D4.1.2. Release of Workflow (re-) formulation tool(s)

Author, company:

- Jente Sonneveld, TU Delft
- Gianfranco La Rocca, TU Delft
- Costijn de Priester, TU Delft
- Max Rein, TU Delft
- Tobie van den Berg, GKN Fokker Aerostructures
- Bastiaan Beijer, KE-works

Version:
1.0

Date:
March 27, 2023

Status:
Final / Released

Confidentiality:
Public

Change log

Revision	Date	Prepared by	Checked by	Description
0.1	27/10/2022	Jente Sonneveld	Bastiaan Beijer	First setup document
0.2	11/01/2023	Jente Sonneveld	Gianfranco la Rocca	Added all new KADMOS advisory developments
0.3	17/03/2023	Jente Sonneveld	Gianfranco la Rocca	Added KADMOS/CMDOWS aided architecture optimization
1.0	27/03/2023	Jente Sonneveld	Max Baan	Incorporate final revisions for delivery

Table of Contents

Acronyms

Acronym	Definition
BLISS	Bilevel Integrated System Synthesis
CMDOWS	Common MDO Workflow Schema
CO	Collaborative Optimization
CPACS	Common Parametric Aircraft Configuration Schema
DEFAINE	Design Exploration Framework based on AI for froNt-loaded Engineering
DOE	Design Of Experiments
FPG	Fundamental Problem Graph
IDF	Individual Discipline Feasible
KADMOS	Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System
MDA	Multidisciplinary Design Analysis
MDAO	Multidisciplinary Design Analysis and Optimization
MDF	MultiDiscipline Feasible
MDG	MDAO Data Graph
MDO	Multidisciplinary Design Optimization
MPG	MDAO process Graph
PIDO	Process Integration and Design Optimization
RCE	Remote Component Environment
RCG	Repository Connectivity Graph
SAS	Surrogate Advisory System
SM	Surrogate Model
VISTOMS	VISualization TOol for MDO Systems
WP	Work Package
XDSM	eXtended Design Structure Matrix
XML	eXtensible Markup Language

Acknowledgements

This research is partly funded by the ITEA 3 Call 6 project DEFAINE of the European Union.

References

[1] DEFAINE Consortium, “D4.3.1. Technology demonstrator for use-cases,” ITEA3 Call 6, 2022.

[2] DEFAINE Consortium, “D4.3.2. Second release of the techonlogy demonstrator for use-cases,” Itea3 Call 6, 2022.

[3] DEFAINE Consortium, “D4.3.3. Final release of the techonlogy demonstrator for use-cases,” Itea3 Call 6, 2023.

[4] AGILE, “AGILE Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts,” H2020-EU.3.4., [Online]. Available: https://www.agile-project.eu/. [Accessed 23 12 221].

[5] AGILE4, “AGILE4.0 Towards cyber-physical collaborative aircraft development,” H2020 No 815122, [Online]. Available: http://agile4.eu. [Accessed 23 12 2021].

[6] DEFAINE Consortium, “Full Project Proposal,” Itea3 Call 6, 2020.

[7] DEFAINE Consortium, “D2.2.1 Industrial Needs and Framework Requirements,” ITEA3 Call 6, 2022.

[8] DEFAINE Consortium, “D4.1.1. First release of Workflow (re-)formulation tool(s),” ITEA3 Call 6, 2022.

[9] DEFAINE Consortium, “D4.1.3. Final release of workflow (re-) formulation tool(s),” ITEA3 Call 6, 2023.

[10] I. van Gent and G. la Rocca, “Formulation and integration of MDAO systems for collaborative design: A graph-based methodological approach,” *Aerospace Science and Technology,* vol. 90, pp. 410-433, 2019.

[11] I. van Gent, G. La Rocca and M. F. M. Hoogreef, “CMDOWS: a proposed new standard to store and exchange MDO systems,” *CEAS Aeronautical Journal,* vol. 9, p. 607–627, 2018.

[12] B. Aigner, I. Van Gent, G. La Rocca, E. Stumpf and L. L. M. Veldhuis, “Graph-based algorithms and data-driven documents for formulation and visualization of large MDO system,” *CEAS Aeronautical Journa,* vol. 9, pp. 695-709, 2018.

[13] I. van Gent, R. Lombardi, G. la Rocca and R. d'Ippolito, "A Fully Automated Chain from MDAO Problem Formulation to Workflow," in *EUROGEN 2017*, Madrid, 2017.

[14] DEFAINE Consortium, "D4.2.1. CMDOWS," ITEA3 Call 6, 2022.

[15] C. de Priester, "Towards the integration of surrogate models in automated MDAO workflow formulations," Literature Review, TU Delft, Delft, 2021.

[16] C. d. Priester, "SAS: an advisory framework to support surrogate model based MDAO," Master Thesis, TU Delft, Delft, 2022.

[17] M. Kaufman, V. Balabanov, A. Giunta, B. Grossman, W. Mason, S. Burgee, R. Haftka and L. T. Watson, "Variable-complexity response surface approximations for wing structural weight in HSCT design,," *Computational Mechanics,* vol. 18, 1996.

[18] L. Jia, J. Alizareth, J. Hao, G. Wang, K. Allen and F. Mistree, "A rule-based method for automated surrogate model selection," *Advanced Engineering Informatics,* vol. 45, 2020.

[19] D. Jones, M. Schonlau and W. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization,* vol. 13, 1998.

[20] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier and J. Martins, "A Python surrogate modeling framework with derivatives," *Advances in Engineering Software,* 2019.

[21] M. Rein, "Adjustment of the MDAO problem formulation using sensitivity analysis to reduce the computational cost within aircraft design," Master Thesis, TU Delft, Delft, 2022.

[22] M. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics,* vol. 33, 1991.

[23] A. Saltelli and R. Bolado, "An alternative way to compute fourier amplitude sensitivity test (FAST)," *Computational Statistics & Data Analysis,* vol. 26, no. 4, 1998.

[24] N. Gigliolo and A. Saltelli, "Simlab 2.2 reference manual," Institute for Systems Informatics and Safety, 2008.

[25] R. Rosolem, H. V. Gupta, W. J. Shuttleworth, X. Zeng and L. G. Gonçalves de Gonçalves, "A fully multiple-criteria implementation of the Sobol· method for parameter sensitivity analysis," *Journal of Geophysical,* 2012.

[26] S. Valencia Ibáñez, "A Review of System Architecture Design and Mixed-Discrete, Multidisciplinary Multi-Objective Design Optimization," Literature Review, TU Delft, Delft, 2023.

[27] DEFAINE Consortium, "D2.1.1. Use-case specifications," ITEA3 Call 6, 2021.

[28] M. Rein, “Improving MDAO problem formlations to speed up the development process,” Literature Review, TUDelft, Delft, 2021.

[29] DEFAINE Consortium, “D4.1.2. Second release of Workflow (re-)formulation tool(s),” ITEA3 Call 6, 2022.

1. Introduction

Work package (WP) 4 aims to achieve automated (re-)formulation of workflows. This deliverable describes the released workflow (re-)formulation tool(s). In an industrial environment, the setup of Multidisciplinary Design and Optimization (MDAO) workflows usually requires manual intervention, which is time and cost intensive, as well as prone to human errors. This frustrates the ideal MDAO process, where designers are typically interested in formulating first simple Multidisciplinary Design Analysis (MDA) workflows, then performing Design of Experiments (DOEs) and sensitivity studies to identify relevant design parameters, before moving to actual optimization and eventually iterate on the previous steps, i.e. to add/remove design variables and constraints, test different objectives, add/substitute some of the analysis tools. Previous efforts in the Idealism, AGILE [1] and AGILE4.0 [2] projects have produced methodologies and tools to provide some of the required agility in MDAO workflow (re-)formulation and execution. However, dynamic re-formulations of workflows during the design-process are not yet possible using State-of-the-Art methods and tools.
By “dynamic re-formulation”, we intend the capability of a given workflow to reformulate itself, based on measured characteristics of the initially formulated MDAO system, such to improve its performance. As DEFAINE aims at performing extensive design space explorations, dynamic re-formulations capabilities are pursued to drastically reduce the computational time by means of various approaches. For example, by sequencing the tools in the MDA workflow such to minimize feedback loops; by partitioning tools such to exploit parallel computing capabilities; by replacing individual (or sets of) tools with on-the-fly generated surrogate models; by reducing the number of design variables and eliminate constraints based on sensitivity information; by selecting the most efficient combination of tools based on their level of fidelity, license availability, computational time, etc.

