S. Multi-Disciplinary Optimization (MDO) Colloquium that was hosted by the German Aerospace Center (DLR) in Goettingen, Germany, from 17–19 May, 2006. The workshop's emphasis was on uncovering gaps and shortcomings rather than reporting on completed projects and highlighting accomplishments, and it served as the foundation for the content of this article. Here, the specific product that emerged from the exchange of opinions at the colloquium is a list of specific research issues and development directions forming an MDO Research Agenda that the Goettingen participants wish to contribute to the advancement of MDO.

The article complements several other summary articles of similar nature, ref. Giesing and Barthelemy (1998), Bartholomew (1998), Alexandrov (2005), Tedford and Martins (2006) and is not a comprehensive review, as this has been done by Sobieszczanski-Sobieski and Haftka (1997) and others. As such, we solicit comments to the broader engineering community with the purpose of contributing to and inspiring debate. Due to the broad nature of work that has been and continues to be done in MDO, the authors admit that certain important accomplishments may not have been equally represented in the three-day workshop and apologize beforehand for any such exclusions. We emphasize that these exclusions are merely a reflection of the colloquium audience composition and not a result of any type of priority rating.

2 Background

The roots of MDO are found in structural optimization. This is not without good reason as in nearly all engineering systems there is a structure to which other subsystems attach, and much of the subsystem interactions involve the structure as a conduit. In the field of optimization the Nonlinear Programming (NLP) formalism was first transplanted to structural optimization practice in Schmit's seminal work on a simple three bar truss (Schmit 1971). The general formulation of an NLP in this context is shown in (1):

$$
\begin{aligned}
&\min\ f\left(\mathbf{x}, \mathbf{p}\right) \\
&\mathbf{x} = [x_1 \dots x_n]^T, \mathbf{p} = \left[p_1 \dots p_m\right]^T \\
&x_{i,LB} \le x_i \le x_{i,UB}, \quad i = 1, 2, \dots n \\
&\text{s.t.}\ \mathbf{g}\left(\mathbf{x}, \mathbf{p}\right) < 0,\ \mathbf{h}\left(\mathbf{x}, \mathbf{p}\right) = 0
\end{aligned}
\tag{1}
$$

where f is the function to be minimized, $\mathbf{x}$ is a n-dimensional vector of design variables with lower and upper bounds, $\mathbf{p}$ is a vector of fixed parameters that influence the behavior of the system but cannot be freely chosen (material properties, operating conditions, ...), and $\mathbf{g}$ and $\mathbf{h}$ are inequality and equality constraints, respectively. This notation can vary but is generally widely accepted. By assigning meaning to the design variables (e.g. beam cross sections, plate thicknesses...) and choosing a relevant objective function (e.g. mass) and constraints (e.g. maximum stress smaller than safety factor times yield stress) the NLP formalism became widely accepted, originally competing against, and later having been reconciled with, the optimality criteria approach rooted in the time-honored concept of fully-stressed design. In a few years, the concept spread throughout structural engineering and attracted interest in other disciplines, first in aerospace applications where weight (mass) was crucial.

Eventually, this led to inclusion of disciplines other than structures in the structural optimization loop as sources of data, again in aerospace first where interdisciplinary coupling to structures was too strong to be neglected. In the next logical step, the design space ($\mathbf{x}$) was extended to include variables intrinsic to the other disciplines such as the aerodynamics (airfoil and wing shape), aircraft performance (mission profile), and propulsion (engine thermodynamics), initiating MDO on a growth path toward encompassing the entire vehicle as a system.

Consistent with the computer technology limitations of the time, the first MDO efforts developed at *two levels*. At one level, structural, aerodynamics, and aircraft performance analysis codes, all processing mathematical models whose fidelity could be categorized as *medium* by the standard of the day, were nested in a single optimization loop (Fig. 1) operating on a very limited (a dozen or less) design variables to optimize a system level objective such as flight range (Fulton et al. 1974). At another level, the number of disciplines was increased to a more complete set typical for a

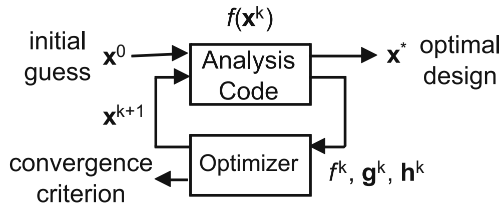

Fig. 1 NLP—simple optimization loop

conceptual design stage at the price of lowered fidelity to gain a fairly realistic representation of the design process at that stage (Vanderplaats 1976).

Advancing computer technology enabled higher fidelity codes to be processed faster and gradually erased distinction between the above two levels leading to large monolithic codes invoking several disciplinary mathematical models in a single optimization loop (see again Fig. 1). Intrinsic practical limitations of the monolithic systems of disciplinary codes soon became apparent (for large *n*) and stimulated development of decomposition methods intended to break large tasks into sets of smaller ones while preserving the couplings (Balling and Sobieszczanski-Sobieski 1996). See Fig. 2 and a summary in Sobieszczanski-Sobieski and Haftka (1997).

At a high level, one began to distinguish between system level optimization and subsystem level optimization, based on the fact that for an internally coupled

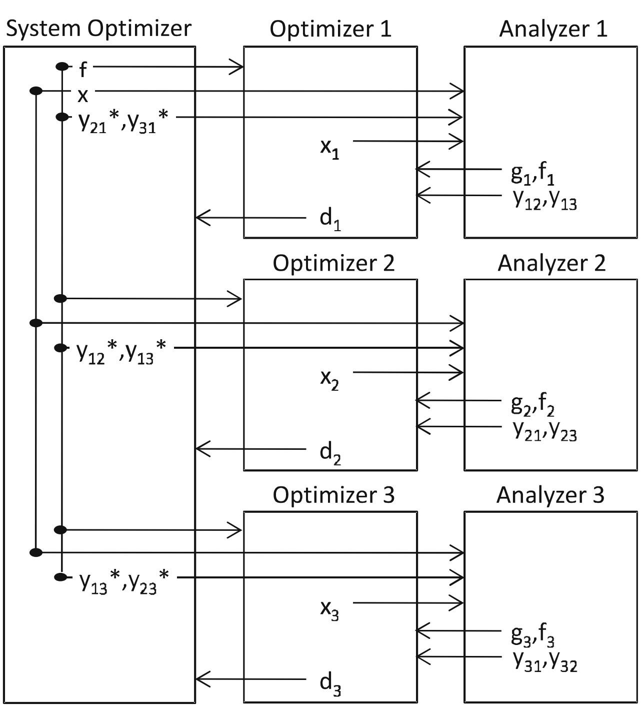

Fig. 2 MDO problem decomposed into two levels—NAND @ disc. level, SAND @ sys. level

system, the optimal system-level design is not simply a collection of individually optimized subsystems. The subsequent emergence of decomposition methods began to differentiate MDO from pure structural optimization, although some of the theoretical underpinnings of the decomposition methodology were adopted directly from this field, including various forms of sensitivity analysis (Sobieszczanski-Sobieski et al. 1982; Sobieszczanski-Sobieski 1990; Van Keulen et al. 2005) and approximation methods (Barthelemy and Haftka 1993), which more recently have become known as surrogate-based methods (Queipo et al. 2005; Simpson et al. 2004).

It should be noted that decomposition as an approach to break a large optimization problem into an equivalent set of smaller, independent but coordinated problems was successfully developed and applied at both the disciplinary and multidisciplinary levels of optimization. At both levels, the motivation for decomposition was the same—compression of the execution time by bringing in more resources, whether human or computational, to operate on the problem at hand. However, the decomposition implementations at the disciplinary and multidisciplinary system levels differed. The former lent itself to a particular partitioning of a set of equations that governed the given discipline. The latter involved many diverse sets of such equations whose interaction often posed a problem within itself, and in large applications involved a number of different teams of specialists. That involvement required (and still requires) dealing with a non-mathematical but crucially important set of human factors as a prerequisite to success.

