11th World Congress on Structural and Multidisciplinary Optimisation
07th -12th, June 2015, Sydney Australia

Multidisciplinary optimization and integration requirements for large-scale automotive and aerospace design work

Moritz Frenzel[1], Daniel Heiserer[1], David Keller[1], Markus Schemat[1], Vladimir Balabanov[2], Rodney Dreisbach[2], Steve Georgiadis[3], David Trop[2]

[1] BMW Group, Munich, Germany, moritz.frenzel@bmw.de
[2] The Boeing Company, Seattle, WA, USA, vladimir.balabanov@boeing.com
[3] The Boeing Company, VIC, Australia; steve.georgiadis@boeing.com

1. Abstract
Automotive and aero industries are rapidly increasing applications of numerical simulations for structural, structure-interfacing, and multi-field analyses ranging from structural stiffness and strength, to crashworthiness and durability. Simulation applications and tool chains are cast into sophisticated, but strict, processes to ensure reliability, design integration, and interaction between partners, departments and suppliers.

Commercial and in-house optimization frameworks, i.e., process integration and design optimization (PIDO) tools, have evolved considerably, allowing for coupling of processes, tools, and individual design parameters. Thus, the designer/CAE specialist is required to master the challenges arising from the complexity of such processes. Although originally intended for this specific purpose, even efficient PIDO implementations may not be suitable for general applications from an enterprise standpoint. Especially for multi-disciplinary optimization when analyses from various disciplines compete and their influences need to be balanced.

This paper presents the background and rationale why PIDO implementations may not be suitable from an enterprise aerospace/automotive perspective. A view of the bottlenecks is also presented, along with proposed approaches to resolve them.

Specifically, to increase the efficient use of commercial PIDO tools in the automotive and aerospace industries, these integration and optimization frameworks should provide:

- Friendlier ways of integrating existing third-party and legacy tools
- Interactive human control of the optimization process, i.e., "on-the-fly" adjustments of the design variables, targets, constraints, and optimization methods
- Intuitive and robust support of heterogeneous computing systems
- Ease of maintaining and modifying the created processes that should be available both in GUI and batch modes.

The PIDO approach demands high flexibility, with strong end-user interaction and interfacing.

2. Keywords: Process Integration and Design Optimization (PIDO), MDO, Complex Engineered Systems, Enterprise Optimization Framework, Big Data, Preliminary Design

3. Motivation

Recent years have seen an enormous growth in computer aided engineering (CAE). In the coming years, the available computing power will increase even faster. Any item and part with structural requirements within complex products such as automotive vehicles or airplanes is designed using high fidelity structural simulations including finite element analyses. Usage of high fidelity, fast computational structural mechanics (CSM), computational fluid dynamics (CFD), and multidisciplinary optimization (MDO) tools and processes has significantly improved product performance and greatly reduced product development costs. By taking advantage of advanced computational analysis tools and coupling analyses to multidisciplinary optimization tasks, designers can simultaneously improve the product design, and reduce the time and cost incurred during every design cycle. There is a widespread virtualization strategy in industry to reduce the number of experimental validations required to "certify by analysis" [1]. The objective is to further reduce product development costs for the aero-industry, and for "development on demand" for the automotive industry with its massive unit numbers and customer drive for superior performance and individualization.

Today's approach within large enterprise CAE development organizations is to cast each and every tool and development step into a clearly defined process. Interaction of the design disciplines, development partners, and suppliers, together with strict project schedules, necessitate strict responsibilities. Such responsibilities are reflected in reasonably strict CAE process chains. To accommodate this CAE development environment, software tools are required to provide very effective and efficient interfaces. There has been a flurry of merger and acquisition activities taking place in the software industry, with the objective of providing the "one and only" software suite that is the best "integrator" around. However, no single application or system is capable of handling all of the product design issues, spread out to all companies and all departments, and to resolve the necessary interactions between the tools and data. Product data management (PDM) systems are mostly in place but have not truly penetrated the CAE simulation world yet. Mastering the entirety of simulation data generated within an enterprise, even within a period of only one year, constitutes a big burden of resources and development cost.

