Proceedings of the IASS-SLTE 2014 Symposium
"Shells, Membranes and Spatial Structures: Footprints"
15 to 19 September 2014, Brasilia, Brazil
Reyolando M.L.R.F. BRASIL and Ruy M.O. PAULETTI (eds.)

Interactive Distributed Optimisation for Multidisciplinary Design

Dion JANSEN[1,2], Anke ROLVINK[1,3], Jeroen COENDERS[1,4],
Mattias SCHEVENELS

[1] BEMNext Lab, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands and White Lioness technologies, Amsterdam, The Netherlands
[2] *dionjansen@white-lioness.com*
[3] *ankerolvink@white-lioness.com*
[4] *jeroencoenders@white-lioness.com*
[5] Department of Architecture, Faculty of Engineering Science, KU Leuven, Leuven, Belgium,
mattias.schevenels@kuleuven.be

Abstract

This paper presents the development of a multidisciplinary optimisation system for the Architecture Engineering and Construction (AEC) industry. The system consists of two major components, namely a multidisciplinary optimisation framework and a distributed cloud-based analysis framework. The former utilises an interactive optimisation search strategy to solve multidisciplinary design problems by giving both insight into the performance of the optimisation problem as well as the performance of the applied multidisciplinary optimisation strategy. The latter provides a flexible infrastructure to rapidly evaluate design alternatives. The system as a whole is aimed at providing designers and engineers with an intuitive tool to define, evaluate and optimise the performance of large multidisciplinary models.

Keywords: design tools, design optimisation, distributed computing, cloud computing, Grasshopper, parametric design, Multidisciplinary Design Optimisation, MDO

1. Introduction

In this paper the development of a multidisciplinary optimisation system is presented. The aim of this system is to support the conceptual design process of the AEC industry. Generally, design problems faced in the AEC industry are characterised by a multidisciplinary character, where often the performance of individual disciplines is in conflict with the performance of at least one other discipline. The concept of overall (multidisciplinary) performance is ill-defined, since often this performance is not solely defined by technical (quantifiable) aspects but also by semi- or non-quantifiable aspects (e.g. aesthetics). Furthermore, the unique character of building designs impose a unique problem statement that makes standardisation of a solution strategy difficult, which greatly affects the applicability of dedicated systems to support the design process of building designs in general. Lastly, building design problems involve many parameters required for the evaluation of performance on several different disciplinary professions which makes automated optimisation in an economically feasible time-frame challenging.

Therefore a system has been implemented using a four development strategies aimed at taking these characteristics into account. These development strategies are interpreted as the base *layers* of which the system is comprised:

(1) The *modelling layer* where a central parametric modelling environment provides a platform in which the model as well as simulation parameters, performance functions and optimisation configuration can be defined.
(2) The *optimisation layer* where a parallel optimisation solver supports the dynamic creation, adjustment and updating of multiple objective functions simultaneously in order to support the dynamic search process of the designer in the conceptual design stage.
(3) The *generation layer* which provides a scalable back-end framework to support rapid generation of geometry and simulations.
(4) The *visualisation layer* where the state of the optimisation search process is expressed to the designer to support the designer in making informed design choices on the current behaviour of the problem.

The system architecture in which all for layers are represented is illustrated in Figure 1. Here several system components are linked together through a series of cloud-based data stores to form the different layers of the system. The layers will be discussed further in this paper in terms of their design, implementation and use for the AEC's design process.

