DOI: ADD DOINUMBER HERE

DEFAINE - Design Exploration Framework based on AI for froNt-loaded Engineering

Authors

Max Baan [1], Bas van Manen [2], Petter Andersson[3], Robin Augustinus[4], Robert Braun[5], Christopher Jouannet[6], Gianfranco La Rocca[7], Bastiaan Beijer[8], Massimo Panarotto[9], Peter Edholm[10]

[1] ParaPy B.V., Delft, The Netherlands, max.baan@parapy.nl
[2] GKN Fokker, Papendrecht, the Netherlands, Bas.vanManen@fokker.com
[3]GKN Aerospace Engine System, Trollhättan, Sweden, petter.andersson@gknaerospace.com
[4] GKN Fokker, Hoogerheide, the Netherlands, Robin.Augustinus@fokker.com
[5]Linköping University, Linköping, Sweden, robert.braun@liu.se
[6]Saab AB, Linköping, Sweden, christopher.jouannet@saabgroup.com
[7]TU Delft, Delft, the Netherlands, G.LaRocca@tudelft.nl
[8]KE-works, Delft, the Netherlands, bastiaan.beijer@ke-chain.com
[9]Chalmers University of Technology, Gothenburg, Sweden, massimo.panarotto@chalmers.se
[10]PE-geometry, Mölndal, Sweden, peter.edholm@pe-geometry.se

Abstract

The DEFAINE project proposes companies to adopt a front-loaded development process. Front-loading significantly reduces the inefficiencies of the current engineering approach by enabling large-scale design exploration early on or even before the start of a project. To this purpose, the DEFAINE framework enables fast generation of distributed, re-configurable multidisciplinary engineering workflows, built on sets of design automation solutions, largely based on Knowledge Based Engineering. The workflows are fed with varying sets of parameters and requirements, to automatically perform multiple design exploration studies. With the help of AI techniques, the produced design solutions are analyzed to derive new knowledge.

1. Introduction

The demand for air transportation has been increasing dramatically. It is forecasted to grow by 50% in the next 15 years [1]. How to satisfy such demand, while dealing with the current (almost) saturated infrastructure and the increasingly stringent environmental constraints is the ultimate challenge. The European goals to reduce the emission levels of noise, CO2 and NOX by 65, 75 and 90% respectively, by 2050 [2] are extremely ambitious and hardly realistic when considering the high technological level already achieved within existing aircraft combined with the often-conservative approach of aircraft manufacturers. Whether and to what extent it is possible to improve the performance of existing aeronautical systems and how to assess the actual value of novel solutions and lower their development risk are key questions for industry, research institutes and academia.

To increase the technological level of existing aircraft even further, a radically different product development process is necessary to enable a larger exploration of the design space, enhance designer's productivity and integrate knowledge and capabilities that are distributed within the company and across the supply chain. In the effort to maintain and extend industrial leadership, modern organizations have already started to significantly alter the nature of their engineering processes, for example moving from the old sequential development process to concurrent engineering. Although nowadays different development phases run concurrently and parts of the process have been digitized, within conceptual development the possibilities to perform full in-depth analysis of design cases within the available time are still limited. The process nowadays features the following bottlenecks:

- **Limited availability and incompleteness of data, information, and limited reuse of existing solutions**
  Although design decisions in the early stages of the process often have the largest impact on product performance, due to the nature of the concurrent design process these are often based on limited data and information. This data and information is incomplete and prone to change in the course of the process. Next to this, information from previous projects is often not readily available for conceptual design teams and its meta-information is not saved adequately. Both the incompleteness and volatility of data as well as limited reuse of existing solutions impede product optimization and often results in costly incorporation of changes late in process.

- **Difficult to assess effects of design decisions**
  In the current process, it is very difficult to completely grasp the effect of design decisions whereas this spans multiple teams and disciplines, each involving many detailed analyses and dedicated software tools. This effect is present within companies between disciplinary teams but is even more profound in case of multiple companies working together in a supply chain. Further limited by time constraints within the process, in combination with constant changes to which the design is subjected to, it is difficult to assess and make optimal design decisions.
- **Inadequate traceability of requirements**
  Due to the often-occurring changes in requirements, the main challenge in product design is to choose a design concept having certain robustness in responding to these changes without having significant impact on the business case parameters (e.g. certification measures, cost & weight factors). However, the proper traceability of requirements is hampered by a lack of adequate tools and methods to support their management and tracking. Other challenges in the handling of needs and requirements within conceptual design are that they are not known yet from the start and that the lack of traceability limits engineers in understanding how changing requirements affect the design. Part of the technical requirements are a result from the conceptual design work. Therefore, changes in requirements lead to cumbersome and time-consuming Validation and Verification (V&V) of results and uncertainty of the compliancy of the designs.

The digital transformation has led to a steep increase in the adoption of automation tools. Software development platforms have become available that allow engineers to effectively develop new applications for design activities where no of-the-shelf tools fit with custom purpose-build solutions. These tools offer engineers significant reductions in effort and lead-time of specific tasks albeit for a limited scope and conditions. Modern-day software products offer engineers powerful capabilities to build hybrid workflows capable of integrating automated design and analysis tools with human-oriented activities. The state-of-the-art software solutions supporting the concurrent development process feature the following bottlenecks:

- **Engineering applications are volatile to changes**
  Custom engineering applications are often effective within a strictly limited and known scope and often fall short in supporting unknown or unforeseen situations and designs. Most software development products lack support to effectively absorb such changes and to handle these situations and as such force engineers to refactor their applications. This refactoring is not only time consuming; it also requires knowledge of software development and a thorough and up-to-date understanding of the existing code.
- **Setup of simulation workflows is time consuming**
  The setup of simulation workflows is time consuming and requires continuous adaptions of the integrated tools as well as of the overall architecture to support the iterative nature of the engineering design process. Non value-adding data reformatting activities, caused by the lack of standard interfaces and data formats, are often observed in practice when changes occur. Given the limited timeframe and increasing pressure engineers are not always able to adapt their automated workflows and are forced to revert to manually generated or adjusted designs. In addition, switching to higher fidelity analysis, often implies the generation of new workflows from scratch.
- **Scalable computing solutions require specialized knowledge**
  Whereas the ‘cloudification’ and ‘servitization’ of tools and workflows is ongoing, the setup of virtual and cloud environments for large-scale computing jobs requires specific expertise of deployment techniques and infrastructures. The setup of large-scale design studies on computing infrastructures and management of the vast amount of generated data is not straightforward and not available as commodity to design organizations.