Optimization, specifically in the sense of PIDO, shifts these challenges into another dimension. Commercial software tools such as Optimus [2], Isight [3], Dakota [4], ModeFrontier [5], among others, are offering a reasonable coverage of system integration from CAD and CAE software to optimization, visualization, statistical analysis, and full product data management (PDM) integration. The underlying approach has always been to introduce a so-called "master flow" which determines the optimization process that includes analyses, iteration loops, along with computing resource and job management.

From an enterprise standpoint, this master flow approach is not reasonable. It inevitably brings out several issues which are presented below. The current design work in industry is already so complex that no single tool, or vendor, or process is capable of adequately adjusting to all of the demands of high fidelity design work. In this paper, this issue is elaborated, along with a vision of how to overcome this trap. Our vision is driven by the belief in computational design, data affinity, and future design processes.

The issues of a master-slave context are presented in the following section with respect to computing resources by considering a specific type of genetic algorithm (GA) as an example. Next, the basis of a vision for a next generation computing and numerical optimization environment is presented by reassembling the building blocks of a (potentially automated) design process. Finally, the requirements for this transition presented, along with a discussion via several open questions.

4. GA with respect to computing resources: Synchronous Master-Slave versus an Asynchronous Approach

Genetic algorithms (GAs) are an attractive class of techniques for solving a variety of complex search and optimization problems. Although they are not the only possible approach even for discontinuous problems, GAs are an integral part of most PIDO tools. GAs offer a global optimization strategy at the cost of heavy computing resources for state evaluation in every generation. Classical GA starts by evaluating responses at a predefined number of points. This set of points is called a generation. For a large number of points, the evaluations are usually performed in parallel. For large-scale problems of practical importance, distributed computing techniques are typically implemented. However, classical GA also requires synchronization; it requires correlation of the results of all the points in a single generation after all the points in the generation are evaluated. This synchronization point becomes a road block for a heterogeneous computing environment when the process to evaluate a single point in the generation is significantly slower than for the others. The overall process is determined by a so-called master process which consists of generation synchronization (environmental selections) and offspring generations. Multiple, parallel point evaluations constitute the slave jobs. Such types of algorithms are called synchronous distributed master-slave GA (SDGA).

The speedup lost in synchronizing a point may be considerable in networked, heterogeneous computing environments [6]. Asynchronous (also called generation-less) GAs have been proposed to overcome this drawback via alternative implementations of the individual's life-cycle dissolving a strict generational evaluation. The reader is referred to [7] for details regarding modelling of an individual's life-cycle and mating strategies. In computing terminology, an asynchronous distributed GA (ADGA) is obtained by "unrolling" the loop of generation, crossover, and mutation until convergence is achieved.

In heterogeneous network environments, one single slow processor may impede the overall progress in executing SDGA. Significant speedup can be obtained by implementing the idea of ADGA [6]. Moreover, complete resource management and scheduling could be decoupled. Implementation of such ADGA may achieve partitioning of GA schemes into pieces of work which can be processed in parallel. An optimal partitioning in terms of runtime speedup should allow for a full utilization of all available resources.

5. PIDO approach and the eternal resource bottleneck in enterprise design work

Transferring the idea of asynchronous versus synchronous genetic algorithms to the whole optimization process reveals significant issues within current PIDO tools and their proposed integration into enterprise design work. For this case, the PIDO tool assumes the role of the master, with the slave-jobs being plugged in via numerous interfaces. The master thus controls the overall optimization process; however, in every enterprise, the resources required to fulfill design work are always limited. This includes the required computing power in the form of the number of computer cores, computing time, disk space, and computer memory; they are referred to as hardware cost. The PIDO tool also assumes the role of a scheduler to distribute slave jobs to any computing environment by controlling all of the associated resources. Moreover, when evaluating sophisticated functions by CSM, CFD, or MDO tools, computation of the state evaluations presents a resource issue on its own, e.g., with respect to software licensing. One should not underestimate the challenge for an enterprise to decide whether a number of licenses is provided for multiple design tasks instead of using the same number of licenses for one sophisticated multi-job task. Smart decisions must be based on the expected payoff for the engineering design problem, but should also include estimation of computing and license resource consumption. Today, PIDO tools lack transparent visualization and tracking of such resources. Furthermore, clear insight into multi-job status, job scheduling, and solution convergence and robustness are essential.