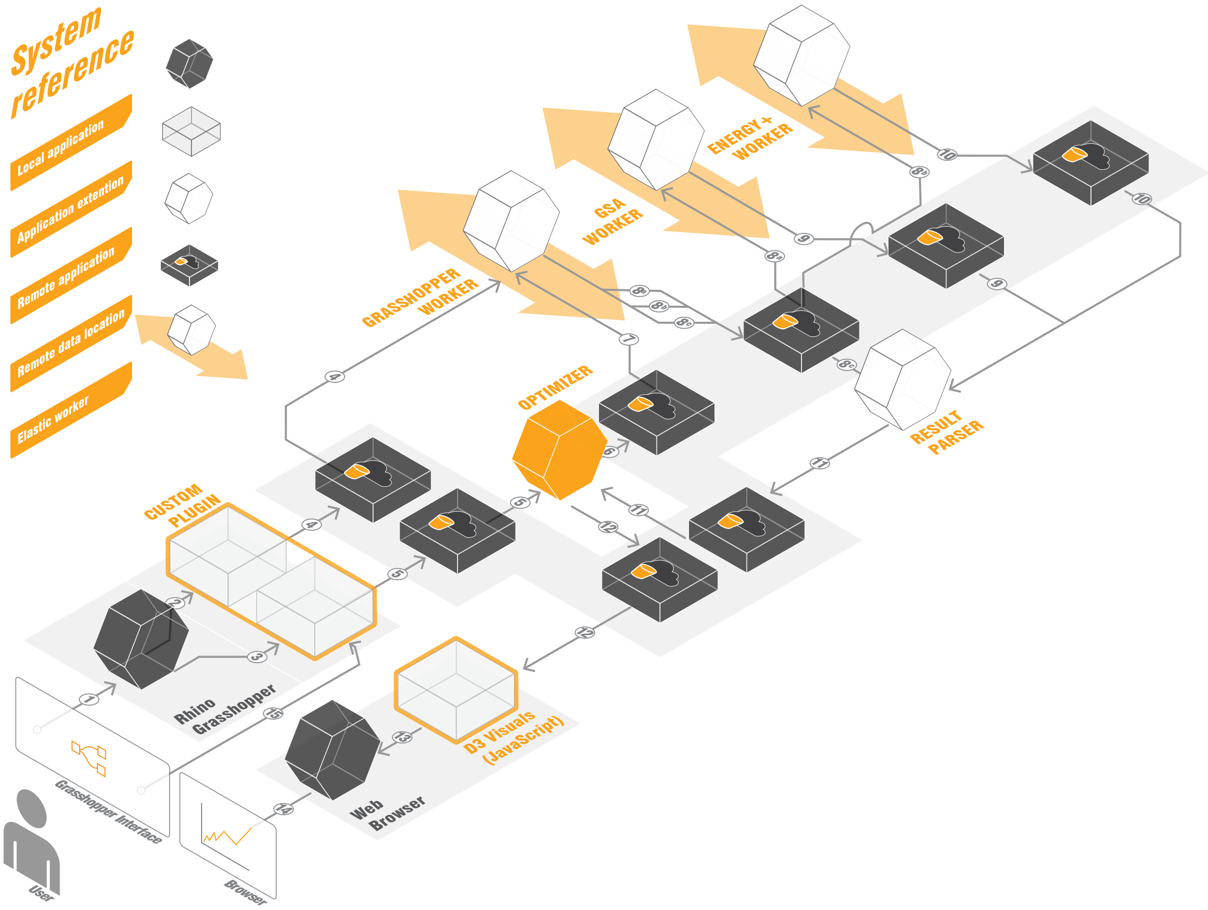

Figure 1. System overview expressing the four layers: modelling, optimisation, generation and visualisation layer

2. MDO architecture

Existing Multidisciplinary Design Optimisation (or MDO) architectures represent the way in which discipline objectives and variables interact to simultaneously optimise the problem. Many architectures exist and have been extensively studied in both theoretical studies (Lambe A.B. & Martins J.R.R.A [7], Lambe A.B. & Martins J.R.R.A. [8]) as well as practical applications (Grossman B. et al. [4],Brown N. [1], Kipouros T. et al. [5]). In the research of a suitable architecture to apply in a typical design process of the AEC industry it was found that existing architectures heavily rely on tightly coupled simulation procedures between discipline simulations (i.e. responses of one simulation influence a second simulation which influences the first). However, for the conceptual design process in the AEC industry such level of coupling is rarely required between simulations of different disciplines. In this way multidisciplinary performance can be measured by the combination of loosely coupled disciplinary performance measurements in a single objective statement. In order to support the user in defining the objective statement, a MDO architecture is developed to allow for the simultaneous evaluation of multiple different statements in so called scenarios. The scenarios are captured in a visualisation interface where the user can evaluate the performance outcomes of different scenarios and based on the knowledge gained from this feedback instantiate new scenarios. Figure 2 expresses this process in an extended design structure matrix (XDSM) (Lambe A.B. & Martins J.R.R.A [9]). The diagonally positioned system components have been defined so that their position from left to right denotes their order of execution by the system. The horizontal lines represent the output or response of a system component whereas the vertical lines define input for system components in the same column. The items at junctions express the data variable such as system variable, constraint or analysis response. Numbers at all items in the XDSM are applied to emphasise the order of execution.