The above elements combined with the data organization in the format of the design dependence matrix, a.k.a. N-square diagram N^2, or design structure matrix DSM (Smith and Eppinger 1997). This enabled a set of new decomposition methods surveyed in (Sobieszczanski-Sobieski and Haftka 1997). A few examples are Linear Decomposition, Collaborative Optimization(Sobieski and Kroo 2000), Concurrent Subspace Optimization, and Bi-level Integrated System Synthesis (Sobieszczanski-Sobieski et al. 2000). Each of the above decomposition methods contributed a different and specific mathematical solution to the problem of optimization of a large and complex system. It is their mathematical content that makes them distinct from and complimentary to the Computer Aided Design-inspired frameworks dedicated to integration of computer codes in the sense of streamlined data passing, storage, retrieval, and user interface.

Another relatively recent contribution of MDO is frameworks that allow finding optimal system designs for a set of desired targets or goals. This is important as designs in industry are often driven by contractual requirements and not by optimum performance. Such frameworks typically seek one or more solutions that meet a design target within a pre-specified numerical performance. Frameworks that enable goal-seeking design are, among others, Physical Programming (Messac 1996), Analytical-Target-Cascading (Kim et al. 2003) and Isoperformance (de Weck and Jones 2006). An important research direction has also been the development of more powerful model reduction methods that retain the key information in models, while reducing their dimensionality (Willcox and Peraire 2002).

Figure 3 provides a notional timeline of the developmental history of MDO.

3 Successes and identified limitations

As one could argue that the ultimate benchmark of a research field's impact is indicated by the realization of its theories into successful products through industry, the focus in the following examples is on MDO through an industrial perspective. Numerous MDO application methods in various industrial sectors were presented at the Germany workshop with the major users of this type of technology being *the aeronautical and automotive industries*. However, the actual use of genuine MDO methods within industry at large (beyond automotive and aerospace) is still rather limited.

3.1 Industrial successes

Within these major industrial sectors the design process follows a regular development process involving broadly three levels of models:

1. Preliminary design that uses explicit behavior models
2. Configuration design employing implicit low-fidelity models

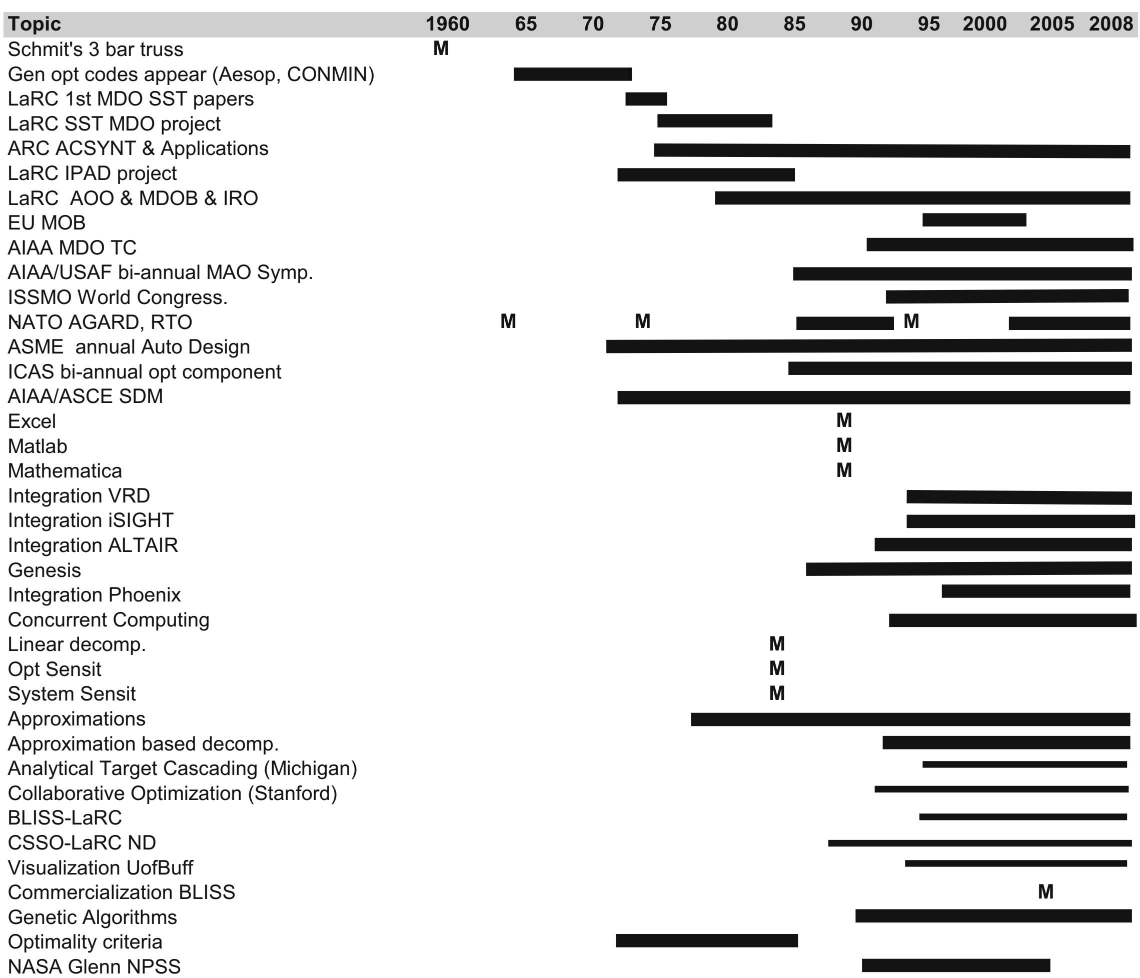

Fig. 3 Notional MDO timeline ("*M*" indicates a milestone)

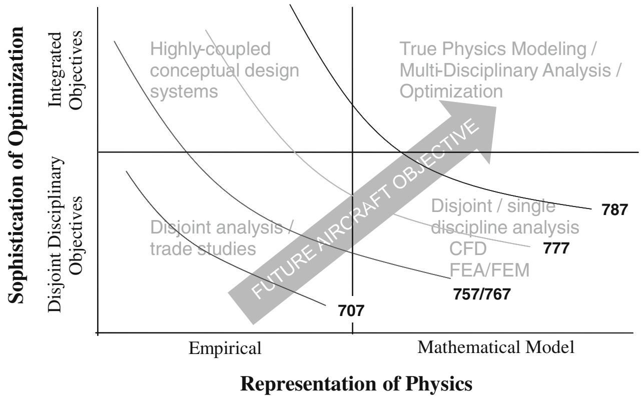

Fig. 4 Reaching towards full a/c MDO across successive a/c families (adapted from Boeing, 2006 European–U.S. MDO Colloquium Proceedings)

3. Detailed design that employs implicit high-fidelity models

The application of MDO methods and techniques in industry has, for the most part, started at the detail design stage and is now slowly, but steadily, moving upstream into employment at the configuration and preliminary design stages. This move parallels a similar trend from component level design, with higher fidelity models, towards a broader system level optimization initially utilizing lower fidelity and/or surrogate modeling. Figure 4 is a depiction of Boeing's systematic progress towards full aircraft MDO over successive new aircraft families as stepping stones through the years.

As discussed, the initial application of optimization methods focused on the creation of satisfactory *structures with minimum weight*, often employing techniques at the detailed design stage that sought vertex solutions in the design space. The field has evolved to the point where major components are optimally designed using very large high-fidelity models in four to five iterations, supported by the continued development of more complex digital models (Fig. 5).

Within the aircraft industry, an MDO approach is being introduced at the detailed design stage by allowing disciplines to interact in a loosely coupled manner. Taking this approach, Boeing reports significant gains in weight reduction on the 787 program, examples of which are indicated in Table 1.

