31st Congress of the International Council of the Aeronautical Sciences
Belo Horizonte, Brazil; September 09-14, 2018

MDO AS A SOCIO-TECHNICAL SYSTEM – A SMALL CONTRIBUTION TO COLLABORATIVE ENGINEERING

Martin Spieck*, Benedikt Aigner, Prajwal Shiva Prakasha***, Pier Davide Ciampa*****
***Thelsys GmbH, **RWTH Aachen, ***German Aerospace Center**

Keywords: *socio-technical systems, complex systems, MDO, collaborative aircraft design*

Abstract

Multidisciplinary Design Optimization (MDO) is a methodology for the design of complex engineering systems that can largely be automated within a formal framework. Thus, research in MDO mainly concentrates on improvements to the engineering software, the MDO framework and the solution sequences for the optimization problem.
In practical applications, however, engineers have to set up the MDO problem, they have to supervise the execution and interpret the results, and they have to present their findings to the decision-makers. Thus, MDO is not only about the synergies from "connecting computers", it is also about connecting the humans behind the computer screens.
As a consequence, MDO is not just an automated IT framework, it is a socio-technical, "human-in-the-loop" system. This paper provides an overview of barriers and bottlenecks for MDO that are driven, or at least severely affected by socio-technical factors, and propose practices to overcome these barriers and to increase the efficiency and effectiveness of ambitious MDO applications in research and industry.

1. Introduction

Multidisciplinary Design Optimization (MDO) is, quite simplistically, design-by-computer.

In theory, an MDO design process has to be set up once in a manner appropriate to a certain range of design problems, and should then be able to deliver an optimal overall solution for any appropriate design task by simply pressing a button.

In practice, this does not work for applications which go beyond simple academic test cases. MDO requires the support of the disciplinary specialists and their management, a good oversight of the process itself, of the capabilities and limitations of the disciplinary tools, and excellent engineering knowledge to interpret the results and develop them further.

And this is the point at which the socio-technical factor comes into play.

Socio-technical systems design is an approach of systems design that accounts for the interaction between people and technology.

Socio-technical systems have long been researched, particularly in the context of the organizational and human interface with the technical system during operation, e.g. in the fields of human-computer-interaction and cognitive ergonomics. Recently, the challenges of addressing increasing system interdependencies in aircraft development while design teams continue to grow in size and intricacy has stipulated research on various aspects of this topic, e.g. knowledge management, [1], communication, [2], cross-disciplinary interaction and integration, [3], or university education, [4]. MDO has been identified as a possible solution, but it is also acknowledged that so far, research and development in MDO has not "focused on the organization or individual engineers involved in executing the design", [5].

In this paper, the authors want to make a contribution to this topic, concentrating on the "human factors" that influence the highly technical system of multidisciplinary aircraft design and optimization, and to propose measures that may improve the performance and impact of MDO in real-world applications.

The stimulus to cover this issue systematically derives from the EU-funded project AGILE[1]. The objective of the AGILE project is to implement the 3rd generation of MDO through efficient international multi-site collaboration in overall design teams. Obviously, the efficiency and effectiveness of these design teams does not solely depend on the possible performance of the distributed analysis and optimization frameworks, but also on the level of integration and smooth cooperation between the scientists and engineers of the teams. AGILE allowed to study the socio-technical issues and their impact on MDO applications in an ambitious but autonomous setting.

It is important to note that this paper draws on the experiences of numerous projects spanning more than 20 years. The presented barriers, bottlenecks and other complications are **not** a summary of shortcomings observed in AGILE! On the contrary, this project allowed to study MDO-driven collaborative teamwork in detail, consult experts from research and industry about their experiences, and perform small tests to highlight some typical effects of collaboration. The authors are deeply indebted to all AGILE members for their kind support, assistance and contributions to this topic.

2. MDO as a Complex Socio-Technical System

Describing MDO as a socio-technical systems recognizes the interaction between the computational MDO process and the professionals who design, launch and monitor the multidisciplinary optimization cycle and interpret the results.

While the technological performance of MDO is improved continuously by researchers, scientists, engineers and programmers, the real performance in practical applications is largely determined by the interaction between the computer framework and the human design team.

To be effective, a software tools needs a good user interface (⇨ergonomics) and a well-trained operator (⇨skills). In principle, this also applies to MDO. But instead of a stand-alone computer program, MDO has an interwoven and delicate framework of heterogeneous software tools and data bases, and instead of a single operator there are diverse, distributed teams of experts. Consequently, there are no easy answers how to improve the ergonomics of an MDO application and the overall skills and efficiency of an MDO design team.

There are two major factors of influence which determine the degree of difficulty in this respect: the complexity of the system and the time constraints of the assignment.

2.1. Complexity

A complex system is "a system with numerous components and interconnections, interactions or interdependencies that are difficult to describe, understand, predict, manage, design, and/or change", [6].

This working definition of MIT already indicates the difficulties of dealing with such a system. *Complex* in this context should not be confused with *complicated*. The degree of complexity increases with the level of attributes (in MDO for example the number of systems and/or design variables) and at the same time the amount of interdependencies. Complexity, however, is not an objectively measurable value. Besides the mere number of variables and connections, it is also influenced by the type and quality of interdependencies.

In aircraft design, the inherent complexity of an MDO challenge represents already a significant technological challenge. Additionally, the human brain is not well equipped to deal with intransparent, interdependent nonlinear interrelations, [7]-[10]. Our evolutionary development has focused on mastering direct cause-and-effect interrelations. Evolution has also equipped man with a powerful "tool" for dealing with complex situations: pattern recognition. As an example, every average driver is able to manage busy urban traffic - still a major challenge for even the most advanced driverless vehicles. But pattern recognition only works in environments which

[1] AGILE: Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts; Project reference: 636202, funded under: H2020-EU.3.4. Duration: 1st June 2015 - 30th November 2018

can be trained (and possess a high factor of recognition), and in which most interrelations are observable and (at least piecewise) linearizable. Most MDO challenges do not belong to this class of complex problems. As a consequence, the complex socio-technical system of MDO-aided aircraft design is not only a technical, but also a human challenge.