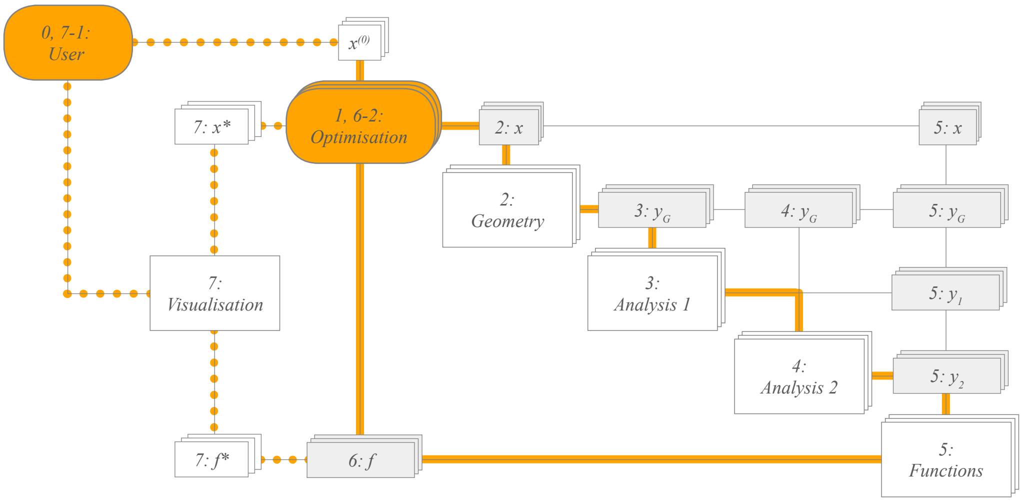

Figure 2. XDSM of Interactive Parallel Multidisciplinary (IPMD) MDO architecture, expressing the relation between user interaction (0) parallel optimisation and generation system structure (1-6) and the visualisation (7) implementations.

3. Modelling layer

The initial stage of the system setup is the definition of a model to optimise. In order to support the rapid generation of design model states the modelling layer is built as an extension (or plug-in) to the parametric and associative design application Grasshopper (*grasshopper3d.com*) for Rhinoceros (*rhino3d.com*). Grasshopper is a powerful tool for representing geometrical modelling data in an algorithmic definition that is expressed geometrically through the Rhino viewport. This allows for the expression of quantifiable aspects of the modelling logic alongside with non-quantifiable aspects such as aesthetics through the geometrical representation. The aim of the modelling layer is two-fold: firstly, it provides the user with a single model environment to define the variables of the system, the model relationships, the interpretation into disciplinary models (e.g. structural, energy). Secondly, the user is given the ability to define performance equations (or utility functions) resulting in one or multiple performance (or fitness) outputs. An example definition in Grasshopper of a simple multi- objective problem is illustrated in Figure 3 along with its aesthetical representation. Once the optimisation process is started the components of the optimisation solver remain under the control of the user. In this way the designer has the ability to rearrange the performance equations so as to start new optimisation processes and stop or update existing optimisation processes.

4. Optimisation layer

The optimisation layer has been developed with the MDO architecture illustrated in Figure 2 in mind. It deliberately allows the user to manipulate the optimisation process by instantiating new scenarios at will to effectively explore different areas of the design space for different scenarios. Within each scenario a number of single objective optimisation processes is processed in parallel where coupling between different objectives is performed solely by the user. This user-centric approach allows the designer to explore the performance of different single disciplines as well as the influence of the trade-off between them.