The result is a product development process initiated on assumptions where significant time is spent on coping with requirement changes by performing unwanted manual (redesign) activities. These processes thereby offer limited to no time to explore design alternatives, limit product optimization and result in high non-recurring cost.

2. State-of-the-Art

Product development process in aerospace industry
The practice for the development of high-tech solutions is nowadays often based on principles of concurrent engineering (illustrated in Figure 1) both within companies and along the supply chain. However, by having the different phases run concurrently and often distributed over multiple parties, inefficiencies are introduced. Assumptions need to be made since certain requirements from preceding phases or other partners are not yet clear the moment a design cycle starts. In many cases, those assumptions prove to be wrong, requiring costly redesign activities.

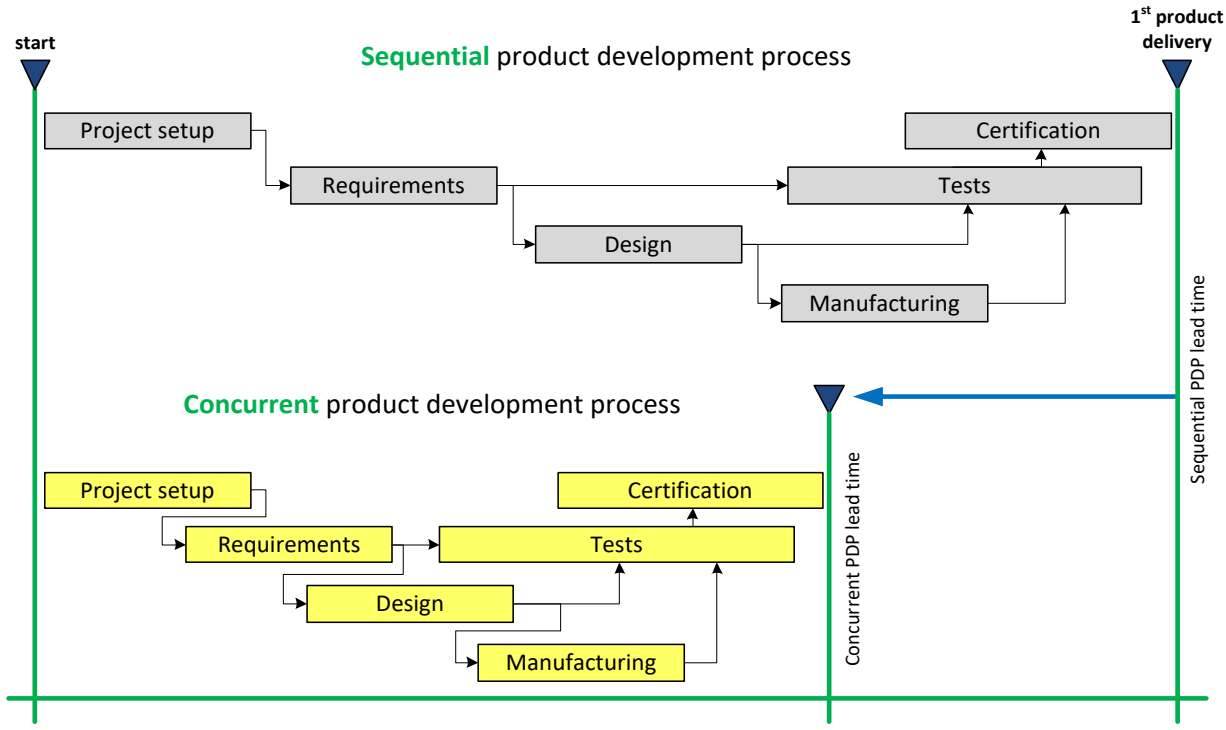

Figure 1: Sequential and concurrent Product Development Process with tests and certification

For the partners across the supply chain, a main focus within their individual product development process is to choose a design concept which can absorb additional requirements change during the design process, without having significant effect on the products' major performance parameters. To achieve this, multiple concepts are traded and their robustness to requirements changes is verified. As requirements tend to change even in the detailed phase of the design process, the design must be flexible with respect to these changes.

In the current state, engineering tools applied within the design process predominantly involve manual operations. The involved tools feature varied, unaligned, and document-based input/output, so data exchange is inefficient and prone to errors. Finally, design process steps are primarily mono-disciplinary driven. This all leads to labour intensive process execution culminating in unnecessary long lead-times and introducing misalignment between the tools involved. This creates obstacles towards enabling design space exploration and multi-disciplinary optimization and trading of design solutions in the early phases of design for finding the most robust and optimal design solution.

Knowledge Based Engineering systems and Model Based Systems Engineering

When the first KBE systems entered the market in the late 80s, they were based on the LISP language, were expensive and required specific hardware. Only large engineering companies could afford them and shorten the lead time of critical programs by means of powerful engineering automation solutions. Specialized knowledge engineers were trained for knowledge acquisition and formalization and LISP programmers were employed to package the formalized knowledge into efficient design automation solutions. The high entry level of KBE, combined with the lack of an adequate development methodology [3] [4], severely limited the industrial exploitation of KBE.