2.2. Time Constraints

A design exercise usually has to meet a specific deadline. Even with vast computational power and unlimited resources, it will not be possible to solve a complex, integrated design problem in detail. At some point in time, there will be the “moment of truth”, at which analysis and optimization have to stop and a design decision has to be made, based on the results achieved so far.

But even if the task was not time-critical at the beginning, time is an issue. We already established that complex situations are difficult to solve. In consequence, most MDO projects will experience mounting time pressure, even without any specific deadline. In a design exercise, we are dealing with a dynamic entity: the situation and the influencing boundary conditions (technical, organizational, etc.) will progress and change, with or without own involvement.

And time pressure introduces an additional stress factor which robs the individual and the team of tranquility, oversight and with those the ability to handle complex systems well… which already touches a common and well-known peculiarity of human behavior in socio-technical systems, *stress-induced cognitive decline*. The next paragraph will highlight typical mannerisms which can significantly impair the efficiency of humans-in-the-loop and consequently also the effectiveness of the entire MDO application.

3. Human Behavior in Complex Systems

3.1. Brief History

Human behavior in complex situations belongs to cognitive science, a field of psychology that gained credence in the 1950s. The interest in the specific topic of complexity started about 1975, when the limits of deterministic control of homeostatic systems became apparent. The 1980ies then saw the beginning of systematic studies in controlled test environments, [11], and formed the basis for today’s models of human complexity processing and management. The findings were then transferred to many applications, from the ergonomics of man-machine-interfaces to executive management, control of supply chains and software programming. Currently, with the new challenges of digitalization and autopiloted vehicles, a positive upbeat mood for more fundamental research can be observed.

3.2. Individual Bias

Individual bias refers to effects that can be observed when a person is interacting with a complex system, [12]. Examples are subconscious coping or avoidance strategies, systematic errors of prediction and estimation, or the various shapes of human stress behavior in complex and intransparent situations.

3.3. Domains of Typical Human Behavior in Complex Situations

The iceberg model is a popular visualization of a context with hidden features.

Our iceberg indicates the main domains we found in human behavior in complex technical

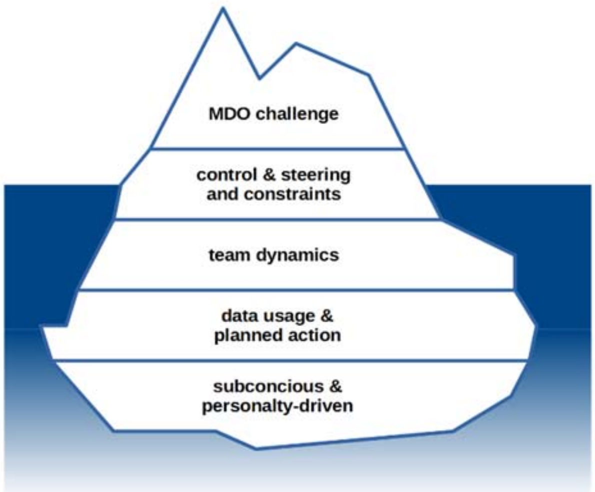

Fig. 1: Human behavior in complex situations

situations. These domains are addressed consecutively in the following paragraphs.

3.4. Subconscious and Personality-driven Effects

This section comprises some of the most typical, least controlled (and controllable?) human behavior. Most effects mentioned below are not specific to MDO, but they can have a great impact on the outcome of an MDO application. *Psychology* is often regarded as dealing with the subconscious, somewhat Freudian action and reaction schemes, whereas the following paragraph highlights some typical human behavior which roots deeply in personal values and beliefs which have been learned, experienced or trained – often early in life.

3.4.1. Applied psychology

Humans, and professionals in particular, want to feel competent to a certain extent when they are taking action. *Encapsulation* is a behavior pattern where the person carves out a part of the problem he or she is confident to solve. All attention and energy is directed to this part, while the rest of the problem is omitted. This effect is closely related to *protection of self-perception*, which subsumes a variety of subconscious strategies to maintain the illusion of competence and control, from belittling problems to complete detachment from reality.

Then, there are various variants and effects of overestimating one's own influence and competence, which are subsumed under the common human trait of *control illusion*. A trivial example is the exaggerated belief of many road users into their driving skills, [13].