During the DEFAINE project the workflow (re-) formulation methodologies and tool(s) are developed and released in three cycles, aligning with the release of the technology demonstrators described in D4.3.1 [1], D4.3.2 [2] and D4.3.3 [3]. This will be done by further developing the open-source software KADMOS for MDAO system formulations developed in the AGILE [4] and AGILE4.0 [5] projects. In the DEFAINE Full Project Proposal [6] the following capabilities are specified to be developed in this project:

- Provide the means to perform gap analyses based on the available set of engineering competences for given design studies (i.e.: can all expected parameter sensitivities be covered in the workflow).
- Create an advisory system for selecting the most fitting set of engineering competences (among similar alternatives) while considering the required level of fidelity, license availability, budget and computation time targets. A strategy will be developed to evaluate whether surrogate model generation for replacing expensive analysis services, is possible and pays off for given execution time constraints.
- Dynamic re-formulation capabilities to react on design space exploration and data analysis results (from WP5) while running the simulation (e.g., addition or removal of design variables and constraints, change of architecture, parallelization of workflows).

The tool released in this cycle is the second release of this deliverable. In Table 1, all releases of this deliverable are shown. It must be noted that the industrial needs and framework requirements described in D2.1.1 [7] will influence the level of priority given to certain capabilities and the potential inclusion of additional developments, which will be discussed in the multiple releases of this deliverable.

Table 1: Releases of workflow (re-)formulation tools deliverables.

Deliverable	Description	Due
D4.1.1	State of the Art	M16 (delivered)
D4.1.2	Extend the existing advisory capabilities **(this deliverable)**	M28
D4.1.3	Enable dynamic reformulation capabilities	M40

1.1. Intended use and purpose of this deliverable

This deliverable is of type “software”. The purpose of this deliverable is to provide an overview of all the developments since the release of the previous version of this deliverable (D4.1.1.). The developments described in this deliverable include:

- KADMOS advisory module for surrogate modelling
- KADMOS advisory module for sensitivity analysis
- KADMOS aided dynamic workflow (re-) formulation for architecture optimization

For a detailed description of the workflow (re-) formulation tool KADMOS please refer to D4.1.1. [8]. Chapter 2 describes the developed KADMOS surrogate modelling and sensitivity analysis modules. In Chapter 4, the initial steps towards a KADMOS aided strategy for dynamic (re-)formulation of workflows to enable multi-architecture optimization is presented. In Chapter 5 further planned developments towards deliverable D4.1.3 [9] are detailed. Finally, in Chapter 5 the overall conclusions are presented.

2. KADMOS Advisory Developments

In this section, the developments related to KADMOS advisory capabilities since the previous release of this deliverable [8] will be discussed. These include the development of a surrogate advisory system and an advisory system based on sensitivity analysis.

2.1. Surrogate Advisory System (SAS)

The Surrogate Advisory System (SAS) is an extension to the KADMOS tool described in D4.1.1 [8], specifically developed in the context of DEFAINE. SAS is developed to provide an MDAO architect with a tool to quickly assess a provided MDAO workflow and give advice on a strategy for replacing one or more disciplines with a surrogate model (SM). Implementation wise, SAS is a Python based package linked to KADMOS and works with CMDOWS files for handling the (re-)formulation process of workflows. SAS has an interface to the PIDO tool RCE enabling the implementation of the provided advice through automatic materialization of the re-formulated workflows. A detailed description of this work can be found in [15] [16]. In Figure 1, an activity diagram is shown of a typical advice request and implementation process as enabled by SAS. In this figure the interaction between SAS, KADMOS and the PIDO tool are indicated. In the following sections, the workings of this advisory module are presented in more detail.

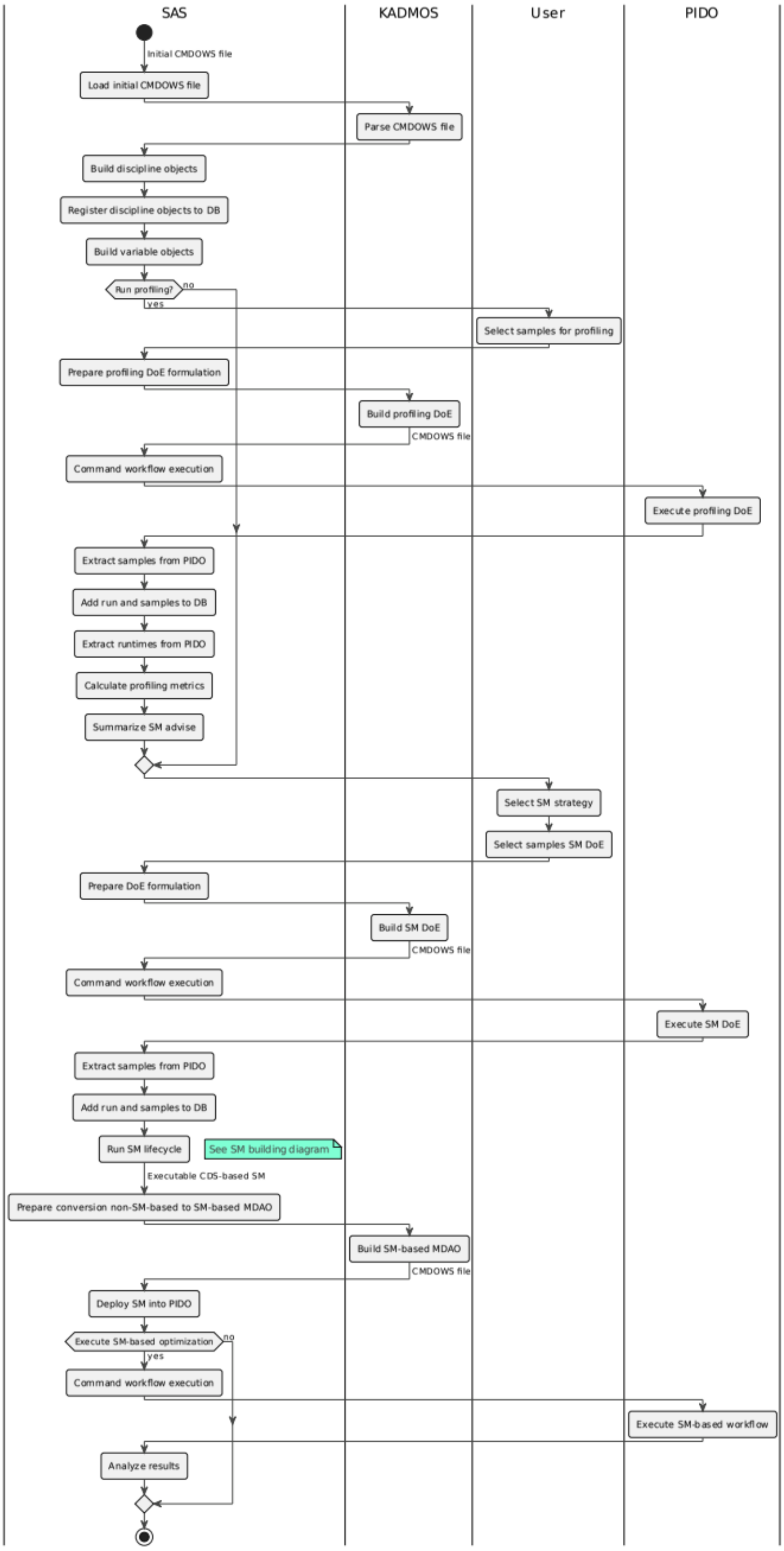


Figure 1: Activity diagram of a typical SAS lifecycle [16].

2.1.1. Assessment of provided workflow

An optimization workflow must be provided to SAS in the form of a CMDOWS file. An exploratory run can be executed in order to determine the relative computational cost of each discipline. To achieve this, KADMOS is used to convert an optimisation workflow into a DoE with a small number of experiments with varying the input values. The results of the exploratory run are used to assess the relative run time for each design competence in the workflow. In Figure 2 an example of the results of such an exploratory run is shown, where it can be seen that the 'AeroAnalysis' discipline is responsible for most computational expense. Replacing the AeroAnalysis discipline with a SM can save time each iteration, however constructing the SM itself may be a large investment as well. Clearly, it would be beneficial to have a means of providing the user with advice on possible SM strategies (design competence, or group of design competences to be replaced with SM) and their consequences, accounting both for the expected time gain by using SMs and the expense to generate such SMs.