The automotive industry has an apparent lead over the aerospace community in the use of system level optimization with low-fidelity models applied at the configuration design stage. Incorporating MDO into the automotive configuration design phase is presumably easier than undertaking this task in the aerospace industry as disciplines in automotive engineering are more loosely coupled. Within the automotive sector, designs are created in a multi-attribute environment rather than a multi-disciplinary environment. Such aspects as noise-vibration-harshness (NVH), crashworthiness, drivability, etc. provide a coupled design environment where attributes are shared through system level variables. This allows the use of response surfaces and more general design of experiments (DOE) procedures, placing the MDO process firmly within the configuration design phase.

Recent developments at Dassault (Ravachol 2006) are geared towards taking into account that the digital design environment must be integrated with distributed design and manufacturing teams, pushing research into creation of methods that can accommodate virtual design teams. Additionally, they are focusing on integrating manufacturing and downstream requirements into the MDO process. This is accomplished by using Lagrange Multipliers, generated at the detailed design stage, to inform engineers involved with the preliminary design stage of important downstream constraints.

3.2 Limitations identified by industry

In addition to the above successes, a number of needs and limitations relating to the use of MDO methods and tools in industry are evident as listed in Table 2.

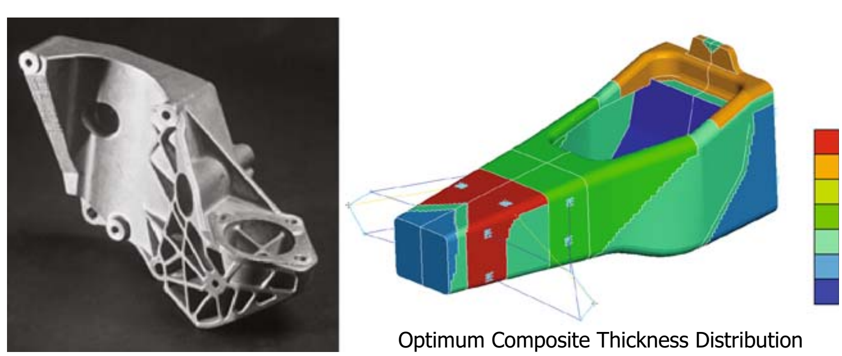

Fig. 5 **a** Lotus optimized engine bracket (Altair OptiStruc). **b** Comp. thickness distribution (VR&D)

Table 1 Examples of gains from the application of MDO technology

Industrial sector	Component/activity	Gain from MDO
Aircraft	Vertical fin major aircraft	Significant increase in effectiveness
Aircraft	Nacelle configuration	Noise reduction with 15% weight reduction
Aircraft	Flight test program	Reduced to less than 1 year (normally 2–3 years)
Automotive	Optimized structural design for crash worthiness	Time to achieve acceptable level of impact performance reduced from 1.5 years to 1.5 days

The last point, the immaturity of model verification and validation—especially across a large design space—is expanded below.

3.3 Model fidelity and validation/verification

An expensive and serious effort is made in most industrial sectors to ensure that the computations process used in the design of a product is based on models that have been both verified and, as far as is possible, validated. Validation implies that a finite element or computational fluid dynamics model has been compared with full scale test and found acceptable. Where no such tests exist because the product has not yet been constructed, structural and aerodynamic specialists will have devised methods that, in some sense, replicate the classical validation process using past experience, specific structural or wind tunnel tests and sensitivity studies. This is very time consuming and expensive but provides models that can be relied upon to replicate the behavior of the in-service vehicle.

In the case of an aircraft, when a structural layout is offered to optimization algorithms within an MDO framework, it is normal to discover that significant changes in aircraft planform result. An experienced engineer would argue that the output from these models cannot be trusted because the structural and aerodynamic models do not relate to the modified configuration. They have not been validated for the modified design. The underlying question being asked is, having started the MDO process with a set of validated and verified models, how far can these models be extended before the results become unreliable? The same questions require answering for all the major industries that intend on using MDO methods. There is, therefore, a real need for industry to provide a basis whereby MDO methods can be validated in a manner that provides limits beyond which MDO results either cannot be accepted or may only be interpreted as merely trend pointers. It is accepted that this is a non-trivial requirement and that a significant amount of work is required. However, information is required on the limits of acceptability of MDO results before involved engineers

Table 2 Needs and Requirements for further MDO development identified by industry

Industry	Requirement/limitations identified
Aerospace	To take full advantage of composites with unconstrained fiber orientation (not only 0, 90, 45)
	Minimize manufacturing and tooling costs
	Composite electromagnetic effect optimization
	Full life cycle business case optimization
	Incorporation of MDO methodology in multi-level efficient strategy, rather than only single level
	Design of truly innovative configurations (e.g. BWB)
Automotive	Response surface and design of experiments methods have difficulty handling the required number of design variables
	Design targets not sufficiently defined to allow the use of MDO, design evolves as more aspects are factored in over time
	Exploit coupling of vehicle physics and manufacturing via variables shared by both areas
	Model verification and validation still immature

make major design decisions based on the output of such systems are made.

3.4 MDO in the workplace

The use and acceptance of MDO in the industry workplace, and to a lesser degree in certain research institutes, hinges upon its functionality at three levels: technical, organizational and cultural. Overcoming the technical barrier is probably the easiest of the three to understand and address. Most MDO 'advocates' and researchers work to prove MDO's technical worth on a day to day basis, many times at the expense of overlooking the other two levels.

At the organizational level, integration that occurs in very large engineering companies has been limited to exchange of data files between departments (organizational units), while departments retain their disciplinary foci. This data exchange facilitates Multidisciplinary Design Analysis (MDA) but not true integration. True integration could be demonstrated by the following example (based on discussion with K. Bhatia of The Boeing Company).

Suppose that during the design process, someone wants to move the engines fifteen inches outboard to improve flow into the engine nacelle. In a integrated process, that desire should be communicated to the computational infrastructure in a natural language, i.e. "MOVE ENGINES 15 INCHES OUTBOARD," to trigger adjustments in the math models associated with structures, aerodynamics, configuration (there might be a flap size to be adjusted), flight mechanics and even "relatively distant" component modification such as vertical tail size, which may need to be adjusted due to the engine out condition. The MDA with the models adjusted would generate data informing the designer of the effects of the change. As far as the designer is concerned, the computing infrastructure would simply be performing a background computation to proceed from the natural language command to the new data output.

In engineering we are nowhere near the above capability. A note of progress made with video games in this regard, however, is interesting. For example, in any one of countless popular combat games the player controls their character's actions by direct console inputs. The other characters on (and even off) the screen act guided by analysis combined with a set of rules. This analysis is surprisingly sophisticated. For example, CFD calculations to show the ripple effects when someone is thrown into a pond of water, or enemy character responses that differ every time the game is played. This 'set of rules' alludes to the use of Artificial Intelligence methods.

Returning to the airplane design, an additional path might be to integrate the configuration group with the disciplinary groups on a similar basis. For instance, fast acting surrogate models would exist as players, where the disciplinary teams hold the console and assume the responsibility for creating and maintaining them, as well as ensuring the process does not push them beyond the bounds of validity. If it does, they update the surrogates. Each team must then have sufficient depth and breadth to monitor, assess, and verify their domain analyses.

The third level is cultural. People strive to live in their most comfortable environment and tend to feel happy there. Only a small minority is positively inclined towards far reaching integration. Many fear loss of control if their domain boundaries open up. Many are concerned about job security. In management, there is more openness to advanced ideas at the higher levels but the middle level managers tend to operate by consensus, busy meeting today's commitments. They are often averse to accepting additional risks. These issues must be purposefully addressed by those in charge before successful large scale integration can reach the next level. This will likely involve allowing an organization's engineers a higher failure rate while trying new things.

3.5 Relationship between academia, industry, and government

Another roadblock to further development of MDO that was discussed at the colloquium exists at the educational level. There are several areas in need of improvement. A primary issue is the lack of education itself, not only at the university level, but also within industry and research organizations.