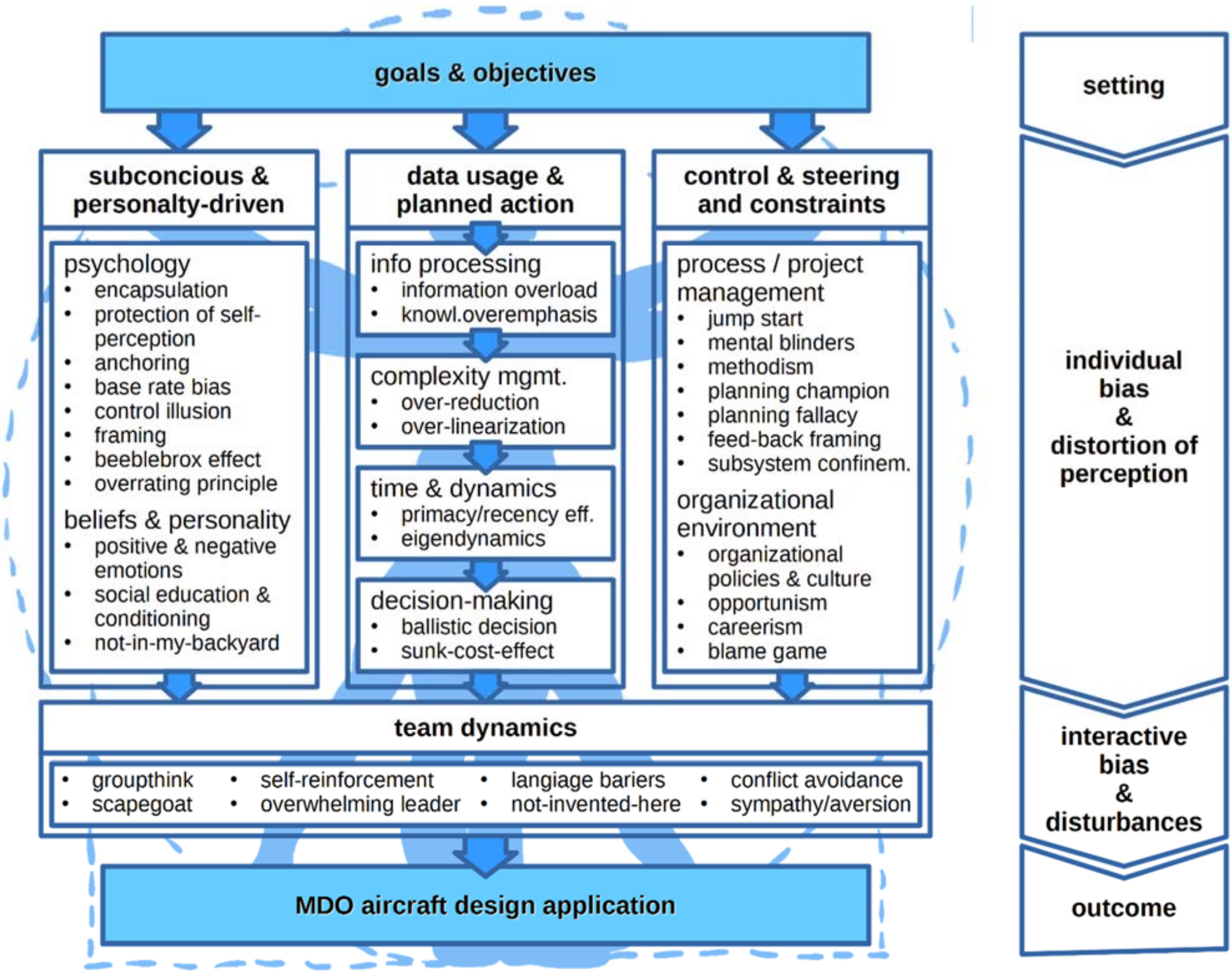

Fig. 2: Individual and interactive bias and distortions of perception in complex situations

The *beeblebrox effect*[2] is a more general attitude. It denotes the human self-conception to put oneself at the center – a view that may be impaired by one's position and impact in a complex multidisciplinary optimization process.

Anchoring is a bias for an individual to rely too heavily on an initial piece of information (the "anchor"). Its power is amplified if this piece of information was gained by substantial efforts early in a design campaign.

The general definition of *base rate bias* is that base rate information, i.e. generic and/or general information, is neglected in favor of more specific (e.g. personalized or illustrative) information. This bias is often mentioned in the context of statistical data versus gut feeling, but the principle applies to complex engineering information as well.

The *framing effect* refers to the fact that people react differently in a similar situation, depending on whether they perceive they are achieving a gain or avoiding a loss. Although this bias is usually attributed to decision-making, it plays an important role when setting up an MDO exercise – starting with the very simple question whether to minimize a function or to maximize its inverse, which alone can make a difference in how the optimization problem is perceived.

The *principle of overrating* means that one objective of a multi-criteria domain receives preferential treatment. Overrating occurs frequently in a complex context with negative concatenations, such as prevention targets which are neither easy to define nor attractive to achieve. Typical preferences are personal predilections (e.g. disciplinary imbalance), avoidance of problems or conflicts, mismatched stimuli or penalizing, ignorance of context and correlations, or lack of top-level overview.

3.4.2. Beliefs and Personal Traits

The importance and impact of *positive and negative emotions* is widely recognized in human life. But it also has an important impact on the strictly logical and objectifiable process of MDO. Just think of the difference it will make if an engineer is asked for an urgent contribution when (a) he or she wants to leave early for a family evening, or (b) this assignment offers the opportunity to skip a boring two-hours meeting.

Social education and conditioning determine our values, beliefs and behavior patterns. They are at the root of many of our decisions – for example why we chose the profession of an engineer or scientist and not that of a surfing instructor at Barbados. The same preferences and doctrines which made us disdain the sun and beaches of the Caribbean also determine our behavior in an interwoven, multilateral field like MDO. How do we cope with other opinions, uncertainty, hierarchies, frustration, responsibility? It is all rolled into the pure technology-oriented challenge of aircraft design.

The *not-in-my-backyard syndrome* is a well-known example of a specific individual behavior pattern. It is mentioned here because it not only affects a multidisciplinary analysis (it does so in many other interactive exercises, too, no matter if they are technical, organizational or social). It is mentioned because MDO is one of the few possibilities to demonstrate and even quantify the impact of this syndrome, and thus a great opportunity to challenge and possibly overcome this impeding behavior.

3.5. Data Usage and Planned Action

There are typical patterns and pitfalls when human mind is dealing with abstract, complex and abstruse data. This involves the intellectual processing of this data on one hand, and the consequential actions building on the mental analysis.

3.5.1. Information processing

Information overload, i.e. too much information in too little time, is a problem everyone has already experienced. Contrary to computers, there is no clear signal like "overflow error". The greatest danger in engineering is *creeping information overload*. Humans are able

[2] The name refers to Douglas Adams entertaining novel „The Restaurant at the End of the Universe“. Readers will remember that the book's character Zaphood Beeblebrox is assured about his own universal importance in the tower of the "Total Perspective Vortex". In this context, it means that scientists and engineers prefer to see their individual domain, and their personal contribution to it, as the most important element of the "MDO universe".

to still "function" in overload mode for a long time. This has significant side effects, like a rapidly growing error rate, stress symptoms and personality changes such as depression or irritability. The situation can be compared to a juggler who adds one egg after another to all those he already has in the air. The audience may still clap and cheer about the astonishing performance as his movements are becoming more and more hectic and erratic, but it will not end well. (Parallels to design exercises are obvious.)

Overemphasis of knowledge is an effect which easily befalls highly-skilled, experienced professionals. New results are sorted into the existing knowledge patterns, leading to marginalization and adaptation of facts if the novel information does not fit. This limits the possible achievement to incremental "more-of-the-same" improvements of existing solutions and involves the danger of overseeing opportunities and risks.

3.5.2. Complexity management

Conceptual models are representations of the actual, real world. They are essential for analyzing and predicting the behavior of a complex system – no matter if the analyzing unit is a computer or a human brain. *Over-reduction* means that important aspects of the system were omitted in the modelling. This does not necessarily mean that the model is too simple; in fact, it can be very detailed and complicated. It means that important aspects of the complex system are not accounted for. A typical cause is thinking (and acting) in direct causal links instead of an interrelated causal network.