Recently, more affordable KBE systems have largely increased the user base, including also smaller engineering companies. In this setting, the domain expert is often playing the simultaneous role of knowledge engineer and KBE developer. This often leads to less rigorous application development processes, the formal knowledge modelling step if often skipped, with consequences on the traceability of knowledge (black box effect), poor reusability of the application and problematic validation process.

Methodologies like MOKA [5] and KNOMAD [6] are available to structure the KBE development process, with emphasis on knowledge modelling, but, in practice, the transition between knowledge modelling and source code generation is not seamless, with consequences on consistency and synchronization. This is largely due to the traditional document-based systems engineering approach embraced by MOKA.

A more efficient model-based approach as the MBSE practice proposed by INCOSE [7] is not yet present in the KBE domain. MBSE is defined as "the formalized application of modelling [rather than documents] to support system requirements, design, analysis, verification and validation activities throughout development and later life cycle phases". Through the definition of a rigorous and complete System Architecture Model MBSE can ensure the traceability of all model elements to system and user requirements. To this purpose, SysML [8], which is a standard visual modelling language for MBSE, offers specific constructs for modelling, validation and verification of requirements. While MBSE is becoming a mainstream practice in complex engineering domains [9], its use in KBE is still uncommon. However, in combination with automatic code generation techniques, it has the potential to close the gap between knowledge model and KBE app. Model-driven development and visual programming are already trending in low-code development platforms like Mendix, Appian and Google App Maker. However, the quality of the code generated at run time can be inadequate, in terms of structure and clarity [10], to enable a potential KBE developer understanding and modifying it.

MDAO and PIDO workflow systems

Process Integration and Design Optimization (PIDO) tools are mainstream workflow management systems used to orchestrate the execution of heterogeneous and distributed sets of engineering applications (e.g. analysis and simulations), generally under the control of one or more optimization toolboxes. Several commercial (e.g., Optimus, ModelCenter, ModeFRONTIER and HEEDS) and open source platforms (e.g. openMDAO and RCE) exist, which mainly differ for their user interface, the provided set of optimization and design exploration tools, as well as their methods to display the simulation/optimization results. Except for very few, PIDO do not provide templates to define multidisciplinary optimization problem according to classic (or customizable) MDAO architectures [11]. The flexibility (and the burden) to arrange a workflow in the form of such architectures is generally left to the expert user. The integration of a workflow mostly takes place through manual operations involving menu selection and "drag & drop" operations. Some PIDO tools offer scripting functionality, thus enabling other software systems to edit or automate part of the workflow definition.

Great advances in the use and extension of PIDO tools have recently taken place in the MDAO research-oriented projects IDEaliSM and AGILE. Both advocate the separation of the MDAO system formulation from its integration into an executable workflow. The INFORMA system (IDEaliSM) makes use of an ontology-based approach to automatically define complete MDAO formulations. It also provides advisory capabilities to help MDAO novices selecting the most convenient architecture to the problem at hand. Finally, it features a loose interface with Optimus, the PIDO platform by Noesis Solutions. Some concepts were taken further in AGILE, leading to KADMOS, a graph manipulation application able to create and manipulate complete MDAO system formulations, based on a repository of engineering knowledge. The neutral standard format, CMDOWS, was developed to transfer the KADMOS formulations to diverse PIDO platforms, which thanks to a dedicated plug-in, can instantaneously generate executable workflows. Other languages or grammar for MDAO problem formulation are described in literature, such as REMS [12] and Ψ [13], but they do not enable the use of any third party PIDO tool. The OpenMDAO framework allows the user to choose some basic MDAO architecture, but contrary to all other PIDO tools, lacks any friendly user interface and requires programming efforts to operate.

Currently there is no platform that offers dynamic workflow re-configurations, to adjust the MDAO problem formulation, or tool selection, "live" based on the insights gained during simulation.

Data analysis and machine learning

Big data analytics, machine learning and relative storage technologies are the main pillars of modern Artificial Intelligence (AI) techniques. These offer a wide set of algorithms and techniques to capture and analyze large data sets and data streams.

Big Data is a term used to identify data sets that are very large and/or complex (i.e., unstructured or semi-structured) so that traditional data processing applications are inadequate to process them. Analysis on such data may reveal insights from the data (Analytics), but it is also possible to learn from data and extract knowledge through machine / deep learning algorithms, that try to solve a problem without the computer being explicitly programmed for this purpose, but rather learning from the experience the best way to reach the desired solution.

In the last decade, a great number of tools for Data Analytics (DA) have been implemented, most of them extended with Machine / Deep Learning modules. State-of-the-art DA tools comprehend frameworks designed for either or both batch and stream processing. Among the most famous we recall Spark, Flink [14], Google Dataflow [15] and the stream processing only Storm. Among state-of-the-art tools specifically implemented for Machine and Deep Learning (MDL) examples include Caffe [16], Theano [17], Torch [18] and TensorFlow [19]. All of them share a common

Tensor-based data model (multi-dimensional array) and a vision of the application as Dataflow graphs, especially when building neural networks. The Dataflow (DF) based model, more specifically a Directed Acyclic Graph (DAG), is used to represent each application.

Computing infrastructure, containers and container orchestration
Container orchestration frameworks, such as Borg and Kubernetes, allow deploying and managing distributed applications as a set of containers on a shared cluster of (virtual) nodes. However, cost-efficient placement of performance-sensitive applications requires an automated, user-friendly approach for determining optimal container allocation and deployment policies. Application deployers have difficulties with determining the optimal resource allocation policies for the containers of their applications. This is because they typically have less expertise about operating cloud-native applications in production environments. For example, a user study conducted by Microsoft [20] found that 70% of jobs submitted to a production cluster were over-provisioned. And for 20% of the jobs 10x more resources than necessary were allocated.