First addressing the university level, the past 5 to 10 years have seen a relatively healthy growth in the number of dedicated MDO courses taught to graduate level students (although numbers are still small). However, many of these classes are taught as applied math courses whose target audiences are limited in scope and often do not include design engineers. Furthermore, design engineers who do enroll in these courses often receive the instruction too late, at a point in their graduate studies after which they have already completed undergraduate conceptual and preliminary design. Coincidentally, this late consideration of MDO also mirrors that which occurs in industry and research, where MDO practices are often not fully realized in the early stages of design or research development. Although the reasons may be many, this likely also points to a lack of education within and between these respective institutes, where the engineers who can benefit most from MDO applications often are not aware of what is available to them until after preliminary design phases are complete.

Of further concern is the *lack of high quality educational material* available for use in studying and teaching MDO. There are few, if any, undergraduate level textbooks that deal either directly (textbooks dedicated to MDO) or indirectly (textbook incorporating MDO into its subject matter) with the topic of MDO as its own field. A number of effective textbooks on design optimization do exist (Papalambros and Wilde 2000), but they typically only cover a subset of topics.

Additionally, an adequate number of real-world test cases to which MDO methods and practices can be applied in an academic setting are still missing from the educational toolbox. This may partially stem from a relationship between industry and academia that has not yet reached its full potential. Often, the models required for worthwhile MDO analysis, i.e. applying a multi-level MDO method to a complete real-world aircraft configuration, are commercially sensitive data that industry does not want to release to external parties. The fact that MDO is 'multi-disciplinary' also means that the required models cannot be limited to just one or two components, but rather must provide a relatively comprehensive representation of the designed system. All of this leads universities to depend on self-developed test cases that are usually very simplified and do not adequately represent the real-world complexities of industrial applications, further strengthening a barrier that can best be surmounted by continuing government funding.

A central problem inherent to the successful application of MDO is that it requires a broad range of engineers, specifically discipline engineers, who understand MDO concepts and methods. It is not enough to integrate an MDO specialist into an organization and expect this person to exercise the necessary discipline expertise to correctly formulate the design problem, make adjustments when necessary, and evaluate the results at the end. The amount of information and the complexity of the problems are simply too large. To remedy this, there are several recommendations that may improve the education of upcoming design engineers such that the MDO mindset diffuses down to even the lowest levels of the design organization.

The first of these is to move the initial instruction of *MDO concepts from the graduate level down to the undergraduate level*. This does not necessarily imply that MDO should be taught as an independent course, but rather that the fundamentals of MDO problem formulation, available tools, concept of constraints, etc., should be taught alongside the more traditional design course topics. The mechanics of the design space search for a constrained minimum as well as development of surrogate models tailored to the problem at hand is now available in many commercial utility codes (see Appendix). While understanding the way the above search operates is important for successful use, the user is relieved somewhat from the involvement with mathematical detail and is left free to focus more on the physics of the problem to be solved. This paradigm change has a profound potential for changing the way optimization is taught to engineering students, allowing MDO instruction from a physical perspective rather than a mathematical one, with less focus on the details of the algorithms and more on their use.

A second recommendation is directed towards the partnership between industry and academia. On the side of academia, universities need to increase their efforts to *bring in more experience from seasoned engineers*. In this manner, they ensure students are exposed to the difficulties of applying MDO in the real-world and hopefully dissuade them from developing a "push-button" attitude. Industry could help out academia tremendously by working to provide: first, expertise in the manner described above, and second, a suite of test cases representative of real-world design problems to be used in education. As academia is providing the future workforce for industry, it would be in industry's own best interest to do so.

Finally, an effort is urged upon those universities already engaged in the MDO field to work towards *developing useful instructional material* on the subject of MDO in general. Current efforts to familiarize oneself with the topic usually require hours and hours of searching though hundreds of technical papers on the subject and putting it together later in a piecemeal fashion. There still remains some disagreement about whether the field is ready for an item such as an '*MDO Handbook*,' but most agree that a textbook, focusing on MDO from an engineering perspective, is needed. The various companies involved in producing MDO software (see Table 4) could be of invaluable help in this process as the majority of them have already collected and written volumes worth of optimization material in support of their own products.

4 Research agenda

The colloquium identified a number of areas that are indeed promising research directions for the further development of MDO. The contents of the papers, presentations, and discussions at the colloquium suggest categorization of the anticipated MDO research and

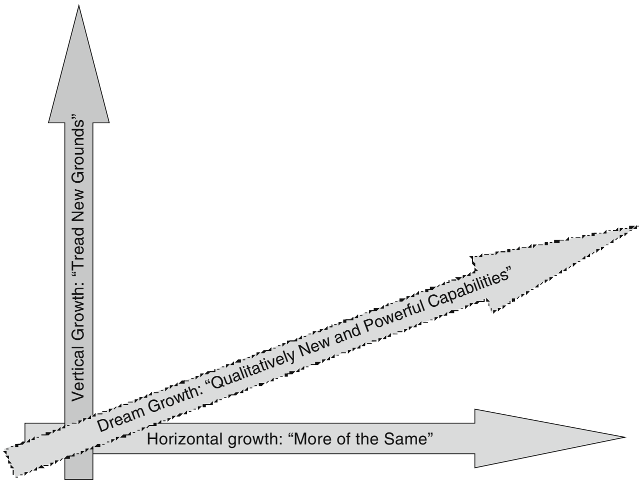

Fig. 6 MDO horizontal/vertical growth

development into two orthogonal directions that might be labeled "*Horizontal Growth*" and "*Vertical Growth*" (Fig. 6). The Horizontal Growth category encompasses developments that improve on the MDO capabilities already established toward greater dimensionalities of the applications and extend the application spectrum, e.g., to include the life cycle, economic factors, uncertainty, and reliability. This leaves the Vertical Growth label to identify developments that are conceptually and qualitatively new, for instance optimization of entire families of products for cumulative return on investment. The two orthogonal growth directions are expected to symbiotically reinforce each other into a "dream growth" delivering capabilities that are both qualitatively new and powerful in terms of the size of the problems they will solve.

4.1 Horizontal growth of MDO

4.1.1 Formal classification of MDO problems (breadth vs. depth of coupling)

Classification of MDO (system optimization) problems in terms of the coupling breadth and coupling strength yields useful perspective on the utility of optimization by decomposition. To see it, consider a system modeled by a number of modules, each module generating its own output Y from input Z, received from some other module. The purpose of such system optimization by decomposition is to make the dimensionally large optimization more tractable by breaking it into a set of smaller tasks, each associated with a single module, plus the coordination optimization at the system level to restore the temporarily suspended couplings. The coupling breadth and strength govern effectiveness of the decomposition; hence, we need to quantify these quantities as system characteristics by well-defined metrics.

The volume of data Z, i.e., the length of the vector Z_{AB}, where A and B designate receiver and sender modules, respectively, is the metric for the A-to-B coupling breadth. The derivatives $\partial Y_i / \partial Z_j$ serve as measures of the A-to-B coupling strength. Engineers often favor the logarithmic derivative format, $(\partial Y_i / Y_i) / (\partial Z_j / Z_j)$ as one that clearly shows how much Y_i changes per unit change of Z_j for better support of judgment involved in design decisions. Calculation of $\partial Y_i / \partial Z_j$ may be accomplished by any of the well-established sensitivity analysis techniques at the module (disciplinary) level. Because the derivatives form a matrix sized by the lengths of the vectors Y and Z, the A-to-B coupling strength is not expressible by a single measure, and there is no consensus how to reduce information embedded in the matrix $[\partial Y_i / \partial Z_j]$ to such a single measure. The average augmented with upper and lower bounds or the standard deviation is one of the obvious options. The concept of optimization by decomposition for a particular system may be examined in context of that system's coupling breadth and strength, the two metrics forming the horizontal and vertical coordinate axes, respectively, as illustrated qualitatively in Fig. 7.