Last but not least, for enterprise design work, the limited resource *time to solution* represents not just computing wall clock time, but also engineering decision time. Typically the analyst is faced with the dilemma of setting up one fast single job for a specific design versus a multi-job design exploration or optimization process. Intelligence and guidance for optimization post-processing to understand why a particular design is superior to another is a key to fast decisions in the time-pressured enterprise project work environment.

From an enterprise standpoint, it is believed that the current master-slave PIDO approaches will never be able to fulfill all of these requirements. Instead, it is proposed that a different approach is followed in the future for reassembling the typical PIDO modules as described in the following.

6. Vision of Optimization Data Management Engine for Resource-optimal Enterprise Design Work

To resolve the issues described above, it is proposed that the classical master-slave approach be decomposed and moved to what is called an Optimization Data Management Engine (ODME). By analogy with the genetic algorithms presented above, any synchronization points within the complete CAE design process can be avoided. To illustrate, consider the multidisciplinary optimization tasks of airplane or car design where CFD and structural analyses such as crash and noise, vibration, and harshness (NVH) are involved. Suppose that the CFD or crash analysis takes more time and resources than for the structural analysis. For a typical master-slave PIDO, all analysis results are synchronized only after all of the analyses are completed. Thus, a quick structural analysis will typically need to wait for the CFD or crash analysis to be completed. This bottle-neck is inherent to master-slave approaches.

Our vision for a multidisciplinary multi-job optimization environment is to break up the master-slave components into process modules. A conceptual draft of the *reassembled* modules is depicted in Fig. 1 which consists of evaluation modules (CSM/CFD/MDO tools), post-processing and visualization modules, resource controller modules (computing load share, licensing control, job scheduling), and driver modules (DOE, RSM, single analysis request, Data Mining, Optimization algorithms, etc.). The core around which those modules are arranged is a high-performance database storing all relevant analysis models and results: CAD/CAE model properties, design variables, dedicated response values, as well as job Meta data such as job run time, resource consumption, submission status, etc. Based on today's technical standards, open or standard protocol (SQL) database concepts can be utilized to implement such a database.

Let us refer again to the MDO optimization example illustrated above and show how it would look within the ODME. The optimization algorithm is initiated from some driver module and sent to a Resource Controller in the database; this is identified by a "to evaluate" marker for the corresponding CFD, Crash and NVH analyses being scheduled by the Resource Controller according to the available software licenses and high-performance computing (HPC) load sharing. As soon as the license and HPC load share are available, every particular Evaluation Module starts its analysis. Thus, a prioritized single job evaluation may "overtake" an optimization evaluation for a period of time, but not prevent the rest of the evaluations in the running generation. Meanwhile, quick and cheap evaluation results, if present, are already fed back to the Database Resource Controller and could be visualized by a selected Visualization Module. Based on the current solution information, the user might want to change the optimization parameters or variables, or to set up a completely new Driver Module. At some point, all of the required evaluations are completed, with the initial optimization task being completed and the results being stored in the Database.