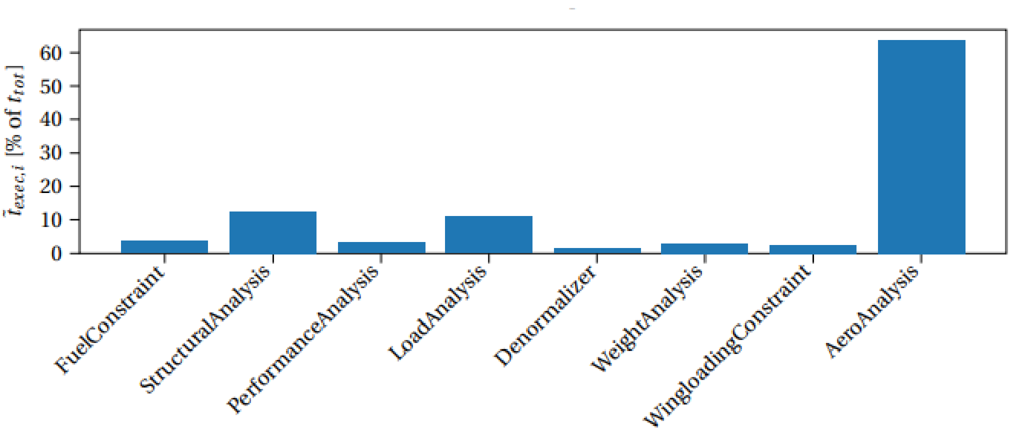


Figure 2: Normalized runtimes for an example MDAO problem, generated using a DoE and Ns = 4. [16]

2.1.2. Provide advice on surrogate model strategies

Before any advice can be given, feasible surrogate modelling (SM) strategies need to be identified, where by SM strategy we intend a possible discipline or combination of disciplines, that can be replaced by a surrogate model. SAS automatically identifies valid strategy options that adhere to the following rules:

1. Can contain a driver (I.e. a converger), only if the complete nested loop is included in the SM strategy
2. Cannot "break out" of a nested loop without including the complete nested loop
3. Cannot contain a DoE driver or the most-outer-level optimization driver

Once all valid surrogate strategy options are identified, an estimation of the time investment required per sample for each surrogate option can be made. This is based on the combined or individual disciplines input vector size and run-time estimates from the exploratory run. In literature, three rules of thumb emerge (see Table 2) that provide an estimation of the required number of samples for 'reasonable' surrogate model accuracy. These rules of thumb are based on the number of input variables a discipline or group of disciplines has. In Figure 4, the advice for the SM strategies indicated in Figure 4 is shown. The advice is provided in the form of a plot where the time investment to implement the identified surrogate strategies is plotted against an 'implied accuracy'. This implied accuracy is a measure of the amount of samples used to train the surrogate model, where 100% is the minimum amount of samples prescribed by the afore mentioned rules of thumb. In addition, a red dashed line indicates the estimated time for performing the optimization using the original workflow. This is based on a user-defined estimate of the amount of iteration required and can be inaccurate as this information might not be available a priori.

Table 2: Rules of thumb from literature to determine the required amount of samples for a reasonably accurate surrogate model (N_s = number of samples, N_v = size of input vector)

Method	Formula
Kaufman [17]	$N_s = \frac{3}{4}(N_v + 1)(N_v + 2)$
Jia [18]	$N_s = (N_v + 1)(N_v + 2)$
Jones [19]	$N_s = 10N_v$

Based on the received advice, the MDO engineer can make a decision about what surrogate strategy to go for. In case of the example plotted in Figure 4, assuming an implied accuracy of 100% is desired, it can be seen that the strategy SM5 is above the red dashed line. This means that the combined time investment of constructing the surrogate model + the actual optimization (using the surrogate in the workflow) is larger than optimizing using the original workflow. The other strategies lie below the -dashed line, meaning a total time optimization reduction is expected by the generation and utilization of surrogates. It must be noted that this advice assumes a single optimization problem execution. In many cases, multiple optimization problems may be run using the same set of disciplinary tools, thus better justifying the time invested in surrogate model generation.

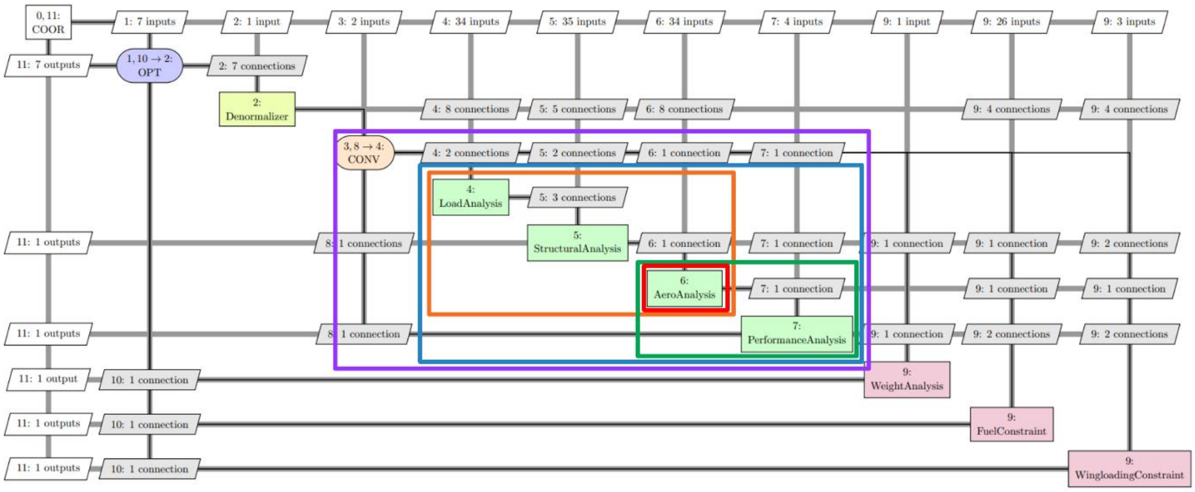


Figure 3: MDAO problem XDSM including the 5 SM strategies.

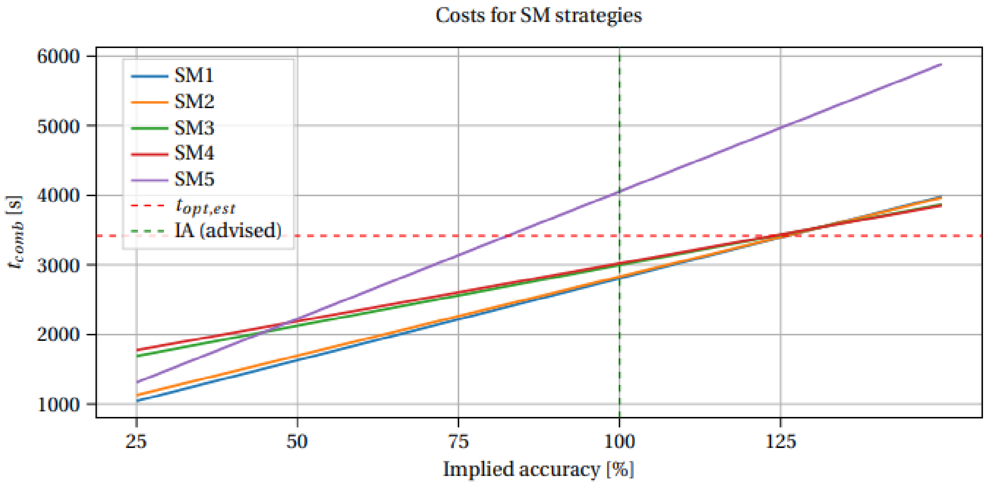


Figure 4: SM strategy advise visualisation for the MDAO problem formalized in Figure 5. [16]