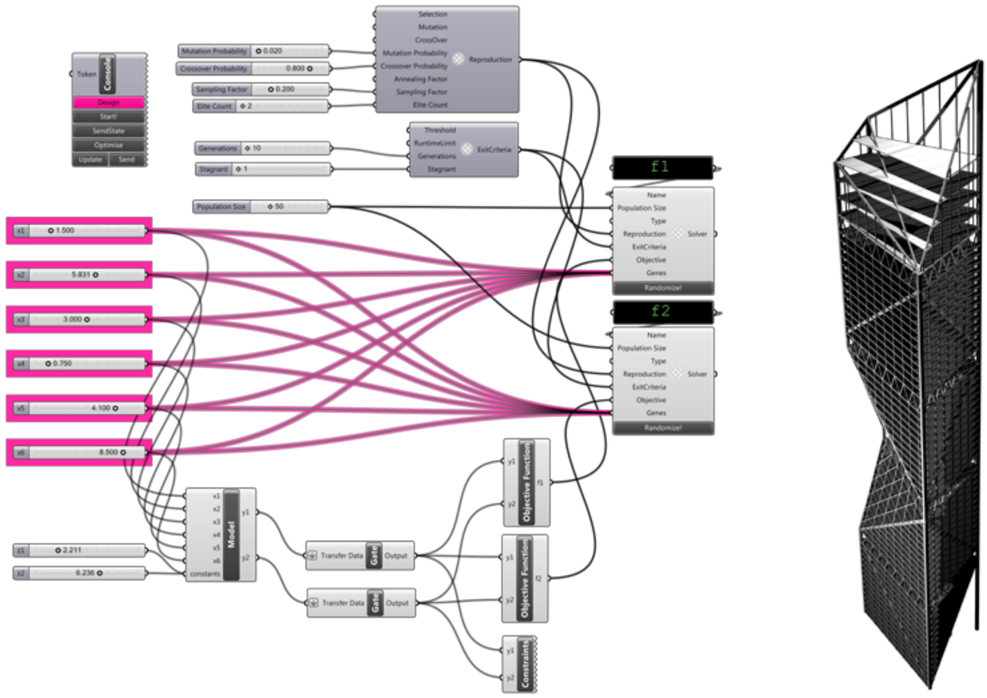

Figure 3. (left) Grasshopper model showing a basic optimisation problem definition with six system variables (x1-x6) as Grasshopper native slider components and two objective functions (f1-f2), (right) the geometrical representation of the model in the Rhino viewport.

The optimisation layer implements a single continuous control thread which orchestrates the optimisation process and allows the user to dynamically provide input in the form of configuration files produced in the modelling layer. The orchestrator can initialise, stop or update slave threads depending on the amount of objective functions defined by user configuration. Each slave thread performs a single objective optimisation procedure using a custom genetic algorithm (GA). The parameters of the GA such as population size, mutation rates and cross-over rates are customised by the user. Each slave thread performs the optimisation of a single objective and maintains a fixed size population of chromosomes. The components of the optimisation layer are expressed in the process chart in Figure 4.

In order to allow continuous evaluation a steady state update mechanism is implemented to avoid blocking of the GA during the evaluation of populations. In this way the process is not blocked by the evaluation of the full population so that operators such as selection can operate on the most recent state of the population. A non-uniform mutation scheme promotes exploration of the design space while gradually fine tuning the search process of a single objective.

5. Generation layer

The generation layer evaluates the design states produced by the optimisation layer in the system. This layer has been developed based on a distributed computing model where components located on networked computers communicate and coordinate their actions by passing messages. This distributed computing model is aimed at providing a scalable system that can either run in small scale (locally for small problems) or in large scale (remotely for large problems). System components can be grouped together based on their dependencies on other system components is so called distribution groups. Distribution groups can be scaled depending on the performance required of the group and communication between groups is handled through queueing. No communication exists between distributed programs within a group. The evaluation of a certain input into a multitude of responses using a distributed computing model is illustrated in Figure 5 (left image).

