D4.1.1. Release of Workflow (re-) formulation tool(s)

Author, company:

- Bastiaan Beijer, KE-Works
- Jente Sonneveld, TU Delft
- Anne-Liza Bruggeman, TU Delft
- Gianfranco La Rocca, TU Delft

Version:
1.4

Date:
April 10, 2022

Status:
Final / Released

Confidentiality:
Public

© Copyright DEFAINE Consortium

Change log

Revision	Date	Prepared by	Checked by	Description
1.0	23/12/2021	Bastiaan Beijer	Jente Sonneveld	First setup of this deliverable
1.1	25/01/2022	Jente Sonneveld	Bastiaan Beijer	Adding a description of KADMOS and its current + planned capabilities
1.2	08/02/2022	Bastiaan Beijer	Max Baan	Included feedback and finalized draft deliverable
1.3	18/03/2022	Jente Sonneveld	Gianfranco La Rocca	Added additional information on planned development
1.4	10/04/2022	Gianfranco La Rocca	Max Baan	Extended KADMOS state of the art description

Table of contents

Change log **2**

Acronyms **4**

Acknowledgements **5**

1. Introduction **7**

1.1. Intended use and purpose of this deliverable 8

2. Workflow (re-)formulation tool KADMOS **9**

2.1. KADMOS and CMDOWS 9

2.2. KADMOS Workflow Reformulation Capabilities 11

2.2.1. Workflow Sequencing 11

2.2.2. Workflow Partitioning 11

2.3. Planned Development 12

2.3.1. Enable gap analysis (D4.1.1) 12

2.3.2. Extend KADMOS advisory capabilities (D4.1.2) 13

2.3.3. Enable Dynamic reformulation of MDAO workflows (D4.1.3) 16

3. Conclusions **17**

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] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] M. A. B. a. J. T. H. a. N. B. a. R. L. a. J. M. a. J. R. R. A. Martins., “A Python surrogate modeling framework with derivatives,” *Advances in Engineering Software,* 2019.

[16] DEFAINE Consortium, “D2.1.1. Use-case specifications,” ITEA3 Call 6, 2021.

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

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

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 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 MDA workflows, then performing 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 and AGILE 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 taking into account 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 state-of-the-art baseline on which the above mentioned developments will be added. 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.

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 the tools and underlying methodology made available during the first release of the technology demonstrator and provide an overview of planned developments. Chapter 2 describes the tool KADMOS developed by the TU Delft, its current workflow reformulation capabilities and planned developments. Finally, in Chapter 3 the overall conclusions are presented.

2. Workflow (re-)formulation tool KADMOS

In the DEFAINE project the tool KADMOS is further developed to enable dynamic workflow (re-) formulations. In this chapter the current capabilities of KADMOS to generate MDAO workflow formulations and support their integration into executable workflow are described first. Then the planned extensions are presented.

2.1. KADMOS and CMDOWS

KADMOS stands for Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System [8] and is a tool[1] for MDAO workflow formulation developed in AGILE [4] and further developed and exploited in the running AGILE4.0 [5] project. As shown in Figure 1 and elaborated below, KADMOS makes use of graph manipulation techniques to model MDAO systems.

KADMOS starts by reading a repository containing the different disciplinary tools (addressed in DEFAINE as engineering services) available to the integrator. Accordingly, the so-called repository connectivity graph (RCG) is generated, showing all the connections between the tool's I/O variables. This graph can be seen as a representation of the tool repository. In the second step, the integrator starts formulating the MDAO problem: problem roles are assigned to the I/O variables (e.g. design variables, constraints, quantity of interests, objectives) and the required tools are selected accordingly from the repository. This results in the Fundamental Problem Graph (FPG), which is a manipulation of the previously generated RCG, where the unnecessary tools and connections, i.e. those not required to model and solve the specified MDAO problem, are removed. Problems like collisions (multiple tools producing the same output) or circular couplings (same quantity of interest required as I and O for the same tool), are automatically flagged and/or solved by KADMOS. Once the MDAO problem is formulated, an MDAO architecture can be applied, i.e. a specific strategy can be selected to solve the stated MDAO problem. KADMOS provides multiple predefined architectures, ranging from basic MDA workflow convergence strategies, parameter explorations (i.e. DOE), up to actual MDAO architectures, both monolithic (e.g. MDF and IDF) and multi-level (e.g. CO and BLISS-2000). The formulation of an own custom architecture is currently not possible, although the order of execution (sequencing) of the tools can be manually changed and partitions (clusters of (de-)coupled tools treated as one macro tool) can be specified. The application of a user-selected MDAO architecture results in two additional graphs: the MDAO Data Graph (MDG) and the MDAO Process Graph (MPG), which together fully define the components and process information of the given MDAO system. At this point the formulation phase of the MDAO solution strategy is completed and the resulting MDAO system can be stored, visualised and converted into an executable workflow. To this purpose, the Common MDO Workflow Schema (CMDOWS) data format is used [9], which is an open source XML-based format, specifically defined to store and exchange MDAO workflow formulations between different applications.