There exists a wide range of techniques for determining appropriate amount of resources for individual units of work (e.g. application components/VMs/containers) and self-adaptive systems for dynamically adjusting the resources of these units horizontally or vertically. Previous work has proposed several methods based on performance modelling for resource allocation of web-applications. Performance models can either be constructed using queuing theory [21] or via various machine learning techniques such as artificial neural networks (ANN) or state vector machines (SVM) [22]. These approaches rely heavily on the modelling capabilities of the application developers. Others have proposed performance modelling methods and associated middleware for horizontal auto-scaling of VMs or containers based on queuing theory [23], control theory [24] and the universal law of scalability [25]. These methods all require expertise in performance modelling.

In parallel, performance optimization via configuration tuning has been proposed by the means of search-based techniques such as Hill-climbing [26] which successively samples the configuration space in search of a (near) optimal configuration. Similarly, black-box auto-tuning techniques have been shown effective in the selection of VM-instances for data-analytics [27]. The latter methods do not require the construction of an accurate performance model but observes the application as a black box. Thereby, they eliminate the need for expertise in modelling.

3. The DEFAINE framework

DEFAINE delivers a design exploration framework that enables large-scale exploration of designs, data analysis and machine learning capabilities to analyze the designs and flexible modelling tools to infuse the engineering applications and processes with new knowledge. The framework enables design and manufacturing companies in the high-tech industries to adopt a product development process based on front-loaded principles by utilizing the framework during the early stages of or even before the start of a new project. This front-loaded process significantly reduces the inefficiencies of the concurrent engineering approach currently applied in industry. By feeding these engineering workflows with multitudes of possible requirement sets, a large set of solutions is generated. The generation of these solutions is accelerated by providing distributed and scalable computing infrastructures. With the help of techniques from the field of Artificial Intelligence, the resulting design data can be analyzed to discover trends and relations as function of the varied requirement sets. Powered by computing infrastructure, large-scale design studies are executed to perform design space exploration. From the resulting solution set, previously unknown rules and constraints can be inferred. Using flexible tools and new modelling techniques, the inferred knowledge can be captured and directly used to be included in the tools and workflows. The updated analysis capabilities are in turn utilized to perform further front-loading of the design process, providing a learning effect to the framework and eventually leading to a more efficient and effective generation of solutions. The general principle of this novel, “AI-enriched” front-loaded development process is depicted in Figure 2.

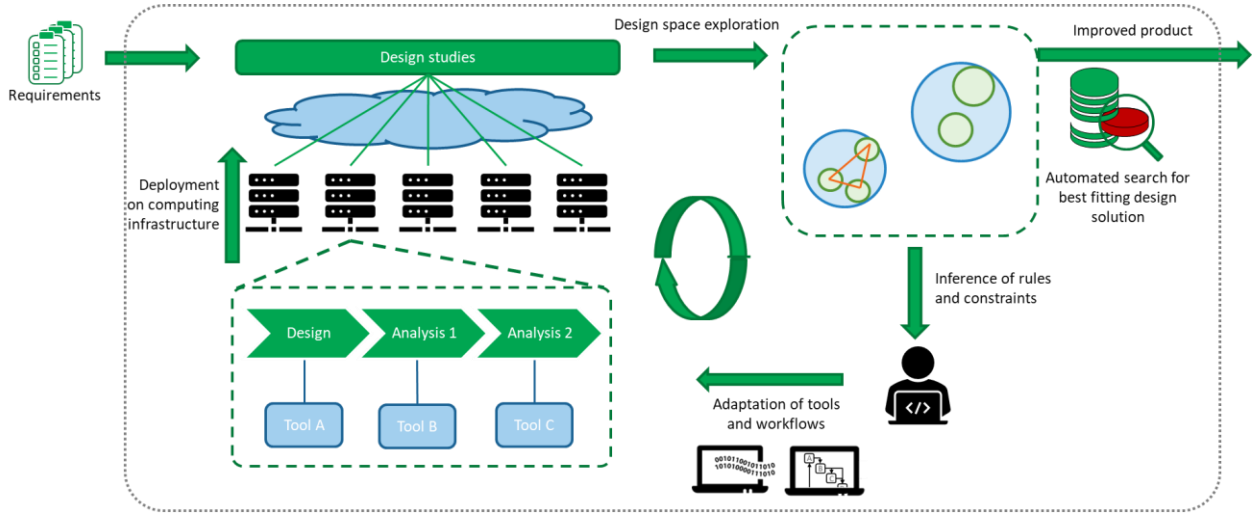

Figure 2: AI-augmented frontloaded design process

This novel front-loaded development process will allow engineers to explore at least a ten-fold of design variants (different requirement sets, different design options) within the timeframe typically required for a single design. Based on the exploration of alternative design variants, it is expected that the recurring cost of the design can be reduced by 10%. Since this also offers design teams increased flexibility in dealing with late changes in requirements, it is expected that this front-loaded approach will reduce the lead-time for design updates by 50%.