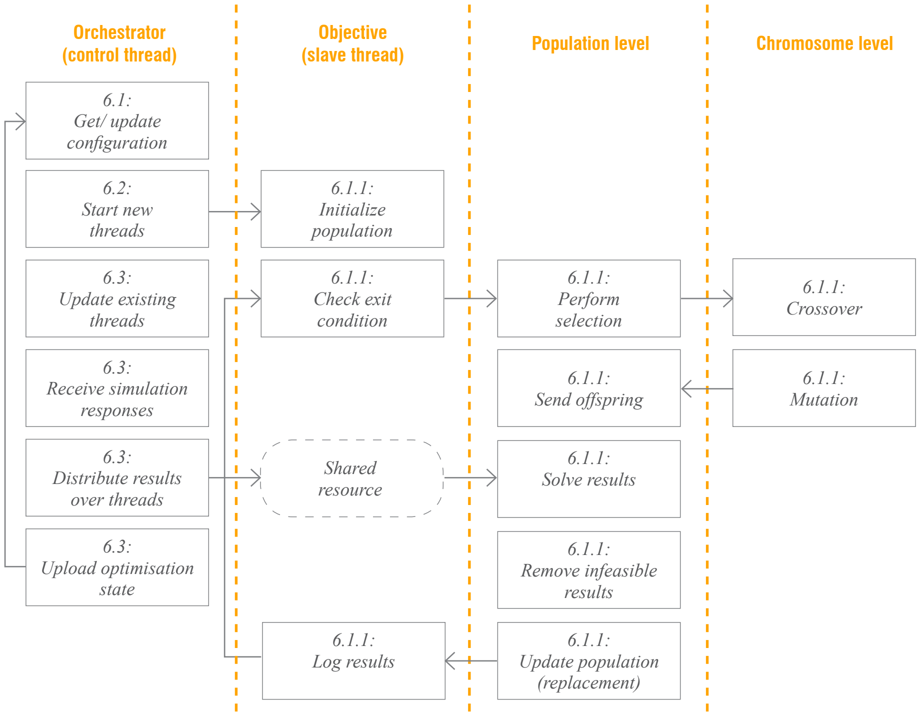

Figure 4. Optimisation process layout and object layers using a control thread to orchestrate the process and variable set of slave threads to perform optimisation on multiple objectives simultaneously.

In the communication between two distributed programs two roles are distinguished: the producer and the consumer. The producer's job is to generate a data item while the consumer processes (and removes) it to perform a task. Exemplary for the system is the optimisation layer which acts as the producer for the generation layer. Within the generation layer producers and consumers are distinguished within aforementioned distribution groups (i.e. the geometry generation consumes jobs of the optimisation producer while subsequently producing input for simulation consumers). Since often producers are producing work faster than the consumers can process it a messaging system between producers and consumers is developed so that the amount of distributed programs in a distribution group can be scaled to a required performance. In order to provide the communication of larger files between producers and consumers a remote data store is used as an intermediate location. Messages reference to specific data items in the data store for consistent distributed processing. This is illustrated in Figure 5 (right image).

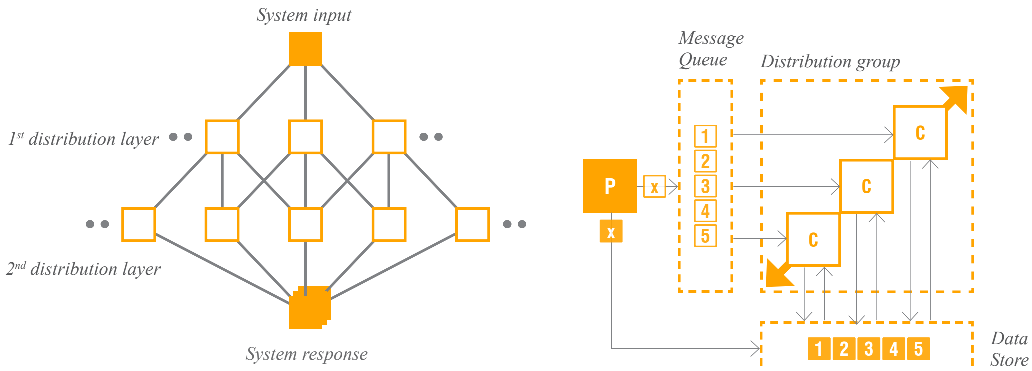

Figure 5. (left) Principle distributed computing process with two distribution groups and a variable amount of distributed programs in each layer (right) Principle distributed computing process with two distribution groups and a variable amount of distributed programs in each layer.