In order to visualize the workflow and to inspect whether all the connections between the different tools and the connections with the convergers and optimizers are generated as expected, the CMDOWS files generated by KADMOS can be imported into VISTOMS (VISualization TOol for MDO Systems) [10]. VISTOMS is a web-based tool capable of producing interactive visualisations

[1] KADMOS is an open-source Python library which can be downloaded at https://bitbucket.org/imcovangent/kadmos

of all the above mentioned graphs, including the very convenient eXtended Design Structure Matrix (XDSM). Finally, the CMDOWS file can be read by a PIDO platform to convert the KADMOS produced formulation into an executable workflow. At this moment CMDOWS translators have been developed for Optimus [11], RCE, and OpenMDAO. More information on the CMDOWS data format will be provided in D4.2.1 [12].

The current KADMOS implementation targets single objective optimization problems, although multi-objective optimization problems could be addressed too, by defining, for example, one pseudo objective function, expresses as the weighted sum of multiple costs functions, and by embedding such single optimization system within a DOE schema, where the weights of the cost functions are iteratively varied, to generate a Pareto front. No automatic formulation functionality is currently available to this purpose, thus requiring manipulations from expert KADMOS users.

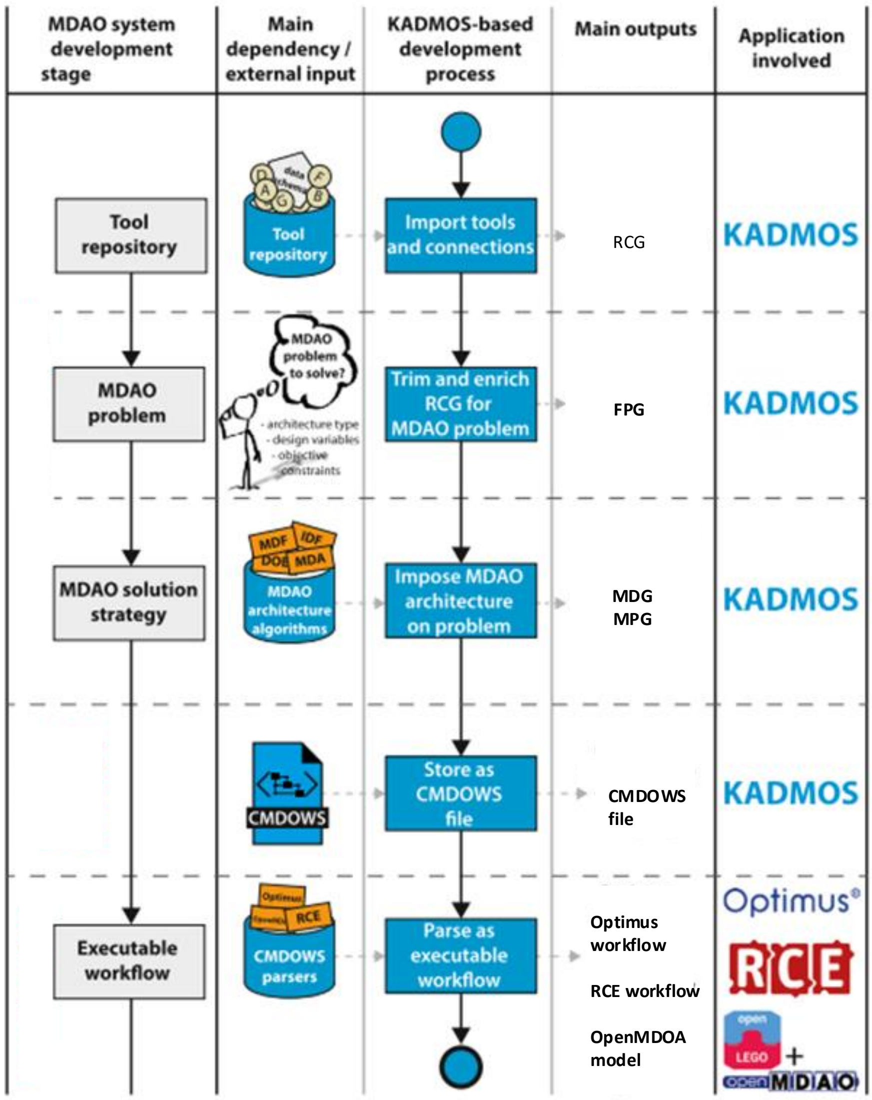


Figure 1: Overview of the formulation phase supported by KADMOS and use of CMDOWS file [8].