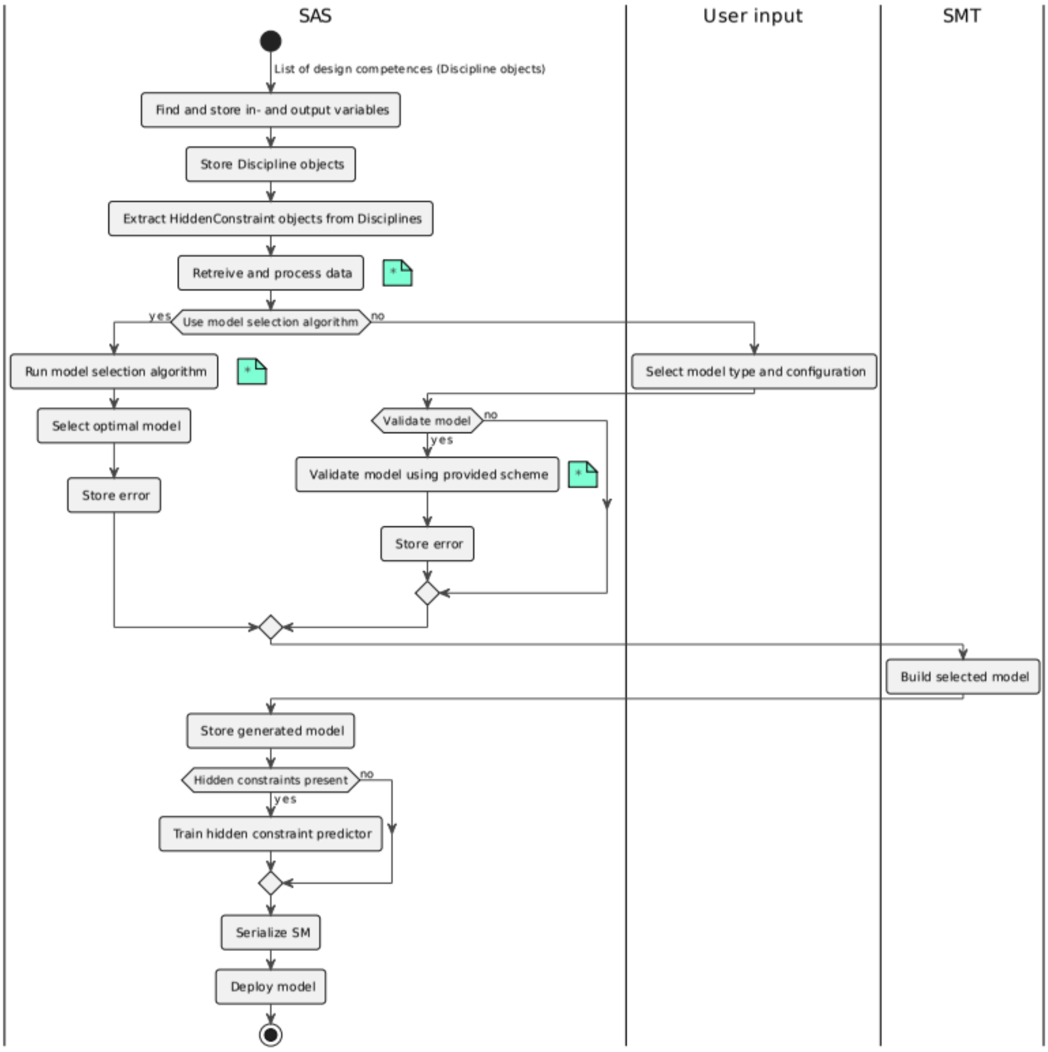


Figure 5: Activity diagram of the normal SM building procedure as implemented in SAS [16].

2.1.3. Generate samples and build surrogate model

A sampling plan can be generated for the selected SM strategy. KADMOS is used to automatically create a DoE workflow with the discipline(s) to be replaced by a surrogate model, according to the selected surrogate strategy. The DoE can be executed in RCE and the results are extracted and saved into a database. For training the actual surrogate model, SAS makes use of the Python-based open-source Surrogate Modelling Toolbox[1] (SMT) [20] . SMT is capable of training various types of surrogate models based on a list of samples. Available surrogate modelling options are shown in Table 3. In Figure 5 the activity diagram of the surrogate building procedure is shown. In this figure the interaction between SAS, the user and SMT is indicated.

The surrogate model training itself generally takes an insignificant amount of time compared to generating the samples. Therefore, SAS enables (in addition to just specifying a specific SM method) all available surrogate model types to be trained and compared to assess the best performing method for a particular case. There are a number of metrics that can be used to compare the performance of a surrogate model, available metrics and methods to evaluate these metrics are listed in Table 3. The SM that has the lowest error will be selected and used.

Table 3: Available options for surrogate modelling methods, validation techniques and error metrics by SAS.

Surrogate modelling methods	Surrogate model validation techniques	Surrogate model error metrics
Radial Basis functions	K-fold	RMSPE
Inverse-distance weighting	Leave-one-out	RMSE
Least-squares approximation	Split-sample	NRMSE
Second-order polynomial approximation		R2
Kriging		
KPLS		
KPLSK		
GEKPLS		

2.1.4. Deploy surrogate model

Once the surrogate model has been trained, it can automatically be saved as a Python Pickle file. This file, in combination with the location of where input file and output file are stored, can be executed from the command line, making it readily available for other applications. SAS automatically integrates the surrogate model into RCE as a design competence. KADMOS is used to formulate an optimization workflow with the surrogate model integrated. Figure 6 shows an example of a workflow where the complete convergence loop has been replaced with a SM, automatically generated and integrated into RCE.

[1] https://github.com/SMTorg/smt

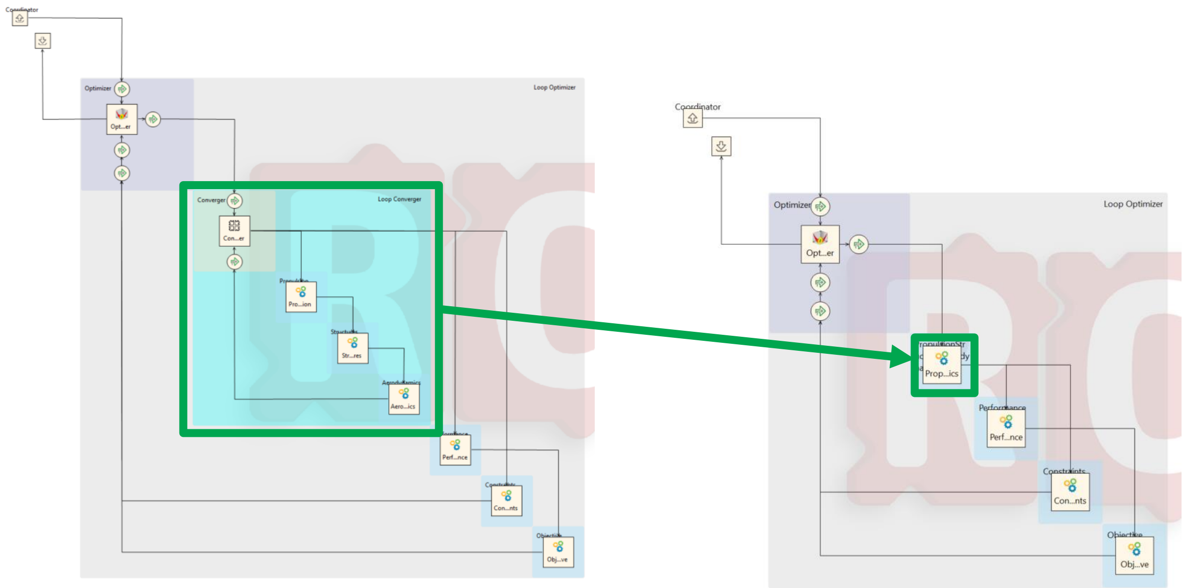


Figure 6: Group of design competences replaced by a surrogate model in RCE by SAS.

2.1.5. Analyse and improve surrogate model

It must be noted that the advice graph shown in Figure 4 is based on estimations for both the runtime of disciplines and the required amount of samples for 'reasonable' accuracy. The actual surrogate performance and time investment can vary and additional samples may be required to achieve satisfactory results.

SAS assesses the accuracy of the constructed surrogate model using the error metrics in Table 3. If the surrogate accuracy is not satisfactory, SAS can propose infill samples, using the Expected Improvement for Global Fit (EIGF) method. The EIGF algorithm is an adjustment of the Expected Improvement (EI) algorithm and aims to improve the global accuracy of the model rather than the accuracy of the surrogate around a specific design point. Once again, KADMOS is used to construct a DoE with the proposed infill samples and the model is retrained. The error metric is re-evaluated and the process is repeated until a satisfactory value of the model accuracy is achieved.

In Figure 7, an example of the actual time investment and surrogate error for different implied accuracies are shown. The data corresponds to SM1 in [16]. The actual time investment is higher compared to the predicted value shown in Figure 4.