6. Visualisation layer

The visualisation layer feeds the processed information of the system back to the user in a comprehensive overview. For the visualisation of the dataset two graphing techniques are implemented; parallel coordinates (Wegman E.J. [10]) and an objective comparison graph where the user is able to identify Pareto optimal points (Knowles J. & Corne D. [6]). All scenarios are accessible through a web interface where the user can investigate the relationship between outcomes of certain objectives on the configuration of the model variables in multidimensional parallel coordinate graphs. By filtering the visualisation of the dataset based on the selection of Pareto optimal design alternatives in the objective comparison graph the user can identify the influence of that set on the variable configurations in the parallel coordinates view. This is illustrated for a dataset of a multidisciplinary optimisation benchmark of structural (SC) and operational costs (EC) of a classroom building in Figure 6. Based on the findings of such an investigation the user can choose to define additional utility functions in the modelling layer and update the current thread solvers. In this way the designer has the ability to dynamically investigate trade-off between the separate objectives in order to make informed decisions on the optimality of the design scenarios on a multidisciplinary level.

Figure 6. Interactive optimisation interface for investigating scenarios using parallel coordinates and objective graphs to investigate Pareto optimal fronts. Dataset: result of multidisciplinary optimisation benchmark of structural (SC) and operational costs (EC) of a classroom building.

7. Implementation

The system has been based on the parametric design tool Grasshopper (*grasshopper3d.com*), where features were developed in C# in a .NET environment (*microsoft.com/net*). The system uses Oasys GSA (*oasys-software.com*) for structural analysis and EnergyPlus (*apps1.eere.energy.gov*) for building energy performance simulation, where worker scripts (to control these applications remotely) have been developed in .NET and Python scripting language (*python.org*) respectively. A cross platform optimisation script has been developed in Python. The visualisation layer was based on (most) modern browsers and was developed in Javascript (*javascriptsource.com*) using D3.js (*d3js.org*). All system components are implemented using Amazon Web Services (AWS) (*aws.amazon.com*) interface for intercommunication through message queues and data stores. They are implemented in separate frameworks both for Python and C# to allow easy integration of new system components for future extensions of the system.

8. Discussion

The current implementation of the system has been developed as a proof of concept and therefore does not contain as much functionality as by which it was initially envisioned. The system has been set up in a decoupled way to allow the introduction of new system components in the future. In the following paragraphs several limitations as well as suggestions of future developments of the system are discussed.

The the modelling layer is limited to two specific disciplinary simulations. For structural simulation the modelling capability is limited to steel structures and simple loading scenarios, for building energy simulation modelling is limited to basic geometry, day lighting scenarios and default building installation systems.

The optimisation layer implements a parallel algorithm which can be dynamically updated by the user through the objective function definitions in the modelling layer. The designer can evaluate new ideas quickly by investigating the data overview of the current progression of the search (through the visualisation layer) and adjusting the corresponding objective functions to create new scenarios. There is a potential in utilising the designers capacity to identify trends or patterns that become visible between these scenarios to gain a better understanding of trade-off between objectives. Currently the implementation of this dynamic design scenario approach is limited to the creation of either new scenarios and the re-randomizing of the population of existing ones.

During this search a massive amount of data (or history of the design problem) is created. A future vision of the system is to extent it with a central database on which the generated data can be stored. In doing so, two potential improvements for supporting the design process are envisioned. Firstly, by maintaining a history of the earlier evaluated design alternatives, the evaluation of similar design alternatives can be avoided. Secondly, rapid re-calculation of the dataset based on adjustment of the objective functions can be exploited effectively by using a database so as to avoid (expensive) re-evaluation of data set under changing performance conditions. The designer can change the logic of the objective functions and receive near-direct feedback of the influence of these changes on the existing dataset. The current state of the system is set up to allow such technologies to be developed as and extension, however more research and development is necessary of the back-end as well as the visualisation techniques to facilitate such developments effectively.