To realise this front-loaded development process DEFAINE delivers the following high-level solutions:

- A novel KBE development methodology, based on MBSE concepts and related visual languages (e.g. SysML), which will . help designers develop more transparent applications, faster. The engineering services, based on principles of Knowledge-Based Engineering (KBE), that automate part of the engineering processes are essential to the large-scale generation of design solutions. This methodology will bridge the knowledge capturing and formalization phase - including the app requirements definition - with the actual code development phase by allowing a visual approach to edit KBE applications, thus automatically generating code from graphs and code-less forms. The flexibility of the generated KBE applications will be enhanced by the development of a “live editor”: a new module of the ParaPy KBE system [28] to host live interactive sessions with KBE applications at runtime. Together, the MBSE modelling and live editing will lower the development time for KBE applications by 50% during the development phase and up to 90% for runtime editing. Also, the verification & validation process will benefit from the enhanced level of knowledge traceability provided by the MBSE approach.
- A system for the automatic (re)formulation of multidisciplinary engineering workflows based on provided design requirements and available engineering services (i.e. analysis tools, KBE apps, workflows templates). DEFAINE will extend the functionalities of the KADMOS system developed in AGILE, and the INFORMA method developed in IDEaliSM. The proposed system will enable smart workflow configurations that (i) account for the required level of analysis fidelity and computation time targets; (ii) identifies gaps in the engineering services based on I/O output analysis and (iii) will enable dynamic workflow re-formulations, based on insights (e.g. low sensitivities, inactive constraints) obtained at run time. The generated formulations will be exported, using the CMDOWS standard exchange format . The time required to set up a simulation workflow is expected to be reduced by 75%, with practically zero time requried for re-formulations.
- A scalable infrastructure for computation that supports both deployment on minimal local infrastructure and workstations as well as deployments on cloud platforms. The solution will use state-of-practice containerization technology and extend it with smart deployment and management tools that autonomously generate and distribute large-scale design studies. and can be operated without expert knowledge. The setup time required to prepare large-scale design studies is expected to be reduced by at least 80 to 90% for users without pre-knowledge of such computing infrastructures.
- The exploitation of (automated) data analysis techniques and AI algorithms that can identify trends and relations in large sets of design data and support the inference of knowledge.

The overall architecture of the solution proposed by DEFAINE is depicted in Figure 3.

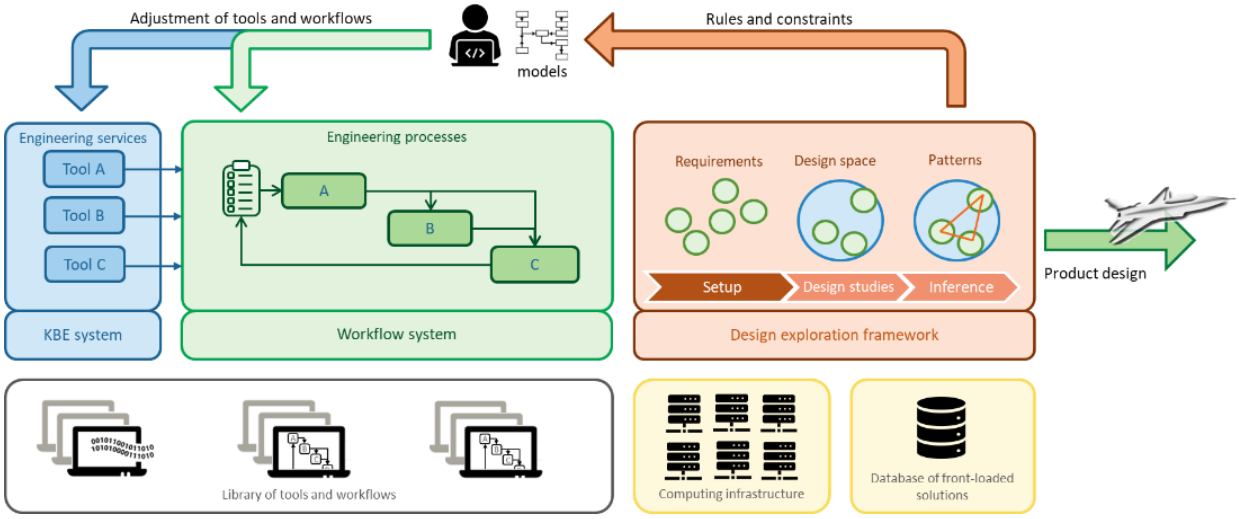

Figure 3: DEFAINE framework architecture

4. Technological innovations

This section presents a subset of the developed innovations within the framework. The selection is based on whether the innovations are discussed in papers submitted for and presentations given during the DEFAINE session at the CEAS conference 2023 in Lausanne.

Development of engineering services
Most of the engineering services developed in the project are based on KBE technology, using different KBE systems (e.g. ParaPy, NX Knowledge Fusion, PACE lab). In general, a KBE system allows engineers to capture engineering logic in a high-level language and provides dedicated libraries to speed up application development, such as automated geometry modelling, mesh generation and integration with a (third party) CAE software. Parametric geometry models can be automatically generated and quickly adapted based on a changing set of input parameters to create varying shapes and models required for the design studies downstream in the process.

To accelerate the development process of KBE applications and guarantee the synchronization between the knowledge models and their codification in source code, TU Delft, in close collaboration with ParaPy, has implemented an MBSE approach (novel in KBE domain) to model requirements, architecture and views of each engineering service based on the ParaPy KBE system. This approach, complemented by automatic code and documentation generation capabilities, accelerates development and reduces the typical black box nature of KBE applications. As additional result of these efforts, the entry level of KBE technology is lowered, enabling less IT specialized developers to tap into its potential.

A live KBE editor is developed by ParaPy to allow users to modify the source code of KBE engineering services (i.e. model configurations, rules, user interface) at run-time, such to alter or include new knowledge, as emerging during the development of the applications and during the execution of the design studies.

Configuration of workflows
The engineering services developed with the KBE system and those services that are already available in the library of tools and workflows are used to configure multidisciplinary analysis workflows to simulate the behavior and performance of the designs. These workflows are formulated using the TU Delft KADMOS system and then stored using the XML-based exchange format CMDOWS developed in the AGILE project. By means of dedicated plug-ins, CMDOWS files containing non-executable formulations of workflows can be automatically translated into executable workflows using existing Process Integration and Design Optimization systems and KE-chain by KE-works. KE-chain allows the modelling of business processes to include the designers in the loop and trigger, under the hood, the KADMOS system functionalities.