2.2. KADMOS Workflow Reformulation Capabilities

In a shared effort with the concurrent AGILE4.0 project [5], KADMOS is being extended with some advisory capabilities for (re)formulating an MDAO workflow. Advice can be given on specific parts of the workflow that can potentially be improved to make the workflow faster and more efficient. At this point this is limited to the workflow *sequencing* and *partitioning* capability introduced above. Both capabilities are briefly explained in the following subsections.

2.2.1. Workflow Sequencing

The order in which the disciplinary tools in an MDAO workflow are executed can have an effect on the amount of iterations required or the time per iteration. KADMOS implements an algorithm that sequences the tools such to minimize the amount of feedback loops in a workflow, thus eliminating or reducing to a minimum the number of coupling variables to be converged in a given MDA workflow. Alternatively, the user can manually define the tool execution order.

2.2.2. Workflow Partitioning

Within KADMOS, a module exists to advice the user on the partitioning of the MDAO problem. Partitioning means that the different design competences within an MDAO problem are divided in several groups. By tearing the coupling between these groups, they can be executed in parallel to reduce the execution time of one iteration. As the data between the different groups need to be consistent, copy variables are introduced that need to be converged. However, when the number of partitions increases, the number of variables that need to be converged increase, thereby leading to an increase in the number of iterations. Therefore, the partitioning problem is a trade-off between the execution time of one iteration and the number of iterations required to reach convergence.

The partitioning advisory capability integrated in KADMOS consists of two steps. In the first step, an advice is given to the user on the number of partitions. A partitioning algorithm determines for each number of partitions the best partitioning that minimizes the execution time of one system iteration and that minimizes the number of variables that need to be converged. In the second step, the user decides on the desired number of partitions, which is then applied to the MDAO problem.

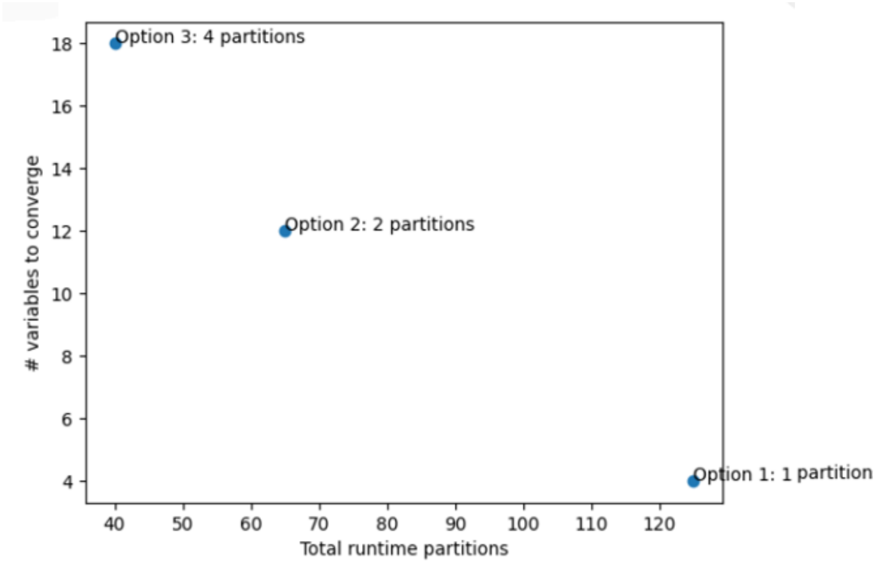


Figure 2: Pareto front showing three different partitioning options by KADMOS

In Figure 2 an example of a partitioning advise given by KADMOS is shown. For each partitioning option, the execution time of one system iteration and the number of variables that need to be converged is shown. It is up to the user to select the option that most suits a particular case. It must be noted that accurate estimates of the execution time of each tool must be available for a good advice to be possible. Besides, the number of variables to converge does not give directly an indication of the computation time required to achieve convergence. To that purpose sensitivity information is needed, which is not available in the current KADMOS implementation.