The visualisation layer currently implements solely the visualisation of data and a static interaction with it. However it might prove useful to support a direct feedback from the user to the optimisation procedure. For instance the adjustment of variable ranges can be updated into the optimisation algorithm directly based on finding of the dataset from the visualisation layer so that the search process can be actively steered into a specific area of the design space or to increase granularity onto a specific area of the design space. Furthermore, is the potential of Rhinoceros as a generator of geometrical representation of the design alternatives not exploited to its full potential in the current state of the system. A rapid render engine that generates images of, for instance a selected Pareto optimal front, could provide a multidisciplinary design team a tool to investigate the aesthetical impact of certain optimal design alternatives. In doing so can the non-quantifiable aspects of the design's performance be effectively integrated in the multidisciplinary analysis and so the decision making of the design team.

9. Conclusions

The system presented in this paper is aimed at supporting the early stages of the design process of multidisciplinary design problems in the AEC industry. By allowing designers of different disciplines to investigate the influence of their design choices on the design outcomes on a multidisciplinary level, informed design choices can be made in an integrated way. By doing so, design changes can be reduced in the later stages of the process, where generally the restricted design freedom requires high costs for making design changes.

Generally in the early stages of the design process design alternatives must be investigated in a limited time frame. This limits the quantity of the total investigated design alternatives and therefore limits the quality of the search towards better or more optimal design alternatives. Furthermore is the required performance or trade-off between multiple disciplinary aspects often unknown to disciplinary parties. The perspective on trade-off is highly contextual and varies between projects and parties. By allowing the designer to rapidly investigate multiple scenarios defined by custom performance functions, a multidisciplinary analysis of satisfactory trade-off scenarios can be investigated and the corresponding set of optimal design alternatives.

This search strategy requires the evaluation of large quantities of design alternatives and the evaluation of their corresponding performance values trough simulation. By providing a fully integrated and distributed system architecture high and scalable performance is be achieved for otherwise time consuming and expensive investigation.

10. References

[1] Brown N., *Evaluation of Multidisciplinary Optimization (MDO) Techniques Applied to a Reusable Launch Vehicle* (Special Project Report) School of Aerospace Engineering Space Systems Design Laboratory, Georgia Institute of Technology, 2004

[2] Crisfield M.A., *Non-linear Finite Element Analysis of Solids and Structures. Volume 2: Advanced Topics.* (2nd ed.), Wiley, 1997.

[3] EnergyPlus, *Energy Efficiency & Renewable Energy website, http://apps1.eere.energy.gov/,* accessed 27 May 2014

[4] Grossman B. et al., *Integrated Aerodynamic/Structural Design of a Sailplane Wing.* Journal of Aircraft, 25(9):855-860, 1988

[5] Kipouros T. et al., *Multi-Objective Aerodynamic Design Optimisation* Integrated Design Consortium, University of Cambridge, Department of Engineering, Cambridge, 2006

[6] Knowles J. & Corne D., *Approximating the non-dominated front using the Pareto Archived Evolution Strategy* Department of Computer Science, University of Reading, UK, 1999

[7] Lambe A.B. & Martins J.R.R.A, *Unified Description of MDO Architectures* 9th World Congress on Structural and Multidisciplinary Optimization Shizuoka, Japan, 2011.

[8] Lambe A.B. & Martins J.R.R.A., *Multidisciplinary Design Optimization: A Survey of Architectures* American Institute of Aeronautics and Astronautics (AIAA) Journal, 2013.

[9] 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.* American Institute of Aeronautics and Astronautics (AIAA) Journal, 2012

[10] Wegman E.J., *Hyperdimensional Data Analysis Using Parallel Coordinates.* Journal of the American Statistical Association Vol. 85, No. 411 (Sep., 1990), pp. 664-675, 2004