Humans are acting in a linearized image of reality that is updated at high frequency. We are having problems in dealing with exponential relations: kids are drawing the trajectory of a ball as a straight line, drivers are underestimating the braking distance at higher speeds. *Over-linearization* is the equivalent scheme in analyzing and designing a technical system. Mathematically, professionals are well equipped to cope with nonlinearity. But after the computation, the results have to be interpreted, interrelated and weighted. This is the point where this bias can strike again.

3.5.3. Time dependency and dynamics

The *primacy effect* and its counterpart, the *recency effect*, refer to human traits to weigh alternative solutions. The *primacy effect* causes people to overstate the first information, or solution, presented to them; the *recency effect* overstates the last. Both are well-proven psychological effects, but we are not aware of a study explaining which effect will dominate under which circumstances. To our observation of technical collaborations, the *primacy effect* will be more frequent when people are new to the topic, the *recency effect* rules when people were bored of the presentation or meeting.

It should be added that the *over-linearization* symptom is even more powerful when dealing with nonlinearities in time. Humans are notoriously bad at prediction the progression of exponential growth over time, even when the growth factor is small.

A system with *eigendynamics* will change its behavior over time without external input or changes to the system itself. *Eigendynamics* and other *time variances* make it very difficult to predict the status of a system at a point in time by computational analysis, and almost impossible by human estimation. Now, fortunately a technical design itself is not subject to eigendynamics[3]: the performance of a design solution does not depend on the point in time it was designed. But the design process does! In a collaborative environment, many workflows progress in parallel, which changes frequently boundary conditions, constraints and available design space.

3.5.4. Decision-Making

The *ballistic decision principle* could also be termed as fire-and-forget decision making. Once a decision has been made, the topic is ticked off and a new one is tackled. There is no follow-up, no monitoring to see if things are working or developing as intended.

[3] This is not to be confused with the engineered system's own behavior, which can very well develop eigen-dynamics, e.g. resonances, vibrations or flutter.

The *sunk-cost-effect* makes people continuing with a task that has already failed, is becoming obsolete or outdated, or should be replaced by a more simple, better approach. The degree to which this effect is effective depends to a large extent to the effort, time, budget and emotional(!) investment that already went into the task. It can be amplified if the change could be interpreted as a failure, or if somebody would "lose his/her face".

3.6. Control & Steering and External Constraints

This section covers the actual "handling" of the complex system and the factors that impair or boost people's ability to do so. For this paper, we concentrate on typical bottlenecks, bad practices and behavior traits with respect to the management and control of the design activities, and the major external influences on managers, scientists and engineers to perform well.

3.6.1. Process / Project Management

Jump start is a behavior pattern often detected in hindsight. Starting dynamically and progressing rapidly is a good thing – unless fundamental issues are not addressed at the beginning, leading to mismatching work, bad results, unclear responsibilities, and a multitude of additional possible problems later in the project.

Mental blinders is one of the terms which describe a too narrow, often disciplinary or single-technology focused view. It leads to conflicts among the design team, unbalanced designs of the overall system or its subsystems, and unpredicted interactions and interferences. Mental blinders also prohibit solutions which use existing technologies in a novel, perhaps unconventional way. The effect is comparable to *tunnel vision*, but it is attitude-driven instead of being a result of high workload, stress and pressure like the latter.

Methodism is a pattern that is "popular" with two different groups of professionals: young and inexperienced designers who are finding security in blindly following a methodological recipe, and senior professional who "always did it this way". Both approaches have the inherent risk of making the problem fit the methodology, instead of selecting the right methods for the job at hand.

Planning champions are no real champions. Instead, they spend all time and effort on drawing the most bold and detailed of plans, thus avoiding the challenge of putting it into action. Some experienced individuals of this species deliberately take on monumental tasks and strive for promotion before they have to deliver.

Planning fallacy is quite different. It describes the firm belief that every detail of an undertaking can be planned and accounted for beforehand. While apparently most engineers do not fall into this category as far as project management is concerned, there is nevertheless a significant number of them around (many of them young, high-performing professionals) who are overly optimistic when analyzing a complex technical system.

The *principle of subsystem confinement* uses the decomposition of the complex system to "phase out" inconvenient problems, either to other teams or team members, or to simply lose sight of the annoying issue.

Feed-back framing describes the tendency to selectively seek for and respond to feed-back. Negative feed-back is often not heard, not accepted as valid or its meaning is twisted or redefined until the message is "acceptable" for the receiver. Positive feed-back, however, is easier to embrace, even if the message was not as enthusiastic as it seems at the first glance. This effect often comes in combination with the human trait to hide criticism behind a compliment: "That was quite good, but…". The feed-backee only hears the initial praise and forgets about the unpleasant things after "but".

3.6.2. Organizational Environment

Organizational policies and culture are arguably the most important boundary conditions for MDO and collaborative engineering in general. The usual hierarchical structures, business processes and daily practices in a large organization do not reward interdisciplinary work. Engineers are paid to do "their work", not to "make life easier for someone in another department". Additionally, the benefits of collaboration are difficult to demonstrate to a manager or controller: it is often not possible to prove in hard numbers how better the overall design performs or how many errors and redesign work have been avoided, compensating by far the time and costs invested into an integrated interdisciplinary approach.

These systematic bottlenecks are of course amplified by the existing, often deeply-rooted company culture. In an organization with a culture of open communication and trust, many systematic and structural deficiencies can be overcome. A close collaboration, on the other hand, can hardly be established in a company where the necessary values are not lived on the floors of the engineering departments. This applies to an even larger extent to collaboration between independent organizations.

Egoistical *careerism* is of course a major obstacle in each collaborative undertaking. Already a small number of individuals playing a self-centered "zero-sum-game" in a joint exercise can jeopardize the entire project. But MDO faces an additional, system-inherent challenge: middle managers feeling that MDO threatens their future career. For them, applying MDO equals less budget and less people for their unit, thus less status. They also fear diminished influence on the design while still having full responsibility for errors. Overall, MDO in their book is obstructive for their personal advancement and has to be thwarted or at least circumvented.