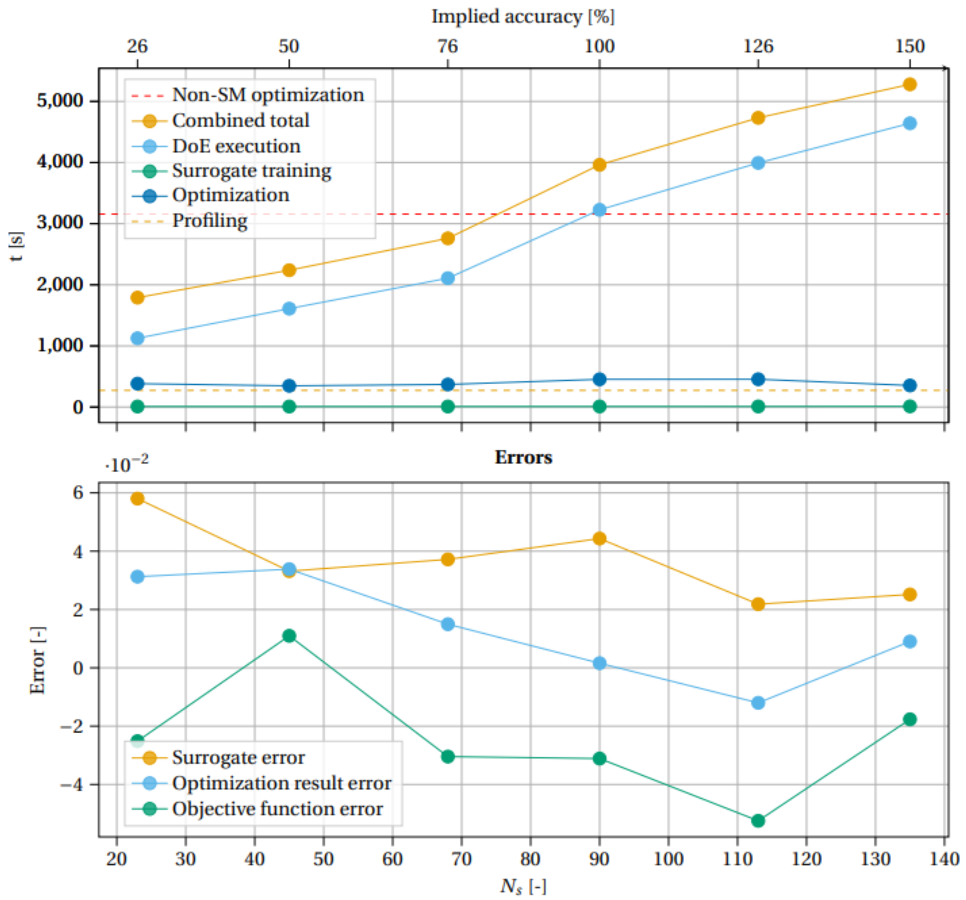


Figure 7: Results of implementation of SM1. [16]

2.2. Sensitivity analysis to aid MDAO formulation

TU Delft further extended KADMOS capabilities by means of a sensitivity analysis module, aimed at reformulating MDAO workflows to improve computational speed. [21]. Sensitivity data, obtained through sensitivity analysis, provide valuable information on how changes in input variables impact the output variables of a system. On a global level, this information can be leveraged to provide an advice on the removal of design variables that have very little influence on the objective functions. On a local level, sensitivity information can reveal the sensitivity of couplings between design competences. This information can be used to improve KADMOS' existing sequencing and decomposition algorithms. Both the global and local approaches are explained in greater detail below. Both the global and local approaches are explained in greater detail below.

2.2.1. Global sensitivity methods to assess influence of design variables

In Figure 8, the process of using global sensitivity analysis to identify the sensitivity of design variables of a MDAO problem is presented. In an initial formulation phase, the MDAO problem is set up using KADMOS and saved as a CMDOWS file. This workflow is used to generate samples and perform sensitivity analysis; based on the results, the problem formulation can be adapted and the final optimization executed.

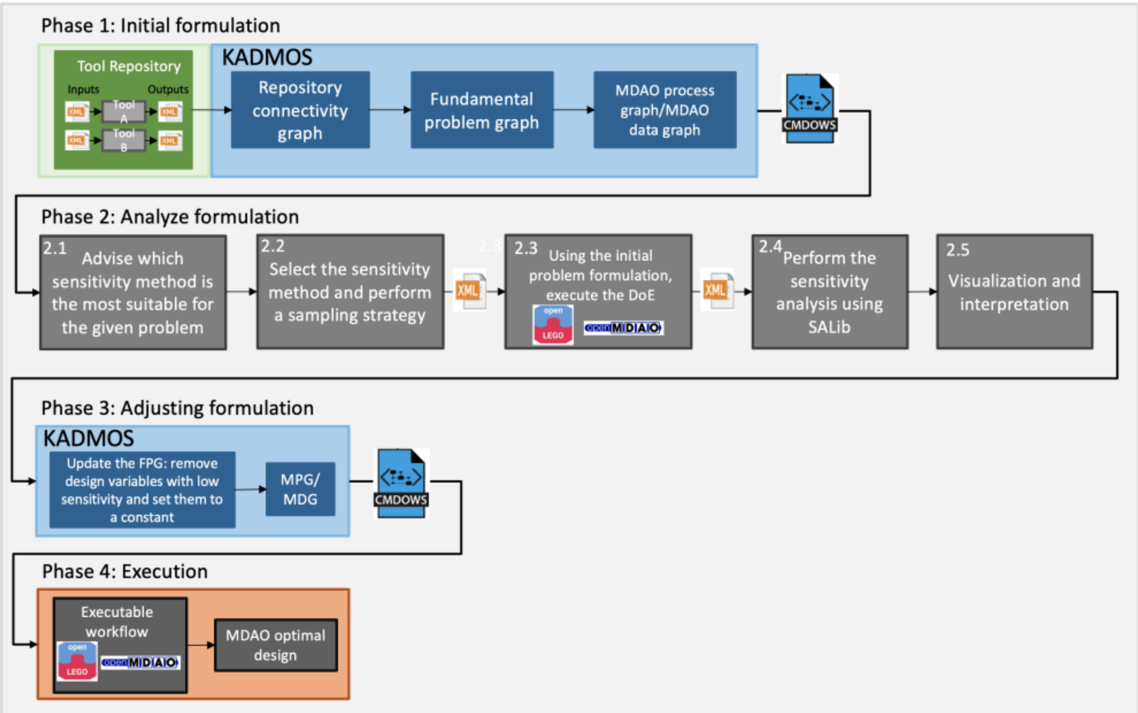


Figure 8: KADMOS MDAO workflow formulation process including global sensitivity analysis to identify non-influential design variables [21].

For making sampling plans and performing global sensitivity analysis, the SALib[2] python package is used. SALib provides Python implementations of commonly used sensitivity analysis methods. In Table 4, the implemented sensitivity analysis methods are shown. In this table, an estimation of the required amount of function evaluations based on the amount of input variables, compatible sampling method and relative measure of accuracy are provided for each method.

Table 4: Implemented global sensitivity methods including required amount of function evaluations, sampling methods and relative metric of accuracy (N_S = number of samples, N_v = size of input vector)

Method	Req. function evaluations [21]	Compatible sampling method	Accuracy [21]
Sobol [22]	$N_s = 32(2N_v + 2)$	Sobol sequence/LHS	High
FAST [23]	$N_s = 65N_v$	LHS	Highest
Morris [24]	$N_s = 10(N_v + 1)$	Morris sampling	Least

The Sobol and FAST methods are both ANalysis Of VAriance (ANOVA) models, where the observed variance in a particular output variable is partitioned into components attributable to the different sources of variation. The sensitivity of an input variable on the outcome can be aggregated into a total effect index. Figure 9An example of a bar plot showing the total effect index (and the 1st order index) for a problem with 4 input variables is shown in Figure 9A. For the Sobol and FAST methods, it implies that if a design variable has a total sensitivity index of less than 0.01 (1%), the variable can be considered non-influential [25]. Both these methods are compatible with Latin Hypercube Sampling, however, the Sobol method is proven to be more accurate using samples generated by the Sobol sequence.

The Morris method investigates how the output responds to a change in the inputs by varying one input at a time, resulting in the elementary effect of each input variable. This requires a very specific sampling and is therefore not compatible with any random sampling technique. The Morris method aggregates the sensitivity of an input variable into ‘μ*’, the mean of the absolute value of the elementary effects and ‘σ’, the standard deviation of the elementary effects. The larger the value of μ*, the more the input variable influences the model output. On the other hand, the higher the σ value, the more non-linear the input is or the more it interacts with other inputs, while a low σ indicates a linear, additive input. In a Morris plot, the μ* and σ of each input variable are plotted against each other, as seen in Figure 9B. For the Morris method there is no fixed threshold value; however, a threshold is usually set as 5-10% of the maximum μ∗ value [25].