The origin of that coordinate system corresponds to a system whose modules are totally uncoupled whose sensitivity analysis degenerates to a set of disjoint module (disciplinary) sensitivity analyses. That situation, that is seemingly happy because the system decomposition is given and requires no action, has the drawback of being completely devoid of any inter-modular synergy that may be beneficial when exploited intelligently by the designer. On the other hand, systems at the NE

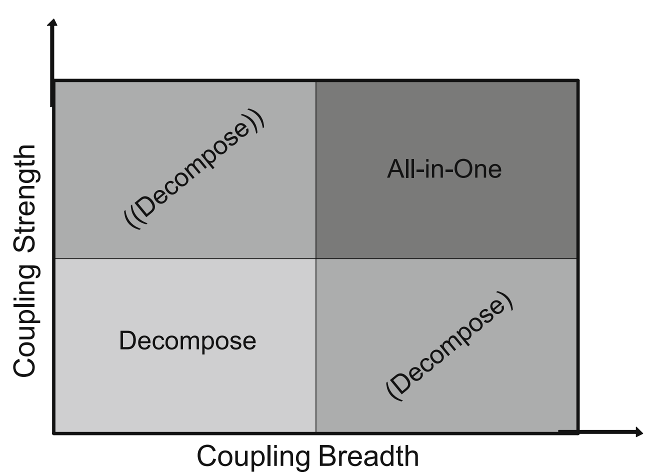

Fig. 7 Quantifying utility of decomposition in terms of coupling strength and breadth in the MDO process

corner of the square couple so widely and strongly that decomposition may not make the system optimization task any easier because dimensionality of the system-level coordination optimization may have dimensionality equal to or exceeding the optimization of the same system without decomposition. If modules A and B couple that way, combining them into a "supermodule" $C = A \cup B$ may be far better approach than decomposition. The "supermodule" C may correspond to a new discipline and aeroelasticity is a classic example of that. At the NW corner we have systems that couple strongly but narrowly. Optimization by decomposition is most necessary and useful in that area and techniques now available are most effective there. Finally, at the SE corner we deal with systems whose coupling is broad but weak. By sound engineering judgment such coupling breadth may be reduced by neglecting the selected small derivatives to make decomposition effective. Of course, opportunities to exploit synergy increase with the distance from the origin of the square in Fig. 7.

We tend to think of the sensitivity data (gradients) as search-guiding input to an optimization algorithm, but the same data may be directly useful in support of the designer's decision making, especially if properly visualized. In fact, these data may be regarded as a quantification of the time-honored notion of the trade-off. Having said that, one must not forget that in a non-linear problem the gradient values computed for an initial design may vary considerably as the optimization progresses toward a constrained minimum. So far, systematic studies to illustrate the extent of the above variability in representative applications have not been widely published and remain a potential issue for research. Research topics that arise in conjunction with the quantification of the system coupling breadth and strength include:

1. Establishing standards for presentation of the coupling strength and breadth for their best precision and utility, considering their use in formal optimization as well as in direct support of engineering judgment.
2. Quantification of the above metrics in a statistically meaningful sample of engineering systems, e.g., aircraft, automobile, ship, etc., to determine averages and bounds, and their distributions among typical engineering systems, i.e., mapping typical engineering systems on the diagram in Fig. 7. Further characterization of uncertainty propagation through the coupled systems. All of this information would help designers in exercising judgment in their applications.
3. Extend the above (#2) to illustrate how the sensitivities (gradients) vary in the passage from an initial to the optimal design.
4. Establishing guidelines for the cost and development of disciplinary math models that are best suited for application of MDO. These are not always necessarily the same as those needed for traditional, detailed disciplinary purposes.
5. Further development of the MDO algorithms highlighted in the Appendix of this paper.

Regarding the propagation of uncertainty mentioned in the second bullet above, one should acknowledge the increasing role that uncertainty in design has taken and its increasing acceptance by the engineering community. While not specifically discussed at the 2006 workshop, there is a broad list of references on the topic, a few of which are Oakley et al. (1998), Sues and Cesare (2000), Yang and Gu (2003), Du and Chen (2005), and Allaire and Willcox (2008).

4.1.2 Massively concurrent computing (MCC) in MDO

Technologies for executing many computing and data processing tasks simultaneously are rapidly become available to engineers. They come in a number of forms, e.g.

- Many processors in one unit. Two or four processors are offered in desktop computers and workstations.
- Clusters of inexpensive computers, each having access to a permanent memory of its own as well as to a mass storage shared with other computers. The number of computers in a cluster ranges into the thousands.
- Large number of processors in a single installation supported by local permanent memories and a shared mass storage. The number of processors exceeds now 100,000 and is likely to exceed 1,000,000 soon.
- All of the above embedded in networks extending continentally or globally.
- Reconfigurable computers of which the Field-programmable Gate Array (FPGA) is an example. Software provided with FPGA enables user to transform it into a multi-processor computer dynamically to fit the problem at hand.

The trend toward MCC is motivated by the ever increasing need for computing speed combined with gradual exhaustion of the single, silicon-based processor's potential for further increase of its processing rate. This is typically limited by heat generation and manufacturing cost of ever denser packaging of the transistors on the chip. It appears that single, silicon-based processor potential for a further speed-up is of the order of 1,000. Combining it with a million processors in a supercomputer indicates acceleration of the order of one billion relative to the computing speed typically available today. To put this in a perspective, computations that would require one month today (therefore never attempted) could be accomplished in milliseconds, i.e., practically instantaneously.

Computing at such speeds, if they actually could be attained, would support the designer in even the most complex tasks with the same effectiveness a word processor with spelling and grammar check aids a letter writer today. This would become "computing at the speed of thought," (Sobieszczanski-Sobieski and Storaasli 2004) to mean that design work would be paced by the engineer's ability to think, formulate 'what if' questions, and interpret the answers, rather than by waiting for the answers. There would be an enormous benefit to human creativity from uninterrupted train of thought and rapid rate of answers.

However, the preceding paragraph opens with a conditional clause because to utilize MCC technology the solution algorithm computational speed must be scalable with the number of processors. In engineering applications that involve large sets of coupled equations most of the present algorithms do not scale up that way and that impedes severely the user's acceptance of MCC technology. The other impediment stems from the variety of computer architectures, i.e., the ways in which processors communicate, the various levels of memories and mass storages, and I/O devices. These architectures tend to be so diverse that a particular problem solution running efficiently on one architecture may be unacceptably inefficient or not even run at all on another. This requires developers to tailor their solutions to the particular architecture at hand, thus driving up the cost.

In the long run, that means the existing solution algorithms in engineering physics will have to be revised and probably mostly replaced by new ones that are intrinsically parallel, therefore, naturally scalable. The cellular automata approach to fluid dynamics is one example of such replacement. In the short and interim outlook, coarse grained parallelism offers a way to utilize MCC technology in many applications immediately. In the coarse-grained mode, existing code is replicated over N processors unchanged, and executed with different inputs to carry out N tasks in the time of one.

Furthermore, the introduction of MCC as a new infrastructure serving engineers ought to change the 'utility metric' by which various analysis and optimization methods are evaluated and compared. Traditionally, numerical efficiency, roughly defined as the number of arithmetical operations to solve the problem, was such a metric. It was 'natural' when a single processor performed the operations but it needs to be replaced in a multiprocessor environment. A metric better suited to gauge a numerical method utility in that environment is the method's capability to distribute numerical operations for concurrent execution over as many processors as possible to compress the elapsed time needed to obtain a solution, even at the price of increasing the total number of operations in that compressed time.

Fortunately, MDO provides an abundance of opportunities for parallelism, e.g., finite difference techniques for sensitivity analysis, genetic algorithms for optimization, Design of Experiments applications, and various decomposition schemes. This enables MDO application to very large problems without waiting for the engineering solution codes to be rewritten. Research topics that arise in conjunction with MCC utilization in MDO:

1. Identify engineering analysis applications that require such excessive elapsed computing times that, despite the need, they are not attempted, e.g., full car crash simulation, unsteady aerodynamics with high temperature and high altitude chemistry effects.
2. Develop scalable solution algorithms for problems selected from the above set, e.g., Finite Element Analysis, or CFD.
3. Demonstrate a coarse-grained MDO application using more than 10,000 processors, e.g., a full optimization of an automobile including crash constraints.
4. Negotiate with the computer science community a single standard for presenting MCC machines to developers of the solution algorithms in engineering physics. Considering that the above developers are physicists and engineers, the purpose of the standard would be to hide the details and diversity of MCC architectures to enable creation of MCC codes of permanent value. It would then be a separate task for computer science specialists to implement such codes written against the standard on a particular architecture.
5. Numerical benchmarking and human-in-the-loop experiments to measure the effectiveness of MDO approaches and methods relative to current best practices that use optimization at the component level.