Opportunism is a behavior pattern with interconnections to *careerism* (see previous) and *conflict avoidance* (see 3.7). An individual trait in nature, it is nevertheless listed in this section because on one hand it is a deep-rooted behavior pattern whose impact can only be minimized by adequate leadership, and on the other hand it is strongly influenced by external factors such as the working environment and company culture.

Blame game is a very popular pastime in some organizations. It is also time-consuming, opposed to progressing professionally, and toxic if a certain limit is exceeded. Relying on the same mechanisms as the *scapegoat* (see 3.7), the *blame game* is rather an organizational than an individual trait, driven by the practices and behavior patterns of the entire ecosystem "organization".

3.7. Team dynamics

Groupthink is a well-known psychological phenomenon: a design team in which the desire for harmony or conformity in the group results in pursuing or supporting irrational or dysfunctional decisions and solutions. However, nearly 50 years after postulating the principle, [14], it is still very "popular" with groups, especially those which are embedded in hierarchical systems or exposed to great stress.

Like the biblical animal which is ritually burdened with the sins of others, *scapegoats* have to carry the blame of a failure or a mistake. A scapegoat can be a person, a piece of equipment or an abstract object like a process or the famous "necessities" which "left no other option". In a similar function, the *lightning rod principle* looks for relief from emotional stress. Both are variants of the more general *black sheep effect*, [15].

Positive or negative self-reinforcements have a dynamic, self-driving effect on teams. Every professional has hopefully already experienced the energy in a vibrant team that rushes from success to success. And most of us know the burden of working in a team that started out slow and dissipates from there. Both are self-reinforcing processes which, once initiated, can continue to spiral for a long time.

The *overwhelming leader* dominates the collaborative work of experts who, for a balanced approach, should cooperate on equal terms. The motivations of those "alphas" can vary: the overwhelming leader could egoistically strive for power or respect, or it may be the attempt to get a lazy bunch moving. The other team members, on the other hand, could be insecure, unmotivated, overawed or frightened, thus supporting or even provoking this behavior.

Language barriers are more subtle than most people recognize. A test in an AGILE technical meeting revealed that after two years of intensive collaborative research, some ambiguity still exists in a design team of experienced scientists and engineers. In the test, the team members were asked to rank 20 sentences addressing the same relation, but in varying degrees of certainty. The figure shows the distribution matrix, ranking the 20 statements vertically according to their average level of certainty and listing on each row the number of individual assessments for that position. The main diagonal is dominant, which overall demonstrates a good common understanding, but

2	5	6	2	6	1			1			1								
7	7	2		2		2	1		2								1		
8	3	1	3		2	2	1		1	1						1			1
3	2	2	2	5	2	2	4					1			1				
1	1	2	5	2	4	2	3	1	1				1						1
	4	3	1	2	3	4	2		2	2									1
1		3	5	3	3	2	2			1	2			1		1			
		1	3	2	4	2	5	1	3		1		1					1	
		1	1	3	1	1	1	6	1	4		1		2	2				
	1	1			1	4	1	5		2	1	1	3	1	2	1			
			1	1	1	1	2	2	4	6			1	3		1	1		
					3		1	3			2	2	1	2	1	2	2	2	3
						1		1	2		5	1	2	4	4	2	1		1
						1	1	1	2	1	2	2	1	5		1	5	1	1
									2	2	2	5	2		5	1	2		3
		1					1		4		2	1	1	3	1	1	1	3	5
							1	1	3			4	1		1	3	3	4	3
				1				1		2		2	1	1	2	2	4	3	5
							1			1	2	1	2			6	4	3	4
											2		3	4	2	1	2	6	4

Fig. 3: Distribution matrix of language comprehension

we can still observe multiple outliers, some of which even reverse the opinion of the majority. The results of other, more diverse teams were even more scattered.

The *not-invented-here syndrome* is the refusal to think outside the team's comfort zone or to accept solutions or support which stem from outside the team's unwritten boundaries (which can be the team, the department, the organization, etc.). It is also fatal to any kind of mutual and productive collaboration.

Successful collaborative teamwork thrives on different views and varying opinions. The pattern of *conflict avoidance* takes the edge off of one of the most effective drivers for excellence and performance: polite professional dissent. Polite is added here because the cure from *conflict avoidance* is not rude behavior and heavy dispute, but respectful technical discussion.

Sympathy / aversion is a very obvious impact factor for each team. Professional team members do not have to like another to collaborate effectively, but a certain degree of sympathy and connection surely supports good teamwork and good results. This is not true for the opposite: professionalism can surely help to be more tolerant and robust, but there are limits to the level of aversion a functioning team can survive. Careful and sensitive planning when assembling the team, and adequate preventative measures at the beginning of a collaboration is important, even when, as in MDO exercises, "only the computers have to talk to each other".

4. Supportive Measures for Effective Collaborative Engineering

About 40 effects, patterns and syndromes have been listed and briefly described above. Some of them are well-known in general, some are familiar to professionals, and some might be new to some of the readers. But they have one thing in common. In the "heat of the work", managers, scientists, engineers, programmers, and all the other contributors to an ambitious design project tend to forget, neglect or shortcut the very practices they know and consider important to successful collaboration. It is one thing to know what one should do; it is another thing to obey the (unwritten and sometimes obscure) rules of good collaborative engineering in the heat of an ongoing project.

So, what to do to? For some of the factors on our list, the solution is obvious. Knowing that even the highly automated methodology of MDO forms a socio-economic system in which human behavior plays an important role is the first step. Being aware of the difficulties of humans in dealing with complexity, accepting the holistic nature of an MDO problem and respecting the professional competence and personality of collaboration partners is the next. Recognizing typical behavior pattern when they occur and avoiding to get caught in the resulting pitfalls is the advanced level.

The following recommendations shall be a small contribution enabling professionals to reach even further. They base on the studies performed during a comprehensive and highly successful MDO research project and the supportive measures that were or will be implemented to improve efficiency and effectiveness of MDO in future RTD projects.

4.1. Communication

4.1.1. Offsite communication

Personal meetings are an important driver for collaboration. In distributed, co-located activities, however, the amount of travelling and thus the number of personal, face-to-face interactions that can take place is limited. Thus, communication relies heavily on electronic means: email, phone calls, teleconferences, web meetings, and many more.

While email messages and phone calls are fine for exchanging small bits of information between few participants, they lack the ability to efficiently initiate an interactive exchange on a complex issue among a higher number of people.