For the configuration of workflows, an extended version of the KADMOS system supports users with smart workflows setup by identifying gaps in the capabilities among the available engineering services based on I/O analysis, dynamic reformulation of workflows based on insights obtained during runtime and selection of tools to match required fidelity and computation time constraints.

Design space exploration
Once the workflows are created with the workflow systems, the design exploration framework can execute these workflows to generate many different designs. PE Geometry has extended current generative design systems by connecting function-means modelling tools to geometrical modelling. The automated generation of geometrical models from function-means representations will allow to generate a wider range of design variants compared to the parametric variations that are possible today.

Chalmers University has developed Club Design to manage the traceability among the often-intangible high-level business requirements (e.g., range, fuel consumption, production efficiency, sustainability) coming from external stakeholders (e.g., airlines) down to the measurable technical requirements (e.g. weight, costs, lifetime) which are normally the objectives that engineers at low-level suppliers work with. Engineering design and business development are traditionally conducted by different teams, which adopt different metrics to assess both a product's increased functionality and its business potential. While business development often evaluates new products in terms of the financial value generated over several business scenarios, engineering design teams base their activities on meeting technical requirements. The consequence is that engineering teams and business teams have difficulties sharing accurate and unbiased assessments of what the value of new technologies and solutions is. To solve these communication challenges,

4. Use cases

The solutions provided in this project will be evaluated and demonstrated using five use cases provided by industrial partners. By providing a common problem to solve, the software providers and universities collaborate and develop in a realistic environment where product design data can flow between partners.

On-board systems architecture design
As of today, early conceptual design for on-board system architectures at Saab (see some examples of systems in Figure 4) is primarily addressed with semi-empirical data, technical data of available components or with estimations through steady-state simulations. To improve confidence in early design, to reduce risk and provide lesser uncertainties on expected performance metrics, it is desirable to simplify the use of higher-fidelity systems analysis, like transient simulations to explore time-dependent phenomena, while still guaranteeing fast exploration of different architectures.

The Saab use case focusses on two aspects regarding on-board systems modeling and assessment within aircraft conceptual design:

1. Exploration of different architecture alternatives for on-board systems
2. Automation of time-domain modeling of on-board systems.

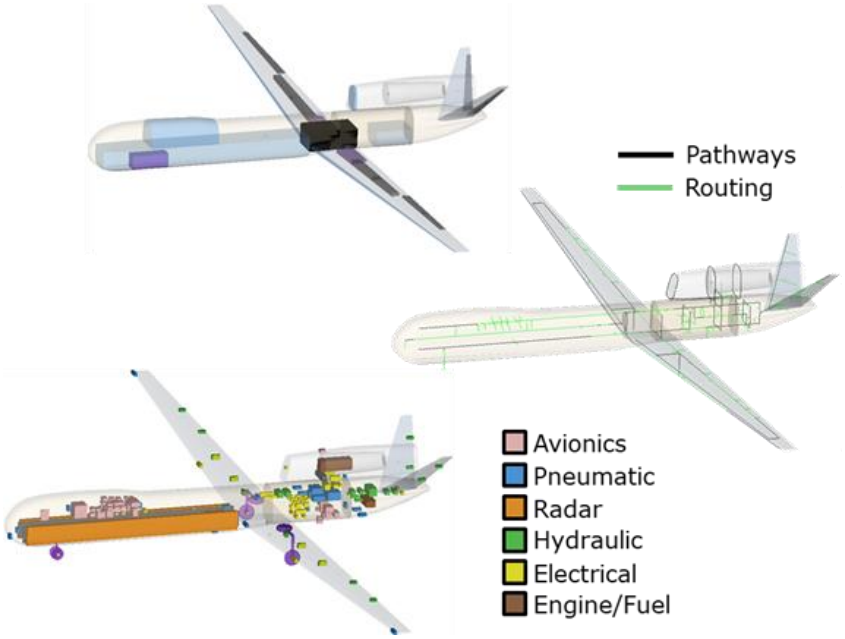

Figure 4: On-board systems relevant for Saab's use case.

Engine component
The GKN Aero Engines use-case focuses on 2 product levels. The first is on engine system level (see Figure 5) and will target electric fan engines. This product is used to connect requirements on system level with requirements on component level, while adding more complexity to the Computer Aided Design (CAD) and analysis models. The second level is engine component level. Different components will be evaluated. For example, the Fan Case of an electric engine or a Turbine Rear Structure of a combustion engine. Most of the technology will be developed for usage on component level.

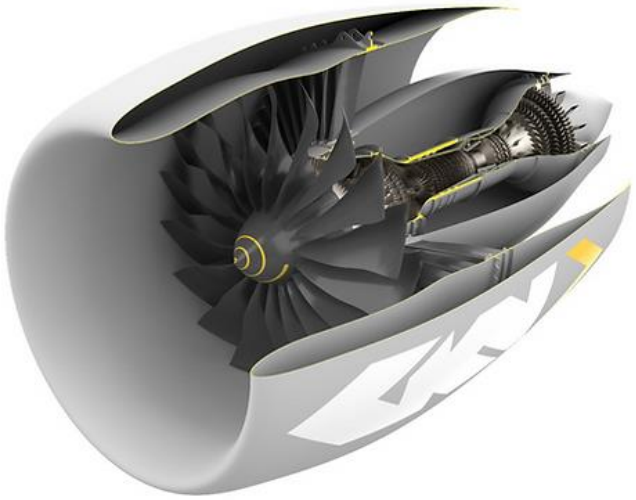

Figure 5: Example of an aero-engine.