6. Development of surrogate models adaptable to the complexity of the design space to strike a proper balance between the cost, driven by the number of the DOE points, and the sparsity of the DOE coverage of the design space. This is in recognition of the danger that sparse coverage combined with nonlinearity of the model may leave important features of the model undetected, hiding between the sparsely located DOE points.

4.2 Vertical growth of MDO

4.2.1 Creativity, cognition and flexibility

Optimization, whether disciplinary or MDO, begins with laying down a concept that potentially may solve the design problem at hand. This initialization is a start, from which the design process begins moving towards a final design. Thus, the optimization state-of-the-art is limited to quantification of the initial design already established qualitatively by a human creativity process that, so far, is poorly understood and subject to research in the field of the cognition science. In optimization, the human creativity translates formally into a definition of the design space and its requirements, whose parameterization determines at the outset what the proposed design may potentially become and what it can never transform to. Thus the design space acts as an enabler and ultimate constraint at the same time (initialization of an aircraft configuration to a monoplane, including a variable for the wing position, produces a monoplane, the wing high or low or in a mid-location, but never a bi-plane). Likewise, mutations in Genetic Algorithms merely explore at random the established design space without transcending its definition, i.e., without adding any new coordinate axes.

Finally, traditional MDO assumes that a system such as an aircraft or automobile is design for fixed requirements. Some requirements, however, change over time and can involve significant redesigns at a later time. These redesigns can be expensive and move the system away from its optimum configuration. MDO should be applied to finding areas where flexibility can be embedded to optimize systems for a range of future scenarios (de Weck et al. 2004).

The obvious possibility of just adding new dimensions to the design space after the search has begun does not address the ultimate constraint issue because the meaning of any new design variable is defined by the way it enters the underlying mathematical model and must originate in the mind of the designer. Generalization of the very concept of the design space to enable qualitative transformation of that space by adding, redefining, and removing design variables simultaneously with the numerical search would simulate what to a competent designer comes naturally and apparently is one of the intrinsic capabilities of the human mind. Lifting the optimization to that level would advance it from its present role of a designer's aid to the role of a designer himself with profound consequences that are difficult even to imagine today. The fact that one sees only limited, if any, presentations at current MDO conferences alluding to the above as an issue is evidence that the optimization community has grown accustomed to considering optimization as merely a designer's aid, and, on the other hand, it attests to the difficulty of the problem.

The science of cognition that includes probing human inventiveness is still far from understanding the underlying mental processes. Opinions vary between extremes. One extreme view is that creativity is nothing more than a random juxtaposition of the concepts present in the immense base of common knowledge of facts and relationships accumulated in every person's mind. Its pole opposite asserts that the human mind is endowed with capabilities beyond what science can explain. One does not need to choose between the above opposites to recommend a pragmatic position that the movement of optimization in general, and MDO in particular, from being merely a designer's aid toward the ultimate new paradigm of simulating what a designer really does will require a collaboration with cognition science and its emerging companion science of complexity (Gero 2006). Research topics suggested by the above discussion:

1. Cognition and complexity sciences literature survey focused on determining whether any concepts have already emerged that may be ready to be exploited as connection points between the state-of-the-art of optimization and these sciences.
2. Investigate whether it is possible to generalize the concept of design mutation as practiced in GA to include the design space redefinition reflected in the underlying mathematical model.
3. Include the notion of time and future redesigns in design optimization in order to allow systems to be changed and reconfigured over time without moving too far away from their static optima.

4.2.2 Designing and co-optimizing families of products and Systems-of-Systems

Many systems are no longer designed as individual, isolated products, but as part of a larger product or system family. This requires including considerations of modularity and especially commonality in the MDO process. While much MDO work has been done in this area in academia, little of it gets used in industrial practice. This is mainly so because commonality decisions can only be made while taking into account their cost impact and the influence on customer demand, in addition to the traditional technical performance evaluation that MDO can provide (Willcox and Wakayama 2003).

Increasingly, vehicles operate as parts of larger networks and operational requirements include interoperability with other vehicles. Examples include the design of a sea base (naval logistics) among others (Wolf 2005). When applying MDO to System-of-Systems problems the performance of an individual vehicle becomes subjugated to the performance of the overall System-of-Systems (SoS) (Crossley and DeLaurentis 2006). In some cases, sub-optimality in one vehicle would be deliberately accepted for the benefit of the overall system. However, since the key feature of SoS is that individual vehicles can still perform useful functions independently (principle of functional independence Maier 1999) this is a challenging problem. In some cases existing bi-level methods might be extended to multiple levels, in other cases entirely new ways of thinking about the problem might have to be generated. Research questions in this area encompass:

Optimizing which design variables, components and modules to make common between variants of vehicles or products in a family of systems that is tailored to specific applications. While much work has already been done in this area, a stronger relationship between design variables and physical elements of form must be established and economic factors must be accounted for.

Systems-of-Systems are an emerging application for MDO where vehicles are treated as subsystems of a larger ensemble of systems and vehicles (Sobieszczanski-Sobieski and Storaasli 2004). One of the challenges of System-of-Systems optimization that must be addressed is that the boundaries and membership in the SoS can vary dynamically over time.

4.2.3 The need to integrate manufacturing into MDO

There are very few references alluding to the title subject. However, one example, presented by R. J. Yang of Ford Motor Co. at the 2006 European–U.S. MDO Colloquium, did so by showing the effect structural modification in the car body may have on both the car performance and its manufacturing cost, hence, indirectly on the return on investment. This case suggests application of MDO to the product life cycle with Return on Investment (ROI) or Net Present Value (NPV) as the objective, and encompassing both the vehicle physics and its manufacturing process.

The uniqueness of Ford's presentation at the colloquium is symptomatic of the prevailing practice whereby the design of the vehicle manufacturing process is to a large extent divided from the physics-based design of the vehicle to be manufactured, with only a limited amount of information crossing the divide. That divide impedes maximization of ROI (NPV) taken as a quantity representative of the highest-level objectives of a commercial enterprise. A passenger aircraft configuration design provides an example. Simplifying the example to make the point, consider the wing aspect ratio, AR, as a design variable typically considered in the physics-based vehicle design phase. That variable strongly affects the aircraft aerodynamic efficiency hence it impacts, indirectly, the aircraft productivity reflected in the cost per seat mile. In general, larger AR (subject to constraints, of course) reduces the seat mile cost potentially increasing the airline profit, consequently enabling the aircraft builder to sell the aircraft at a higher price. On the other hand, higher aspect ratio wing of the same total area is generally more expensive to make. The ROI depends, among other factors, on the trade-off between the selling price, operating cost and the manufacturing cost so it could be maximized by an MDO process spanning the physic-based vehicle design and the manufacturing and operations process design.

For such an MDO process to operate in context of the above-simplified example, the mathematical model must exist that relates the cost per seat mile to AR. Similarly, a mathematical model is needed to assess the manufacturing cost as a function of AR. Both models exist in the current practice and MDO tools are applicable on both sides of the physics/manufacturing divide. What has not become a part of the practice as yet is an MDO loop to manipulate AR as a variable active on both sides of the trade-off involving both models defined in the foregoing to reveal dependency of that tradeoff on AR. Knowing that dependency would enable design of the vehicle and its manufacturing process together toward maximum ROI or NPV. Research topics that arise in conjunction with unification of the physics-based vehicle design with design of its manufacturing process are as follows:

1. Select a design case of a vehicle in which manufacturing cost is sensitive to physical variables traditionally decided in the vehicle configuration design.
2. Identify mathematical models for the vehicle physics and for the vehicle manufacturing, both

models sharing as many design variables as possible.
3. Apply an MDO system optimization including the vehicle physics and manufacturing to demonstrate benefit from such unified optimization for maximizing a quantity such as ROI and NPV, contrasting the result with that obtained from disjoint treatment of the vehicle physics and manufacturing.