Web meetings are the obvious tool to have a real-time, interactive gathering of distributed individuals. But so far, it is also one of the most fragile and annoying communication methods.

Investigations across half a dozen RTD projects revealed that in web meetings, many "golden rules" of good meetings are violated: poor planning and structuring of the moderator, insufficient preparation of the participants, multitasking or working on other issues while in the meeting, inadequate wrapping and following up, little documentation, to name but a few. Bad audio, instable connections and connection problems are additional bottlenecks.

Measures to improve the impact factor of web meetings are:

- Treat a web meeting as professionally as a face-to-face meeting, with the same level of preparation, diligence, documentation and general courtesy.
- Establish one common web meeting system that every project partner uses. This may prove to be a challenging undertaking, as it may involve aligning IT departments, security rules and additional costs for software and licenses. But if it is too difficult to agree on a common platform at the beginning of a project, how is the offsite communication supposed to work during the exercise?
- Distribute sets of appropriate headgear to all partners. Good audio quality is essential for discussing difficult technical topics in a telco or web meeting. An important feature is a mute (even better: speak) function to minimize disturbing background noises.

This may seem like a very basic and simplistic list, but these small measures have proven to make a big impact on the overall communication.

4.1.2. Visibility of ongoing activities

Distributed environments mean that people are working together alone. For maintaining momentum and minimizing errors, double-work, harmonization problems, to establish a joint understanding and to foster team spirit, it is important to assure a high visibility of ongoing activities. This cannot be achieved via a conventional web site, automated messages or shared file servers. Communication must be easy to access, fast to scan and focus on being interesting rather than comprehensive.

4.2. Commitment

It is widely accepted that the level of commitment of the participants is a decisive factor for a successful project. But less attention is devoted to how this can actually be achieved. Now, the interests and motivations of the participants can be very diverse, so no "magic formula" exists. But there are some general recommendations which may help to generate high commitment:

- It is easy to destroy commitment, but very hard to get it back.
- Projects usually start with a "clean sheet". It is essential to achieve the "buy-in" of all participants right from the start.
- Sometimes there is already some history even before a new projects starts: from past projects with the same partners, from drawing the project plan, from quarrels about content or budget, etc. Often, the problem was caused by issues which have little to do with the actual technical work. Solve these issues before the kick-off so the project can actually start with a "clean sheet".
- The project manager should not rely on the project work to assure commitment. It is often a good idea to start with a dedicated, small exercise which helps to (a) gain

momentum fast, (b) clarify fundamental questions like definitions, methods and approaches, (c) get to know the partners, (d) establish processes and procedures for the project (incl. reporting, communications. etc.).

- Do not take commitment for granted. Motivation and dedication are not assured because someone signed a contract or declared to be “thrilled by this great project”. Project managers should check back with their partners regularly to assure the project is well underway (see also next paragraph).

4.3. Feed-back and recognition

One of the things which were very important to most people who were interviewed in the course of this research is receiving feed-back on a regular basis.

Feed-back, of course, is a two-sided sword. There are a lot of mistakes that can be made easily and in the best intentions, and which may poison the further collaboration. Avoiding feed-back, on the other hand, is also not an option.

Heaps of management literature exists on how to give and collect feed-back. Most approaches are for hierarchical relationships: boss-to-employee or vice versa (e.g. 360° feed-back). With adequate caution, many of these recommendations are also valid for a collaborative engineering environment.

Some basic points in this context:

- Feed-back within the design team has to be asked for / has to be given on a regular basis.
- Golden rule #1 when receiving feed-back: Listen. Do not argue, justify, excuse or talk away feed-back points. Accept the feed-back as the other person’s valid opinion in the first place. Reconsider it in a quiet minute to decide if you want to act upon it.
- Golden rule #2 when receiving feed-back: When you are answering to feed-back, make sure that the other party does not get the impression he/she is not taken seriously.
- Golden rule #1 when giving feed-back: Good feed-back gives observations and facts. It does not give interpretations, accusations and valuations. It is about what a person did, not about the person itself.
- Golden rule #2 when giving feed-back: It is not the idea of feed-back to “settle everything for now and forever”. So do not wait for a regular feed-back session to address specific things which annoy you; clarify this immediately. Feed-back is for issues that develop slowly, are of a general nature or jump the hierarchy.

4.4. Situation awareness

Good and comprehensive overview of the overall system, the status of the subsystems and the interconnections is essential enabler for designing and optimizing a complex system. From the top-down view of the project manager and the integrator, this is quite clear.

For a disciplinary expert or a detail-design engineer, this is not as obvious. It is an ambitious but rewarding challenge to find the right focus and the right balance. For all the project reviewed in the course of this work, however, the findings were that most participants would have liked to have a better idea of the overall status, the achievements and the impact of the individual contribution.

4.5. Visualization

Not only in the field of (collaborative) MDO, visualization of relevant data in a human-comprehensible way is an essential factor to understand complex systems. Showing data trends in diagrams instead of just the raw numbers or data points, visualizing interdependencies between different design competences (and therefore, responsible design specialists) and usage of ergonomic coloring are some of the many techniques that can be used to achieve such an understanding. In the field of MDO for instance, a number of techniques have been developed over the years to visualize MDO problem formulations. This means the way a possible tool workflow solving an MDO problem is set up, containing information about tool interdependencies as well as their execution order. Some of the well-known techniques are the N2 chart developed by Lano in 1977, [16],