Wing box structures
The GKN Fokker Aerostructures use case covers the conceptual design of ailerons, including material trade-offs and sensitivity studies. An aileron (Figure 6) is a wing control surface and its structure typically consists of skin panels, ribs, spars, stringers and hinge brackets. It must comply with a number of requirements including design, stress and manufacturing constraints. Typically, a design is optimized for minimum cost and weight, while making sure that the structure does not fail (stress reserve factors higher than one).. Using traditional methods, time is often limited for detailed analysis and iterations to optimize the design, which. This leads to high project risk and/or conservative design concepts.

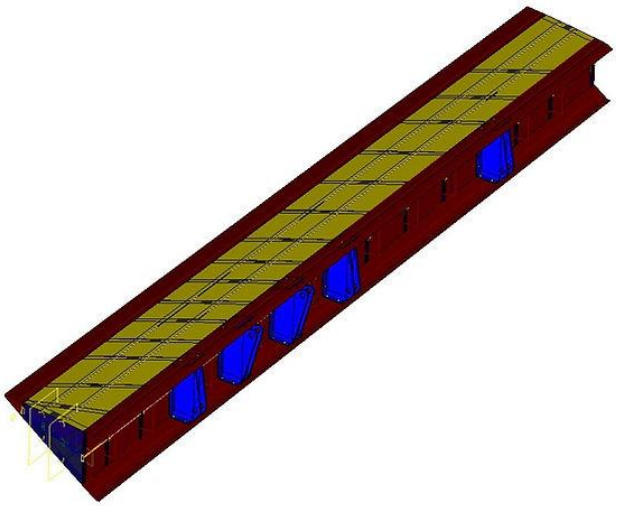

Figure 6: Example of an aileron structure.

Electrical Wiring Interconnection System
The GKN Fokker Elmo use case focuses on the automatic generation of Electrical Wiring Interconnection Systems (EWIS) architecture concepts (see Figure 7 for an example). By automating these processes and using the envisioned DEFAINE framework the aim is to be able to quickly generate different EWIS architecture concepts which can be used to feed more detailed analyses and substantiate both aircraft design and EWIS design trade-offs.

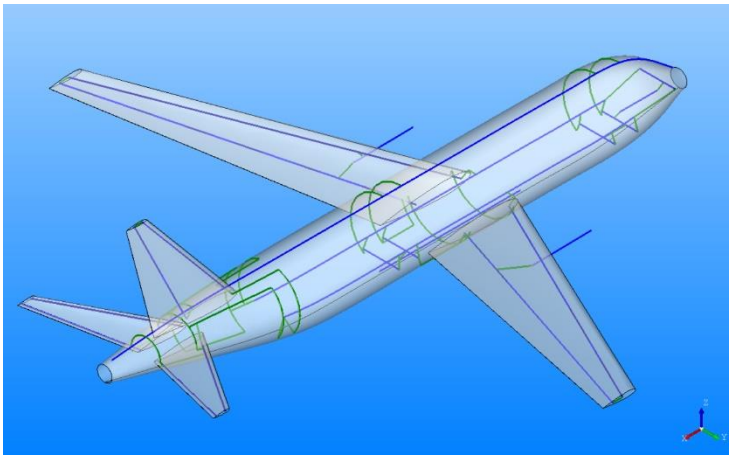

Figure 7: an example of a channel architecture.

Integral aerospace demonstrator
The capabilities of the four aforementioned use cases are combined in an integral demonstrator. This demonstrator will focus on the design of a UAV and several of its systems. Within this use case, Saab AB will be in the role of OEM for the UAV design. Subsequently, Saab AB will perform the design of the on-board systems of the UAV, GKN Aerospace Aero Engines will generate an engine component design, GKN Fokker Aerostructures will design the ailerons and GKN Fokker Elmo will design the EWIS architecture for the UAV. Each of these individual UAV systems will have interrelations with each other.

The systems will all need to be adjusted to each other to generate a design that meets the top-level requirements. The design exploration framework will support the trade-off between the requirements and solutions from the various subsystems by exploring what-if scenarios and alternative designs based on varying sets of requirements. In addition, alignment and optimization between these different systems is to be performed to ultimately obtain an optimal UAV design.

5. Conclusions

Current results within the uses cases have shown that the solutions of the DEFAINE project have significant business impact. The solutions can provide the following advantages to industry

- **Lighter, cheaper designs**
  Conceptual design is a very complex process that involves a lot of uncertainty compared to a detailed design process. Engineers are faced with a design with limited input data that is sensitive to changes. Current business forces engineers to take large margins of safety to compensate for this uncertainty. While not all uncertainty can be taken away, sensitivity studies and design exploration via the DEFAINE framework decreases this uncertainty drastically. Thus, while safety margins remain for requirements that have a large sensitivity with respect to optimization parameters or safety, safety margins can be decreased for requirements that have less sensitivity to these factors. This enables more optimization of designs and thus lighter and cheaper designs.
- **Better understanding of design**
  Design studies are often time consuming. Setting up new types of simulations, developing new methodologies, setting up workflows and executing simulations are all time-consuming tasks. This severely limits the analyses that can be performed. DEFAINE reduces the time and effort required to link simulations and create workflows, in addition to reducing computation time through large-scale deployments on cloud platforms. This enables companies to do more analyses and simulations, getting a more thorough understanding of a system design.
- **Reduced risk**
  In addition to giving more insight into conventional aircraft concepts, a more accessible and thorough analysis of conceptual designs opens the floor to more out-of-the-box and unconventional design concepts such as new low-noise, low CO2 emission platforms. These concepts have a larger risk associated with them, due to these areas not having been explored to the same amount of detail compared to conventional concepts. The easiest way to solve this issue is to try to make up for this lack of exploration. The ability to thoroughly analyze these

new types of concepts will help in making new design concepts more viable for businesses by reducing the risk that they inherently come with.