5 Summary and concluding remarks

The paper is based on observations from the 2006 European–U.S. Multidisciplinary Design Optimization Colloquium and outlines the authors' opinion of the state-of-the-art of Multidisciplinary Design Optimization and its industrial applications rooted in major previous developments, briefly reviewed for a historical context. The MDO theory and means of implementation are examined as constituents of the state-of-the-art, linked to principal commercial sources and providers. The paper points to the technology of Massively Concurrent Computing, and its distributed variant, now becoming ubiquitous as a key enabler that already begins to make possible what was impossible before and is certain to open new areas of applications in the future.

With the foregoing as a definition of a current MDO infrastructure, a number of industrial applications are brought up as lessons learned and indicators of limitations. Since MDO developments require a relatively high front-end cost and deferred benefit, these developments rely to a large extent on government sponsorships whose specific, recent, principal instances are discussed, with references to crucially important participation of the academic community in contributing research and education of future MDO practitioners and industry as a primary user and provider of challenges and case studies.

Future developments of MDO are projected in two orthogonal but mutually reinforcing directions: a horizontal growth for more capability, spanning an ever greater repertoire of applications using existing theory and tools, and a vertical growth attacking qualitatively new problems with innovative solutions to explore new, uncharted territory for potentially even greater benefits.

A key issue included in the vertical growth category is the present limitation of MDO in particular and optimization in general. Specifically, the confinement of the search for optimal design to the space qualitatively defined by the initial choice of the variable coordinate axes and problem parameterization. The paper speculates that breaking out of that restricting mold will require collaboration with cognitive science and studies of the inner workings of human mind and its intrinsic creativity, likely to extend far beyond MDO and its engineering origin.

The third direction of future MDO developments, regarded as a resultant of the above two, is a "dream trend" that combines the increasing power of the well-established approaches with the capability to solve problems that heretofore remain beyond the MDO theory and practice boundary.

Looking forward along the above axis, the paper recommends specific research topics to the attention of the research community, practitioners, and research and development sponsors. Definitions of these specific topics constitute the concrete, tangible product of the paper.

Acknowledgements We acknowledge the support of DLR, the German Aerospace Center, Goettingen, Germany, in organizing the colloquium, as well as the invaluable contributions of all those who attended. Additionally, special acknowledgement goes to Kumar Bhatia of the Boeing Company for his comments and discussion, especially in the development of Section 3.4.

Appendix

A MDO theory and frameworks

The issue of system decomposition began to arise over two decades ago, followed by the development of several formal MDO methods each characterized in large part by the ways and means for partitioning the system optimization into sub-tasks and representing the couplings (interactions) among the subtasks. It is largely these decomposition frameworks' approaches that set MDO apart from straight applications of NLP techniques in engineering design.

While numerous such methods have been introduced, evolved, tested, and applied (some successfully and some unsuccessfully) over the past two decades, here we discuss only three of the more mainstream developments and realize that there are others that merit consideration as well. This short discussion includes: Bi-Level Integrated System Synthesis (BLISS), referenced in Sobieszczanski-Sobieski et al. (2000) and Sobieszczanski-Sobieski et al. (2003); Collaborative Optimization (CO), referenced in Sobieski and Kroo (2000) and Braun (1996); and a more recent decomposition technique called Analytical Target Cascading (ATC), referenced in Kim et al. (2003) and Kim (2001). For the sake of completeness, Concurrent Subspace Optimization (CSSO) is also briefly reviewed below. A summary of various similarities and differences between methods is shown in Table 3.

Bi-Level Integrated System Synthesis has been under development at NASA Langley Research Center since 1996 and has undergone various changes over time. This algorithm uses an approximation technique to represent a range of optimized subsystem or subtask designs, upon which the system-level optimizer operates with the goal of minimizing an overall system objective function. Although the choice of the approximation technique is left up to the user, response surfaces and Kriging are the two most commonly used and tested. The connection between the system and subsystem levels is preserved through the formation of the subsystem objective function, $f = w_1 Y_1 + w_2 Y_2 + \ldots + w_i Y_i$. Here, a range of weighting coefficients (w_i) are appended to each output (Y_i) of each subsystem, included as independent variables in the formation of the approximation model, and then treated as design variables in the system level optimization. In this manner, the system level optimizer essentially varies the weighting coefficients to explore different design attributes imparted by the subsystem optimizations.

Whereas BLISS uses weighting coefficients to direct subsystem outputs such that they populate the entire design space before reaching the system level optimization, Collaborative Optimization uses target values of the design and state variables, created at the system level, to steer the outcome of subsystem optimizations in a more step-wise manner. CO was originally developed at Stanford University in 1996 and, like BLISS, has undergone various changes and improvements over the past decade. The most recent variant is found in Roth and Kroo (2008). The subsystems receive targets for interdisciplinary parameters from the system level, which they then try to match while converging to a feasible design. The typical subsystem objective function, $f = (X_{T,1} - X_{SS,1})^2 + (X_{T,2} - X_{SS,2})^2 + \ldots + (X_{T,i} - X_{SS,i})^2$, is formulated such the subsystem controlled parameters ($X_{SS,i}$) are allowed to deviate from the target value ($X_{T,i}$) in order to ensure a feasible design is met. Optimum values of the subsystem objective function are then returned to the system level where new targets are set and the process continues until convergence. Although the traditional CO algorithm does not use approximation models, response surfaces and other surrogate models have been used in several cases to enhance certain aspects of the method.

Analytical Target Cascading, which was initially developed at the University of Michigan in 2001, is unique in that it is not specifically intended as a competitor to BLISS or CO but offers a very flexible, multi-level, hierarchical approach to the decomposition problem into which other formal MDO methods may possibly be integrated. Although the mathematical formulations of ATC and CO are similar, ATC was designed for problems which are object or component aligned rather than discipline aligned. Design targets cascade down from the uppermost level, which may be the vehicle level or even the management/enterprise level, to each respective sublevel, typically defined by various physical components of the design. Adjacent sublevels then iterate successively to convergence in a pair-wise fashion until the entire system converges. Achieving convergence of the overall system is one of the key challenges in successfully implementing ATC. Several versions have been successfully implemented, the most recent of which is found in Tosserams et al. (2008).

Concurrent Subspace Optimization (CSSO) was initially developed at NASA Langley Research Center in 1988 and subsequently expanded in a series of papers from the University of Notre Dame Sobieszczanski-Sobieski and Haftka (1997) into several variants so diverse that they no longer constitute a single method. In the initial version, the subspace objective functions are formulated as the subspace contributions to the overall system objective. The subspaces are responsible for meeting local constraints as well as constraints from other subspaces, approximated through sensitivity derivatives obtained from the Global Sensitivity

Table 3 Overview of MDO decomposition frameworks

Method	BLISS	CO	ATC	CSSO
Trait				
System-level analysis required?	No	No	No	Yes
Subspace sensitivity analysis required?	No	No	No	Yes
# of levels	Two	Two	Multiple	Two
Partitioned by:	Disc. analys.	Disc. analys.	Object/component	Disc. analys.
Subspace optimization influenced by targets?	Yes, indirectly[a]	Yes	Yes	No
Autonomous subspace optimizations?	Yes	Yes	Yes	Yes

[a]Of interesting note is that the BLISS subspace objective function could also be formulated as that of CO and ATC with the target value taking the role of the weighting coefficient. However, in the BLISS method, slow convergence and possible non-feasibility of the returned solution have led to the use of the weighting coefficients as the preferred alternative

Table 4 Overview of commercially available software tools in support of MDO

General purpose tools	Embedded optimization	Process integration tools
Excel (Microsoft)	SolidWorks-Cosmos	ModelCenter, CenterLink (Phoenix Int.)
Matlab (Mathworks)	GENESIS (VRD)	iSIGHT, FIPER (Engineous)
Mathematica (Wolfram)		modeFRONTIER(Esteco)

Equations (GSE). Variants include the use of neural nets, response surfaces, variable sharing between subspaces, and numerous other refinements.