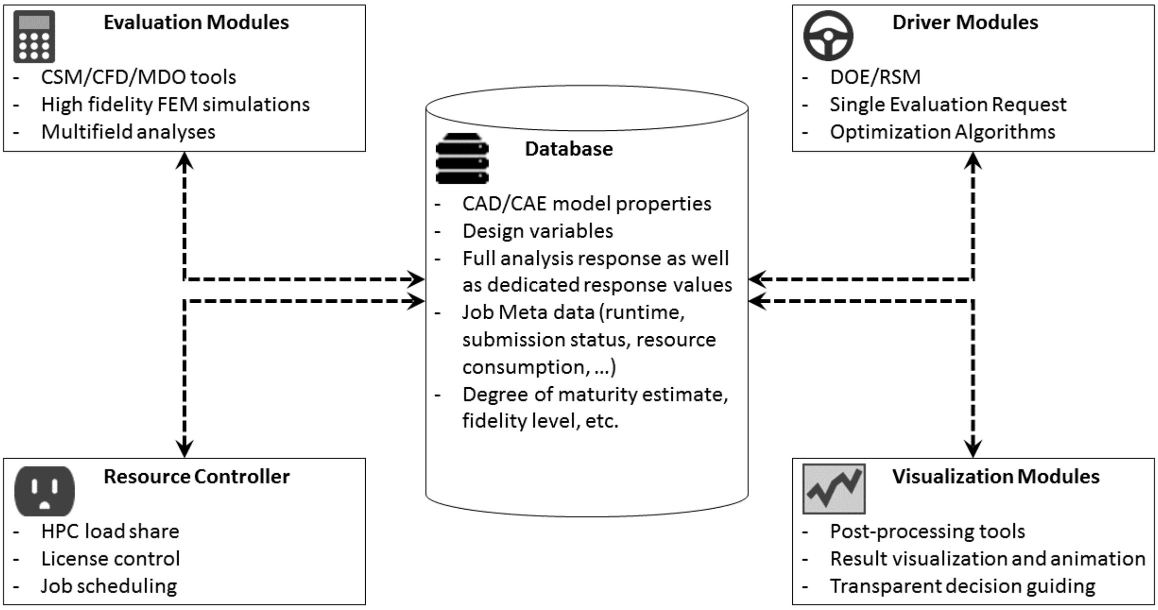

Figure 1: Optimization Data Management Engine as a reassembly of PIDO master-slave approach: Database interacts with Resource Controller, Driver, Evaluation, and Visualization modules

Within the proposed environment, more features could be implemented successively. Eventually, a classification of the maturity of the analysis results is envisioned to account for different levels of fidelity. Such a classification would allow for efficient data usage in different design stages, from predesign and design exploration up to detailed design, life cycle analysis, and certification. More practical features could easily be implemented such as multiple user and client access, web-based interfaces, etc.

7. Conclusions

The classic PIDO-based numerical optimization approach is presented in this paper, along with its conceptual drawbacks in the context of the enterprise design work requirements from the automotive and aerospace perspectives. To overcome the underlying issues of this approach, numerous changes are proposed for the standard master-slave approach being implemented in today's PIDO tools. An enterprise computing environment based on an optimization data management engine (ODME) is envisioned by reassembling the existing PIDO modules around a central analysis evaluation and storage database with a resource controller.

It is expected that when the proposed new paradigm is implemented, multidisciplinary and large-scale optimization capabilities will be easier to adapt to existing design processes in the automotive and aerospace industries. Furthermore, it is expected that the PIDO-based approach and other approaches to optimization will gain more industrial support and enterprise-wide implementation. As a result of implementing the new proposed PIDO concept, it is anticipated that a new phase of growth of numerical optimization and associated numerical simulation in enterprise design work will ensue.

8. References

[1] Somanath Nagendra et. al., Industry Perspectives on Integrated System Optimization and MDO, *10th World Congress on Structural and Multidisciplinary Optimization,* In: Proceedings of World Congress on Structural and Multidisciplinary Optimization WCSMO-10, 19.-24.05.2013, Orlando, FL (USA)
[2] Optimus Software Package, Release 10.15, 2015, Noesis Solutions, Gaston Geenslaan, 11 B4, 3001 Leuven | Belgium, http://www.noesissolutions.com/Noesis
[3] Isight/Simulia Execution Engine (SEE) Software package, Release 5.9, 2015, Simulia, Simulia World Headquarters, 166 Valley Street, Providence, RI 02909 USA, http://www.3ds.com/products-services/simulia/products/isight-simulia-execution-engine
[4] Dakota Software Package, Release 6.1, November 15, 2014, Sandia National Laboratories, 1515 Eubank SE, Albuquerque, NM 87123, https://dakota.sandia.gov
[5] modeFRONTIER Software Package, Release 2014, Esteco, Padriciano 99, 34149 Trieste Italy, Building L2, http://www.esteco.com/modefrontier

[6] John W. Baugh and Sujay V. Kumar, Asynchronous genetic algorithms for heterogeneous networks using coarse-grained dataflow, *Genetic and Evolutionary Computation - GECCO 2003*, 2003.
[7] David C. Keller, Evolutionary Design of Laminated Composite Structures, available from http://dx.doi.org/10.3929/ethz-a-006082073, ETH Zürich, Zürich, Switzerland, 2010.