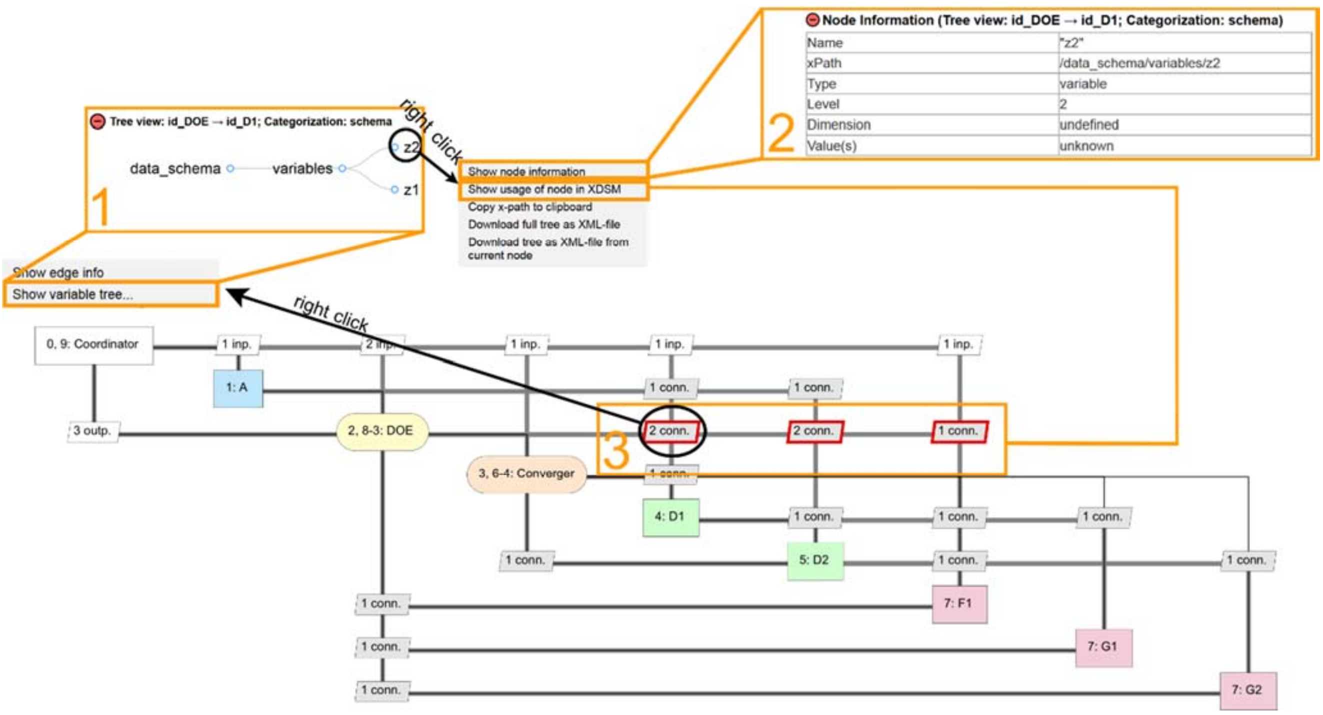

Fig. 4: VISTOMS XDSM for the "Sellar Problem" MDO system, [21]

the Design Structure Matrix (DSM) introduced by Steward in 1981, [17], and the Functional Dependency Table (FDT) by Wagner and Palambros, [18]. More recently, Lambe and Martins investigated state-of-the-art visualization techniques in a survey, [19]. The result of their research is the so-called Extended Design Structure Matrix (XDSM), which enables the full description of an MDO problem, including data and process information.

While the XDSM covers the entire MDO system description, in its natural form, it still lacks readability for humans as soon as the MDO system becomes very large. Therefore, within the AGILE project, the XDSM from Lambe and Martins was further enhanced base on an already existing software package from Lafage, [20].

The result is the HTML-based interactive software tool VISTOMS (VISualization Tool for MDO Systems), which allows for investigation of MDO systems of almost arbitrary size. Zooming in/out, right mouse clicks, etc. make it possible for an MDO specialist, to debug an MDO system effectively, [21].

A simple but effective supporting feature which can improve interactive communication, interpretation and modification of technical data in a complex interdisciplinary multilevel system is the "system navigator": a visualization feature which shows, in an additional display next to the working screen, which region of the design space or which level, part or connection of the overall system the team is currently working on. This display is favorably equipped with a "zoom in / zoom out" capability which intuitively visualizes the current focus of the team and how it is connected to the overall context.

4.6. Collaborative project management

4.6.1. Leadership

Collaboration projects are raising high demands on the management and leadership skills of the coordinator and his or her supporting managers, e.g. local representatives and work package or task leaders.

A dedicated, focus training on technical collaboration would be most valuable for young professionals who are in a responsible position for the first time. But there are few appropriate courses, so consequently, candidates receive general (project) management training, the quality of which can differ significantly from provider to provider. Even more common is to simply depend on a "learning-by-doing" approach, which can be a very frustrating experience for the young manager, demotivating for the team members and expensive in the long run for the company. The more important it is to carefully select the appropriate training program(s): for the individual trainee as well as for the requirements of the specific project.

4.6.2. Transparent day-to-day management

Open and interactive management and a good flow of information is one of the issues which is mentioned almost ubiquitously. Important as this topic is, there should also be a word of caution: participants see it often as a one-way street. Getting to know what is going on is as much an obligation for the team member as for the team leader.

4.6.3. IT versus engineering

In most if not all MDO projects, there is a hidden conflict going on between information technology (IT) and engineering. Many software tools for analysis and simulation require a lot of attendance: programming interfaces, establishing and maintaining the IT networks, managing the data bases, etc. All this activity providing support to the MDO exercise. But the objective is to solve an engineering problem, not an IT problem. Resources (time, budget) going into "making the system run" have to be carefully planned and continuously monitored to ensure that the MDO exercise can deliver meaningful results when they are needed.

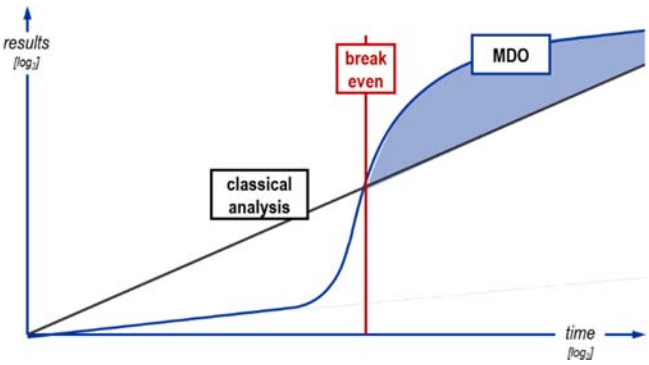

Fig. 5: The "MDO Hockey Stick" and the challenge to "break even" in good time

Similar to the "hockey stick growth" investors expect from start-ups after an initial funding phase, the investment into MDO has to bring a significant dividend to the design program. The achievement is not to have a highly sophisticated framework up and running, but to deliver relevant answers that cannot be achieved otherwise.