Software tools and MDO providers

Software tools now available commercially to support multidisciplinary design optimization have evolved over the last two decades and can be classified into three broad categories:

(a) general purpose optimization tools,
(b) embedded optimization tools, and
(c) dedicated process integration and design optimization (PIDO) tools.

General purpose tools (Excel, Matlab, and Mathematica) are general software environments and programs that can capture both the model of the system that is to be optimized, as well as the optimizer in the same environment. Typically, these tools are easy to learn and use, but their computational performance tends to break down for large problems. Also, these tools do not easily integrate high-performance engineering codes (such as CFD, FEM etc…). For educational purposes, however, the general purpose tools are the preferred choice due to their simplicity. The embedded optimization tools are a recent trend, where existing engineering design and analysis tools (e.g. CAD Solidworks) are starting to incorporate within them an embedded optimization capability. While useful for local optimization, this capability typically only benefits one engineering discipline at a time. Finally, a small industry has recently started to emerge in order to provide a process integration and design optimization capability (PIDO). The focus is on integrating a firm's natural design processes with multidisciplinary design optimization. Popular tools operate at the individual desktop level (ModelCenter, iSIGHT) or more recently at the team or even at the enterprise level (CenterLink, FIPER). Here the focus is on data management, configuration control, workflow management and ease of integration of disparate analysis codes.

Increasingly, firms are migrating from in-house custom-made solutions to commercially available tools such as the ones listed in Table 4.

References

Alexandrov N (2005) Editorial—multidisciplinary design optimization. Optim Eng 6:5–7

Allaire D, Willcox K (2008) Surrogate modeling for uncertainty assessment with application to aviation environmental system models. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper no 2008–5890. Victoria, B.C

Balling RJ, Sobieszczanski-Sobieski J (1996) Optimization of coupled systems: a critical overview of approaches. AIAA J 34(1):6–17

Bartholomew P (1998) The role of MDO within aerospace design and progress towards an MDO capability. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, AIAA paper no 98–4705. St. Louis, MO

Barthelemy J, Haftka RT (1993) Approximation concepts for optimum structural design—a review. Struct Optim 5(3): 129–144

Braun, R (1996) Collaborative optimization: an architecture for large-scale distributed design. PhD thesis. Dept. of Aeronautics and Astronautics, Stanford University

Crossley WA, DeLaurentis DA (2006) Methods for designing, planning and operating systems of systems. In: Workshop proceedings, Purdue University, AFOSR, 17–18 May

de Weck OL, Jones MB (2006) Isoperformance: analysis and design of complex systems with desired outcomes. Syst Eng 9(1):45–61

de Weck OL, de Neufville R, Chaize M (2004) Staged deployment of communications satellite constellations in low earth orbit. J Aero Comput Inform Comm 1(3):119–136

Du X, Chen W (2005) Collaborative reliability analysis under the framework of multidisciplinary systems design—special issue on multidisciplinary design optimization. Opt Eng 6(1):63–84

Fulton RE, Sobieszczanski-Sobieski J, Storaasli OJ, Landrum E, Loendorf D (1974) Application of computer-aided aircraft design in a multidisciplinary environment. SYNOPTIC J Aircr 11(7):369–370

Gero JS (ed) (2006) Design computing and cognition '06, p 713. Springer.

Giesing J, Barthelemy JM (1998) Summary of industry MDO applications and needs. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, AIAA paper no 98–4737. St. Louis, MO

Kim HM (2001) Target cascading in optimal system design. PhD dissertation, Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan

Kim HM, Michelena NF, Papalambros PY, Jiang T (2003) Target cascading in optimal system design. Trans ASME J Mech Des 125(3):474–480

Maier MW (1999) Architecting principles for systems-of-systems. Syst Eng 4(1):267–284

Messac A (1996) Physical programming: effective optimization for computational design. AIAA J 34(1):149–158

Oakley DR, Sues RH, Rhodes GS (1998) Performance optimization of multidisciplinary mechanical systems subject to uncertainty. Probab Eng Mech 13(1):15–26

Papalambros PY, Wilde DJ (2000) Principles of optimal design—modeling and computation, 2nd edn. Cambridge University Press, Cambridge ISBN 0 521 62727 3

Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. Prog Aeronaut Sci 41:1–28

Ravachol M (2006) Multilevel multidisciplinary optimization of aircraft. In: 2006 European–U.S. MDO colloquium proceedings, German Aerospace Center, Goettingen, Germany, 17–19 May

Roth B, Kroo I (2008) Enhanced collaborative optimization: application to an analytic test problem and aircraft design. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper no 2008–5841. Victoria, B.C

Schmit LA (1971) Structural synthesis 1959–1969: a decade of progress. In: Gallagher RH, Yamada Y, Oden JT (eds) Recent advances in matrix methods of structural analysis and design, University of Alabama Press, Huntsville

Simpson T, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang RJ (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Multidisc Optim 27(5):302–313

Smith RP, Eppinger SD (1997) Identifying controlling features of engineering design iteration. Manag Sci 43(3):276–293

Sobieski IP, Kroo IM (2000) Collaborative optimization using response surface estimation. AIAA J 38(10):1931–1938

Sobieszczanski-Sobieski J (1990) On the sensitivity of complex, internally coupled systems. AIAA J 28(1):153–160

Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace optimization: survey of recent developments. Struct Optim 14(1):1–23

Sobieszczanski-Sobieski J, Storaasli O (2004) Computing at the speed of thought. Aerosp Am 42(10):35–38

Sobieszczanski-Sobieski J, Barthelemy JF, Riley KM (1982) Sensitivity of optimum solutions to problem parameters. AIAA J 20:1291–1299

Sobieszczanski-Sobieski J, Agte JS, Sandusky RR (2000) Bilevel integrated system synthesis. AIAA J 38(1):164–172

Sobieszczanski-Sobieski J, Altus T, Phillips M, Sandusky R (2003) Bilevel integrated system synthesis for concurrent and distributed processing (BLISS 2000). AIAA J 38(1):164–172

Sues R, Cesare M (2000) An innovative framework for reliability-based MDO. In: 41st AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference and exhibit, AIAA paper no 2000–1509. Atlanta, GA

Tedford NP, Martins J (2006) On the common structure of MDO problems: a comparison of architectures. In: 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper no 2006–7080. Portsmouth, VA

Tosserams S, Kokkolaras M, Etman LFP, Rooda JE (2008) Extension of analytical target cascading using augmented Lagrangian coordination for multidisciplinary design optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper no 2008-5843. Victoria, B.C

Van Keulen F, Haftka RT, Kim NH (2005) Review of options for structural design sensitivity analysis. Part 1: linear systems. Comput Methods Appl Mech Eng 194(30–33):3213–3243

Vanderplaats GN (1976) Automated optimization techniques for aircraft synthesis. In: Proceedings, AIAA aircraft systems and technology meeting, AIAA paper no 76–909. Dallas, TX

Willcox K, Peraire J (2002) Balanced model reduction via the proper orthogonal decomposition. AIAA J 40(11):2323–2330

Willcox K, Wakayama S (2003) Simultaneous optimization of a multiple-aircraft family. AIAA J Aircraft 40(4):616–622

Wolf R (2005) Multiobjective collaborative optimization of systems of systems. SM thesis, Engineering Systems Division, Massachusetts Institute of Technology

Yang RJ, Gu L (2003) Experience with approximate reliability-based optimization methods. Struct Multidisc Optim 26(1–2):152–159