[2] https://github.com/SALib/SALib

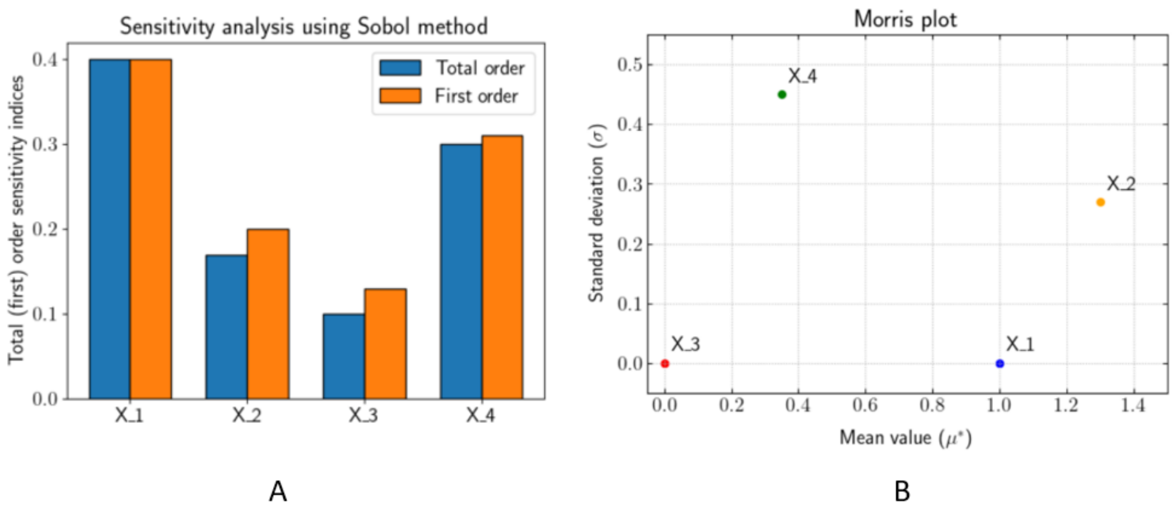


Figure 9: Example of bar plot showing total and first order sensitivity indices of input variables (A) and Morris plot of mean value versus standard deviation of input variables (B) [21].

Based on the plots in Figure 9 and the advised threshold values a decision can be made on removing certain non-influential design variables. In [21] global sensitivity analysis using the Morris method is applied to an aircraft design MDAO problem. Based on the sensitivity analysis, 2 non-influential design variables were identified and removed from the problem formulation. This proved to reduce the optimization time by 40% while the found objective stayed within 0.5% of the original optimization problem. However, including the time spent to generate samples and perform sensitivity analysis, an increase in total time for a single optimization can be observed. In many cases, multiple optimization problems may be run using the same set of disciplinary tools, thus better justifying the time invested in surrogate model generation.

2.2.2. Local sensitivity methods to improve sequencing and decomposition algorithms

As discussed in section 2.1.1, KADMOS has sequencing algorithms to minimize the number of coupling variables in a given MDA workflow. Instead of using the number of feedback variables, local sensitivity information can be used instead. It is expected that if the sensitivity of feedback variables is lower, the number of iterations required for convergence is reduced. In Figure 10, an example problem XDSM with 4 disciplines all having one output variable is illustrated. A finite-difference method can be used to determine the partial derivative of the output with respect to the input for each discipline. This sensitivity information is shown in the figure as well. The existing KADMOS sequencing algorithms require a coupling dictionary as shown in Figure 11a. Instead of using the number of feedback variables, the sensitivity information can be provided as shown in Figure 11b. In Figure 12, the resulting sequencing for both methods are presented. It can be seen that in case the sensitivity data is used, more coupling variables are present, however, the combined sensitivity is lower compared to the solution provided by the original method.

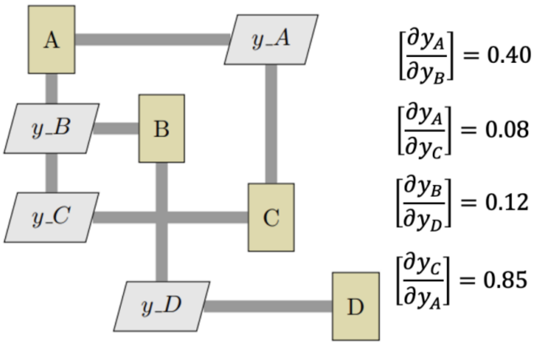


Figure 10: Example problem consisting of four disciplines with the corresponding sensitivity information [21].

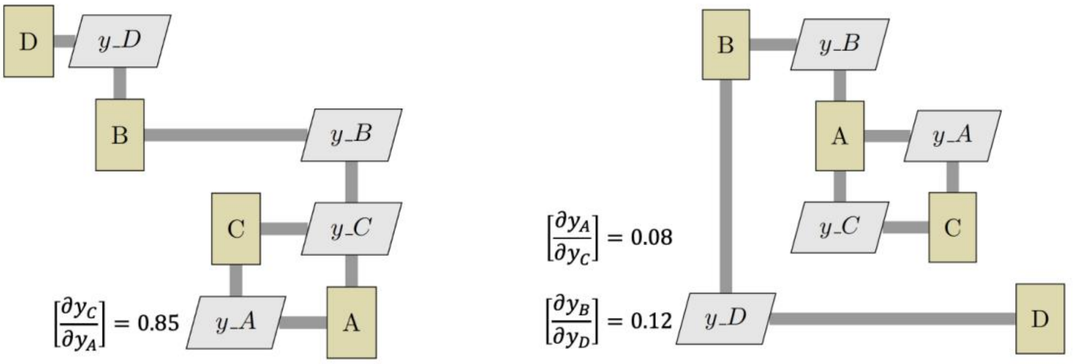
A: {B: 1, C: 1,},
B: {D: 1},
C: {A: 1},
D: {}

a) Original coupling dictionary with number of variables

A: {B: $\frac{\partial y_A}{\partial y_B}$, C: $\frac{\partial y_A}{\partial y_C}$},
B: {D: $\frac{\partial y_B}{\partial y_D}$},
C: {A: $\frac{\partial y_C}{\partial y_A}$},
D: {}

b) Modified coupling dictionary with sensitivity information

Figure 11: Example of coupling dictionary with (a) number of feedback variables and (b) sensitivity information [21].



(a) Sequence for minimum number of feedback variables

(b) Sequence obtained when considering sensitivity of feedback variables

Figure 12: Comparison of the sequencing solution for two different approaches [21].

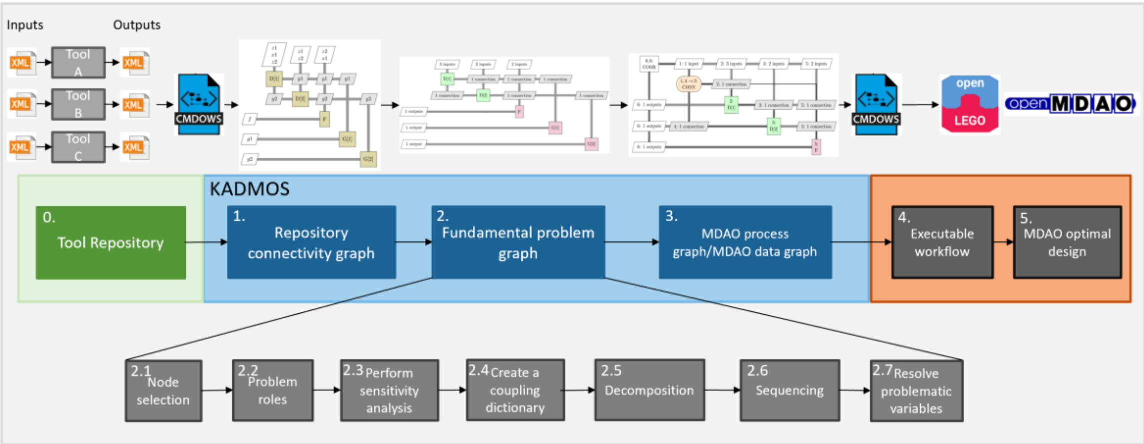


Figure 13: Overview of the workflow for implementing sensitivity analysis within the sequencing and decomposition algorithms.