4.7. The AGILE Experience

The research project AGILE achieved significant technical progress on MDO, especially in the areas of data management and agile and flexible composition of complex MDO problems. But the project also highlighted the necessity to consider an MDO application as a socio-technical instead of a purely technical challenge.

4.7.1. Best practices of collaborative design

With the experiences and research results of AGILE, a composition of "Best Practices" for an MDO-supported collaborative design environment was created.

Available at the moment as a deliverable for the project, the project team is aware that lengthy reports are not the best means of communication to project leaders and their teams when commencing or being engaged in a design exercise. Activities are underway to make this compendium available in a more user-friendly, e.g. in the form of a sequence of short online clips.

4.7.2. AGILE thinking

AGILE thinking is an approach to combine the possibilities of MDO with the methodology of *design thinking*, [22].

Design thinking bases on three principles:

- *Ideation*, i.e. idea generation in an alternating process divergent and convergent thinking;
- Open-minded *collaboration* for the exchange and advancement of design ideas;
- Rapid *implementation and prototyping* of innovative ideas to cut to the chase.

Basing on a variety of approaches for innovation and collaboration, design thinking has been established as a formal problem-solving method at Stanford University in 2005. The method is now used by many large organizations, especially in the fields of marketing and user-centric development of products and services.

AGILE thinking takes up the underlying mechanisms which make design thinking dynamic and innovative, and transform it to an approach that is suitable to the abstract mathematical/technical relationships of aircraft design, with MDO as an essential enabler.

References

[1] Crowder, R.; Bracewell, R.; Hughes, G.; Kerr, M.; Knott, D.; Moss, M.; Clegg, C.; Hall, W.; Wallace F.; Waterson, P.: *A Future Vision for the Engineering Design Environment: A Future Sociotechnical Scenario*. ICED 03 - International Conference on Engineering Design; Stockholm, Sweden; 2003.

[2] Törlind, P.; Larsson, A.: *Support for Informal Communication in Distributed Engineering Design Teams.* Proceedings of 2002 International CIRP Design Seminar; Hong-Kong, 2002.

[3] Rivas McGowan, A.-M.; Papalambros, P. Y.; Baker, W. E.: *A Framework of Working Across Disciplines in Early Design and R&D of Large Complex Engineered Systems*. ICED 15 - 20th International Conference on Engineering Design; Milan, Italy; 2015.

[4] Cohen, B.; Stroud Rossmann, J.: *Introducing Engineering as a Socio-technical Process.* 121st ASEE Annual Conference & Exposition; Indianapolis, IN, USA; 2014.

[5] Bloebaum, C.: *The Design of Large-Scale Complex Engineered Systems: Present Challenges and Future Promise*. 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; Indianapolis, IN, USA; 2012.

[6] Magee, C. L.; de Weck, O. L.: *Complex System Classification*. Proc. of 14th Annual International Symposium of the International Council On Systems Engineering (INCOSE); Toulouse, France; 2004.

[7] Dörner, D.; Schaub, H.: *Errors in Planning and Decision Making and the Nature of Human Information Processing*. Applied Psychology; Vol. 43, Issue 4, p. 433-453. International Association of Applied Psychology, Wiley; 1994.

[8] Funke, J.: *Problemlösendes Denken.* Verlag W. Kohlhammer, Stuttgart, 2003.

[9] Funke, J.: *Complex problem solving: a case for complex cognition?* Cognitive Processing; Volume 11, Issue 2, pp 133–142; Springer Verlag, 2010.

[10] Fischer, A.; Greiff, S.; Funke, J.: *The Process of Solving Complex Problems*. The Journal of Problem Solving; Vol. 4, Iss. 1, Article 3; 2012.

[11] Dörner, D.: *On the Difficulties People have in Dealing with Complexity*. Simulation & Games, 11, p. 87-106, SAGE Publishing; 1980.

[12] Dörner, D.: *Die Logik des Mißlingens. Strategisches Denken in komplexen Situationen*. Reinbek: Rowohlt, 1989.

[13] Gosselin, D; Gagnon, S.; Stinchcombe, A.; Joanisse, M.: *Comparative optimism among drivers: An intergenerational portrait*. Accident Analysis & Prevention; Vol. 42, Issue 2, p. 734-740, Elsevier Ltd; 2010.

[14] Janis, I. L.: *Victims of groupthink; a psychological study of foreign-policy decisions and fiascoes.* Houghton Mifflin Company; Boston, USA; 1972.

[15] Marques, J. M.; Yzerbyt, V. Y.; Leyens, J. P.: *The black sheep effect: Judgmental extremity towards ingroup members as a function of ingroup identification*. European Journal of Social Psychology; Vol.18, Issue1, p. 1-16; John Wiley & Sons, 1988.

[16] Lano, R. J.: *The N2 Chart*. TRW Software Series, 1977

[17] Steward, D. V.: *The design structure system – A method for managing the design of complex systems.* IEEE Transactions on Engineering Management, Vol. EM-28, No. 3, pp. 71-74; IEEE Technology and Engineering Management Society; 1981.

[18] Wagner, T. C.; Papalambros, P. Y.: *General framework for decomposition analysis in optimal design*. ASME Advances in Design Automation, Vol.2, pp. 315-325; New York, NY, USA; 1993.

[19] Lambe, A. B.; Martins, J. R. R. A.: *Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes*. Structural and Multidisciplinary Optimization, Vol. 2, pp. 273-284; Springer, 2012.

[20] Lafage, R.: *XDSMjs package - XDSM diagram generator written in javascript*. ONERA, 2016, https://github.com/OneraHub/XDSMjs, last checked July 16, 2018.

[21] Aigner, B.; van Gent, I.; La Rocca, G.; Stumpf, E.; Veldhuis, L. L. M.: *Graph-based algorithms and data-driven documents for formulation and visualization of large MDO systems*. CEAS Aeronautical Journal Vol. 40; 2018.

[22] Plattner, H.; Meinel, C.; Leifer, L. J.; (eds.): *Design thinking: understand, improve, apply. Understanding innovation*. Springer-Verlag; Berlin, Heidelberg; 2011.

5. Contact Author Email Address

martin.spieck@thelsys.de

Copyright Statement