2.3. Planned Development

As summarized in Table 1, three main releases of the workflow (re-)formulations tools are planned. Next to KADMOS, these include the CMDOWS data standard for MDAO systems exchange, which might need extensions to support the new KADMOS developments. These extensions will be reported in D4.2.1 [12]

Table 1: Future releases of workflow (re-)formulation tools

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

2.3.1. Enable gap analysis (D4.1.1)

The first goal, as stated in the full project proposal [6], is to provide a means to perform gap analysis based on a given set of design competences for a specific design study. Looking at the current state of KADMOS, it can be concluded that gap identification is already possible with existing functionalities. Based on a given repository of design competences and the user specification of the problem roles (e.g. specification of design variables, and various quantities of interest, such as objectives and constraints), KADMOS creates the fundamental problem graph (FPG). To this purpose, KADMOS selects the tools from the repository that are required to compute the assigned quantities of interest. Based on the input required to run those tools, KADMOS will include in the FPG the tools producing such input as output, and so on. When input required to run any of the selected tools are not produced by any tool in the repository, either they are design variables or user defined parameters, or the consequence of a gap in the design competences. Thus a new competence must be added to the repository, such to produce those missing input values.

The FPG graph can be inspected using VISTOMS to identify these gaps. This means that no development for this functionality is envisioned and can be considered available in the state of the art release of the workflow (re-)formulation tools.

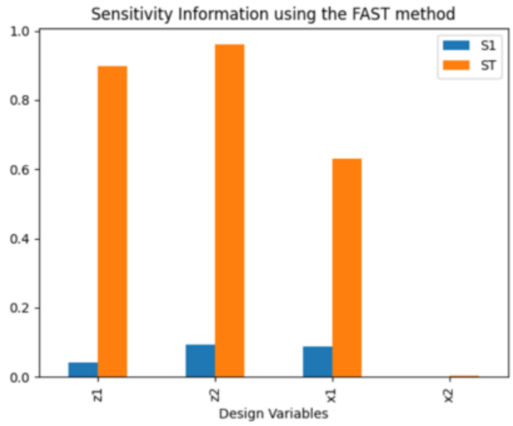
2.3.2. Extend KADMOS advisory capabilities (D4.1.2)

In this section, the additional advisory capabilities envisioned for KADMOS are presented. These kinds of advices will support the user in improving the MDAO workflow formulation and make it faster and more efficient. In addition, they will support the final goal of enabling dynamic reconfigurable workflows.

Sensitivity Analysis

In the industrial needs and framework requirements described in D2.1.1 [7] a need for performing sensitivity analysis is stated. TU Delft has produced a literature review [13] on the topic and is currently developing an advisory system to make use of sensitivity information to adjust MDAO problem formulations in order to reduce computational effort. A new KADMOS module is being developed to formulate a DOE based on an appropriate sensitivity method and asses the most influential design variables in the MDAO problem. An example output of this advisory system prototype is given in **Error! Reference source not found.** (left), where the FAST sensitivity method is used to measure and rank the sensitivity of the 4 design variables of a given MDO benchmark problem. Using a certain threshold value, the lowest sensitivity parameters are excluded from the MDAO problem (x2 in this example) without it significantly effecting the optimization outcome, as shown in **Error! Reference source not found.** (right). This can reduce the amount of time needed to solve the optimization problem, not only because of the lower amount of involved design variables, but, possibly, also because of the exclusion of certain disciplinary tools (e.g. the tool required to produce x2 may be eliminated at all).

In addition, sensitivity information can be used to improve the sequencing and partitioning algorithms discussed in Sections 2.2.1 and 2.2.2 respectively, thereby leading to a more informative output than shown in Figure 2, based on better estimated of the computation time to converge coupling variables.



	Original problem formulation	Problem formulation with SA
# of design variables	4	3
Objective	4.126	4.139
Time [s]	7.52	6.77

Figure 3: Example of sensitivity analysis results (left) and effect of excluding a low-sensitivity variable on the optimization result of a benchmark problem (right).

Surrogate Advisory System

It will be advantageous to have a strategy to evaluate whether surrogate model (SM) generation for replacing expensive analysis services, is possible and pays off for given execution time and accuracy constraints. TU Delft has completed a literature study on the integration of surrogate models in automated MDAO workflow formulations [14] and a Surrogate Advisory System (SAS) is currently in development.

Figure 4 shows the steps such an advisory system can take to arrive at a (partially) surrogate based workflow. First bottlenecks must be identified, this can be done by using sensitivity information (prev. section) or setting up and running a DOE using KADMOS. This provides information on the runtime of each discipline in an MDAO problem, an example is given in Figure 5. Based on runtimes and additional information like dimensionality of the tool input, linearity of the tool output and amount of function calls, a strategy can be formed on which tools or groups of tools can best be surrogated. By plotting the expected loss of accuracy against the gained time performance for a number of different strategies a Pareto front can be visualised. An example of such an advice is shown in Figure 6, here an advised minimum number of samples combined with the Pareto front can help select a suitable strategy.