In Figure 13, an overview of the methodology is illustrated. When creating the FPG two new steps are introduced: perform local sensitivity analysis and create a coupling dictionary. In [21] finite difference local sensitivity methods were applied to a variable complexity problem. The sensitivity data enriched KADMOS' sequencing and decomposition algorithms were used to reformulate the workflow. It was observed that, a reduction of 9% compared to the original sequencing algorithms based on the number of feedback variables.

2.3. Combining capabilities in one module

Due to the similarity of actions required for both surrogate modelling and sensitivity analysis, both developed capabilities can be combined in a single toolset. This means no unnecessary duplicate actions are required like generating sampling plans executing DOE's and acquiring outputs. In addition, existing sample sets can be reused. I.e., samples generated for sensitivity analysis can be used to generate a surrogate model and vice versa. To achieve this, the SAS workflow analysis module is expanded with an interface to SALib's sensitivity analysis capabilities.

3. KADMOS aided dynamic workflow (re-)formulation to enable architecture optimization

Consolidated strategies and optimization algorithms can be used to perform multidisciplinary and multi-objective optimization of engineering products, as far as the architecture of the system to be optimized is fixed. Accounting for the complete system architecture design space in an optimization process is very challenging due to the mixed, dynamic and hierarchical nature of the involved design variables. Some are categorical, some integer, some continuous; some depend, in number or existence, on the value of other variables. Take for example an aircraft moveable design: design variables like 'number of ribs' will influence the number and physical dimensions of material zones at which the thickness must be optimized. In other words, the full design vector only becomes known after some higher level design variables are set, as illustrated in Figure 14.

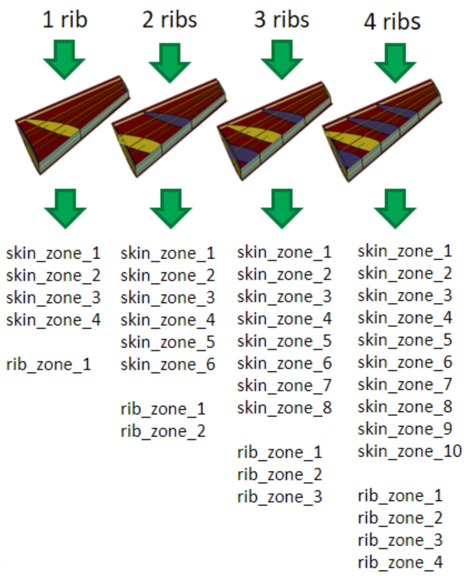


Figure 14: Example where the value of one variable governs the presence and quantity of other variables.

TU Delft completed a literature review on system architecture design optimization strategies in [26]. A way to deal with this hierarchy in the design vector is to implement a nested optimization strategy. An outer loop evaluates the design variables governing the architecture of the product. Based on this the remainder of the design vector is determined, which is in turn evaluated in an inner -nested- loop. In this chapter, an approach is presented to achieve a nested architecture optimization implementation supported by KADMOS and CMDOWS. First, a means to configure a hierarchical design study is presented.

3.1. Design Study Configuration file

As a means to configure a hierarchical design study, the XML based Design Study Configuration (DSC) file format was developed. In a DSC file, a number of nested design steps can be configured, an example with two design steps is illustrated in Figure 15. The variables specified in a nested design step depend on the variables configured in higher level design steps, this way the hierarchy of the problem is configured. In principle there is no limit on the amount of nested design steps that can be specified. The DSC file format is inspired by the CMDOWS standard and uses the same format when specifying parameters and problem formulations.

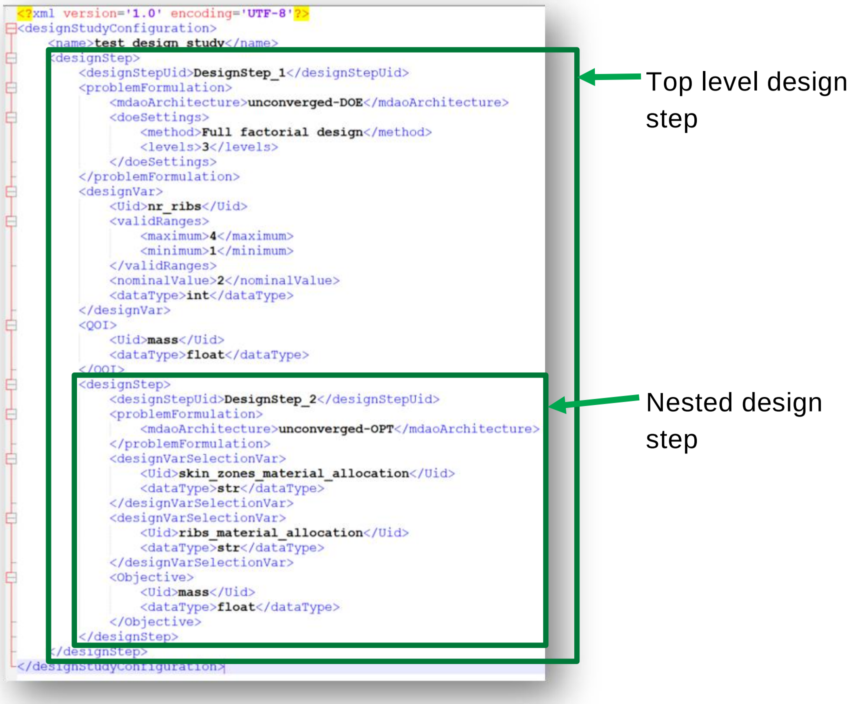


Figure 15: Example of Design Study Configuration file with nested design step

Within each design step it is mandatory to specify a '**designStepUid**' and '**problemFormulation'**. The problem formulation governs what type of MDAO will be performed in the design step, i.e. a design of experiments or an optimization. Apart from the above mentioned elements, within each design step, a number of parameter types can be specified as well:

- **designVar:** design variable
- **designVarSelectionVar:** design variables whose type and quantity are not known a priori, but depend on the value of design variables in a lower design step. Cannot be specified in a top-level design step.
- **Constraint:** constraint variable
- **ConstraintSelectionVar:** a constraint, of which the type and quantity is not yet known and depends on the value of design variables in a lower design step. Cannot be specified in a top-level design step.
- **Objective:** objective variable

- **ObjectiveSelectionVar:** an objective variable, of which the type and quantity is not yet known and depends on the value of design variables in a lower design step. Cannot be specified in a top-level design step.
- **QOI:** Quantity of interest variable
- **QOISelectionVar:** a quantity of interest variable, of which the type and quantity is not yet known and depends on the value of design variables in a lower design step. Cannot be specified in a top-level design step.
- **designStep:** Within each design step it is possible to specify a nested design step.

The option to specify SelectionVars enables a study to be configured without a priori knowledge of the complete system architecture. In other words, it allows for unknowns that only become known while running the actual design study. Taking the example from the beginning of this section and looking at Figure 15, 'skin_zones_material_allocation' is specified as a 'DesignVarSelectionVar' in the second design step. This means that the amount of skin_material_zones depend on the value of he design variables in the previous step, in this case 'nr_ribs'. In the next section, the process to go from DSC file to actual workflow formulation is explained.

3.2. Dynamic workflow (re-)formulation using KADMOS and CMDOWS

Once a design study is configured, the next step is to formulate the actual MDAO workflows. At this moment the implementation is limited to a use-case in which the analysis of the system of interest is managed by a single KBE application that can be interfaced with simple 'set' and 'get' calls for setting and retrieving information from the model. For this use-case a standard workflow can be constructed, whose basic lay-out does not change depending on the design study configuration. In Figure 16, an example of a main workflow and nested workflow for a 2 step design study are shown (this is a VISTOMS visualization of two CMDOWS files).