Based on the selected strategy, the SM's themselves must be constructed. To this purpose, the open source surrogate model toolbox SMT[2] is used [15]. Once again, KADMOS can be used to create a DOE based on the to be surrogated tools and the amount of samples required. The surrogate model is then added to the original tool repository and can be used to generate a new surrogate based optimization workflow. The last step is about using the intermediate results of the optimization iterations as infill points to improve the surrogate accuracy.

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

Figure 4: High level flow diagram showing the position of a Surrogate Advisory System in relation to KADMOS, CMDOWS and PIDO structure.

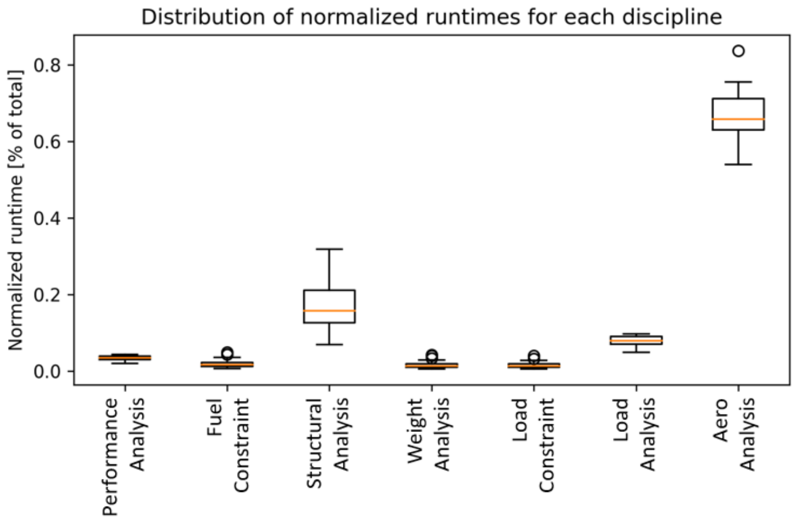


Figure 5: Example of normalized runtimes of disciplinary tools in an MDAO problem

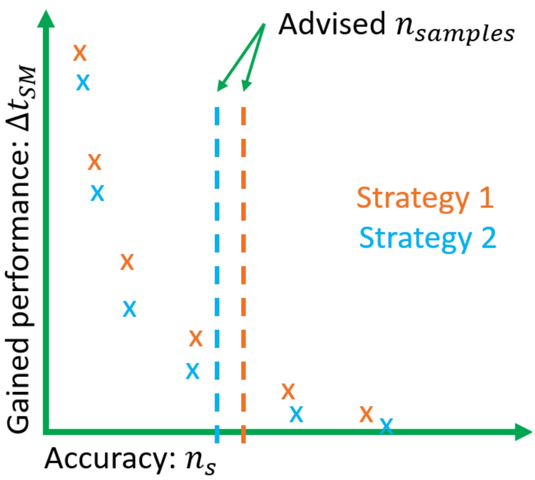


Figure 6: Example of advice given on surrogate modelling strategy by the Surrogate Advisory System

Similar Design Competences

An advisory system for selecting the most fitting set of engineering competences (among similar alternatives) while taking into account the required level of fidelity, license availability, budget and computation time targets was planned. In the industrial needs and framework requirements described in D2.1.1 [7], no urgent need for this functionality is expressed by the industrial partners, also because it is not clear yet, whether multiple engineering competences will be available in the repository to perform the same task (e.g. computing a certain quantity of interest using different level of fidelity). Developing this functionality will therefore be given a lower priority compared to the functionalities discussed above.

2.3.3. Enable Dynamic reformulation of MDAO workflows (D4.1.3)

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 above 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. Dynamic re-formulation capabilities will be enabled in the final release of this deliverable (D4.1.3. [7]).

3. Conclusions

WP4 aims to achieve automated (re-)formulation of workflows. This deliverable describes the first release of workflow (re-)formulation tool KADMOS which are used in the technology demonstrator described further in D4.3.1 [1]. KADMOS can be used by the industrial use-cases (WP2) described in D2.1.1. [16] for the setup and generation of executable simulation workflows.

The described tools in this deliverable will be developed further resulting in a second (D4.1.2. [17]) and final (D4.1.3. [18]) release.