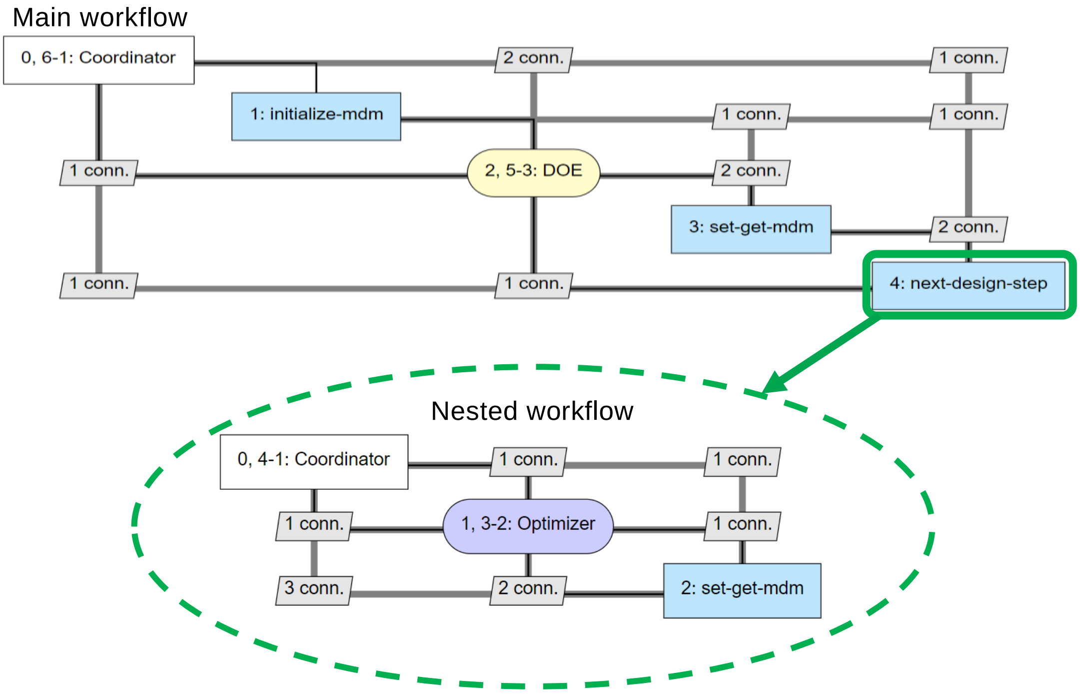


Figure 16: VISTOMS visualization of generated CMDOWS file for a nested multi-architecture optimization workflow calling GKN Fokker's 'MDM' KBE tool.

It can be seen that a number of design competences are present. This workflow is configured to call GKN Fokker Aerostructures 'mdm' KBE tool. However, this setup is general and can in principle be used for any correctly configured KBE tool. The design competences are:

- **initialize_mdm:** this design competence initializes an MDM instance on a local server based on the baseline JSON file. This competence is only present in the top-level workflow as the same MDM instance can be used for the complete design study.
- **set_get_mdm:** this design competence sets and gets whatever information is specified to/from the previously initialized MDM instance. It can be seen that the same tool is present in both the main- and sub-workflow.
- **next_design_step:** this design competence is where the dynamic reformulation of workflows takes place. A sub-workflow is generated based on the information that has become available while running the main workflow. This competence is only present if a lower design step is present, in this example only the top level workflow has this competence. In principle, a sub-workflow can also have a 'next_design_step' competence which calls a sub-sub-workflow (and so on).

In Figure 17, the mapping of the DSC file contents onto the standard workflows for a 2-step design study are shown. The specified 'problemFormulation' in each design step configures the type of MDAO workflow, in this case the main-workflow is a DOE and the sub-workflow is an optimization. In order to evaluate the 'designVarSelectionVars' specified in the 2nd design step, they also become an output of the 'set_get_mdm' of the main-workflow. The amount of variables can be retrieved from the model after setting the nr_ribs. Based on this information, the ‘next_design_step’ competence inject them as design variables into the sub-workflow. Before executing the workflow, the selection variables are put unto the sub workflow as placeholders until the actual values become known and are replaced by the ‘next_design_step’ competence.

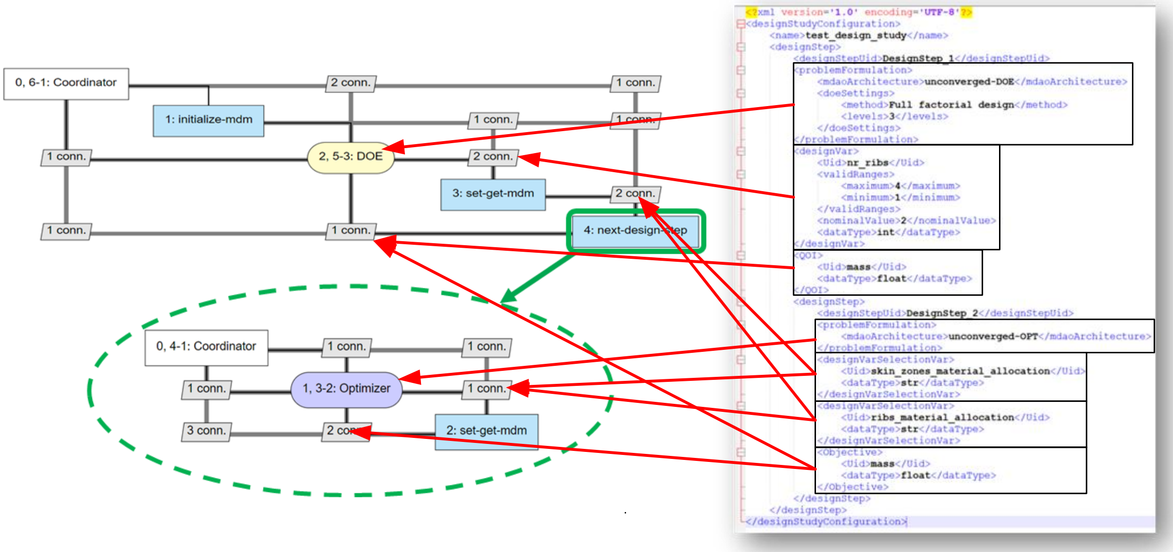


Figure 17: mapping of DSC file onto standard workflows for a 2 step design study

The presented approach has been tested using RCE and GKN-Fokker’s multidisciplinary modeller MDM on a hierarchical architecture optimization problem in D4.3.2 [2].

4. Planned Development

As summarized in Table 1, this report describes the second release of the workflow (re-)formulation tools. Next to KADMOS, these include the CMDOWS data standard for MDAO systems exchange, which might need extensions to support the new KADMOS developments. In this section, the planned developments towards the final release of this deliverable are presented.

4.1. Further integration of surrogate and sensitivity advisory capabilities

As presented in section 2.3, effort have been made to integrate both the surrogate modelling and sensitivity analysis capabilities into one module, minimizing duplication of work and improving usability. At this moment, the advisory modules work as standalone programs calling upon KADMOS to do the workflow reformulation steps, further steps can be taken towards integrating the advisory modules into KADMOS itself.

4.2. Multi-architecture optimization strategies improvement

Adapt CMDOWS and KADMOS to natively support the formulation of sub-workflows and hierarchical variables within MDAO workflows. At this moment it is not possible to do this and a workaround using the above described DSC file using a standard format workflow was needed. A more elegant solution would be to incorporate all of this within the CMDOWS standard, including the additional required KADMOS functionalities. This would also mean the approach could become more general, not only for single KBE applications.

The previous goes hand in hand with a CMDOWS importer to a PIDO tool that can handle the changes made to the CMDOWS standard and materialize workflows with sub-workflows and hierarchical variables. Any extensions to the CMDOWS standard will be reported in D4.2.1 [14]. A solution to this is the development of a CMDOWS-Optimus importer discussed in the next section.

4.3. CMDOWS-Optimus importer

From the developments discussed in the previous sections, it has become clear that the limitations of using RCE, and the envisioned changes to the CMDOWS standard a new importer for materializing CMDOWS files is desired. Within the AGILE project effort has been put into making a CMDOWS importer for NOESIS Solutions' PIDO tool: Optimus [13]. Due to the changes in the latest version of Optimus, however, the importer requires major adjustments.

4.4. Dynamic reformulation of MDAO workflows based on advisory capabilities

The final goal is to enable dynamic re-formulation of workflows to react on results from design space exploration and data analysis from WP5. While most of the advisory capabilities addressed in chapter 2 are based on characteristics of the MDAO problem that can be measured before running the system, or through limited exploratory runs, dynamic reformulations aims at changing the workflow formulation while running the simulation. This might imply keeping the aforementioned advisory capability active during the simulation, in combination with strategies to monitor the data accumulated at run time. E.g. sufficient experiments may become available to trigger the generation of a surrogate model and the substitution of a certain analysis tool in the workflow being executed.

5. Conclusions

WP4 aims to achieve automated (re-)formulation of workflows. This deliverable describes the second release of workflow (re-)formulation tool KADMOS. In this deliverable, developments on advisory capabilities and aiding architecture optimization have been presented. These developments can be used by the industrial use-cases (WP2) described in D2.1.1. [27] for the setup and generation of executable simulation workflows.

The described tools in this deliverable will be developed further resulting in a final (D4.1.3. [9]) release.