- **More competitive design proposals**
  Another benefit from the front-loading concept is the ability of suppliers to provide an OEM with more detailed, more accurate and less conservative design proposals, thus increasing their competitiveness. At the same time, also OEM will benefit from the increased confidence in the more mature design proposals received by their suppliers.

The advantages mentioned above lead to significant improvement towards the business goals of the industrial partners within the DEFAINE consortium. Products quality improved in key aspects such as weight, cost or lead-time. Design efforts are reduced, leading to either a reduction in cost or refocusing of engineering effort towards more complex design issues and generation of out-of-the-box concepts. In addition, providing more insight into the effect of conceptual design choices enables the industrial partners to take more responsibility in the design, leading to an extension of the services that the partners can provide and hence an increase in market share and revenue.

Acknowledgement

The research presented in this paper has partially been performed in the framework of the DEFAINE (Design Exploration Framework based on AI for froNt-loaded Engineering) project and has received funding from ITEA 3 programme.

References

[1] “Long-Term Traffic Forecasts,” icao, 2018.

[2] “Flightpath 2050, Europe’s Vision for Aviation,” Report of the High Level Group on Aviation Research, 2011.

[3] G. La Rocca, “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design,” *Journal of Advanced Engineering Informatics,* vol. 26, no. 2, 2012.

[4] Verhagen, W.J.C., Bermell-Garcia, P., van Dijk, R.E.C., Curran, R., “A critical review of Knowledge-Based Engineering: An identification of research challenges,” *Journal of Advanced Engineering Informatics,* vol. 26, no. 1, 2012.

[5] M. consortium, “Managing engineering knowledge : MOKA: methodology for knowledge based engineering applications,” in *ASME Press*, 2001.

[6] Curran, R, Verhagen, W.J.C., van Tooren, M.J.L., van der Laan, T.H, “A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD,” *Expert Systems with Applications,* vol. 37, no. 11, 2010.

[7] I. T. Operations, SYSTEMS ENGINEERING VISION 2020, 2017.

[8] Friedenthal, S., Moore, A., Steiner, R., A practical Guide to SysML, OMG, 2011 .

[9] Crawley, E., Cameron, B., Selva, D., System architecture. Strategy and Product Development, Pearson, 2016.

[10] Klein, J., Levinson, H., Marchetti, J., Model-Driven Engineering: Automatic Code Generation and Beyond, Software Solutions Division, Carnegie Mellon University, 2015.

[11] J. R. R. A. Martins, A. B. Lambe, “Multidisciplinary Design Optimization: A Survey of Architectures,” *AIAA Journal,* vol. 51, no. 7, 2013.

[12] N. Alexandrov, R. Lewis, “Reconfigurability in MDO Problem Synthesis,” in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[13] S. Tosserams, A. T. Hofkamp, L. F. P. Etman and J. E. Rooda, , “A specification language for problem partitioning in decomposition-based design optimization,” *Structural and Multidisciplinary Optimization,* vol. 42, pp. 707-723, 2010.

[14] P. C. e. al, “Lightweight Asynchronous Snapshots for Distributed Dataflows,” *CoRR,* vol. 1506, no. 0860, 2015.

[15] “Google Cloud Dataflow,” Google, 2015. [Online]. Available: https://cloud.google.com/dataflow/.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint, 2014.

[17] T. D. Team, Theano: A Python framework for fast computation of mathematical expressions, vol. 1605, arXiv e-prints, 2016.

[18] R. Collobert, K. Kavukcuoglu, C. Farabet, “Torch7: A Matlab-like Environment for Machine Learning,” in *NIPS Workshop*, 2011.

[19] M. A. e. al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015. [Online]. Available: http://tensorflow.org/.

[20] Sangeetha Abdu Jyothi, Carlo Curino, Ishai Menache, Shravan Matthur Narayanamurthy, Alexey Tumanov, Jonathan Yaniv, Ruslan Mavlyutov, Inigo Goiri, Subru Krishnan, Janardhan Kulkarni, et al. Morpheus, “Towards automated slos for enterprise clusters,” *OSDI*, p. 117–134, 2016.

[21] Yuan Chen, Subu Iyer, Xue Liu, Dejan Milojicic, Akhil Sahai, “Sla decomposition: Translating service level objectives to system level thresholds,” in *ICAC’07*, 2007.

[22] Sajib Kundu, Raju Rangaswami, Ajay Gulati, Ming Zhao, Kaushik Dutta, “Modeling virtualized applications using machine learning techniques,” *ACM*, vol. 47, p. 3–14, 2012.

[23] Cornel Barna, Hamzeh Khazaei, Marios Fokaefs, Marin Litoiu, “Delivering elastic containerized cloud applications to enable devops,” *IEEE Press*, p. 65–75, 2017.

[24] M. Berekmeri, D. Serrano, S. Bouchenak, N. Marchand, B. Robu, “Feedback autonomic provisioning for guaranteeing performance in mapreduce systems,” *IEEE Transactions on Cloud Computing*, p. 1004–1016, 2018.

[25] P. Moura, F. Kon, S. Voulgaris, M. van Steen, “Dynamic resource allocation using performance forecasting,” in *International Conference on High Performance Computing Simulation*, 2016.

[26] Bowei Xi, Zhen Liu, Mukund Raghavachari, Cathy H Xia, Li Zhang, “ A smart hill-climbing algorithm for application server configuration,” in *13th international conference on World Wide Web*, 2004.

[27] Omid Alipourfard, Hongqiang Harry Liu, Jianshu Chen, Shivaram, Venkataraman, Minlan Yu, Ming Zhang, “Adaptively unearthing the best cloud configurations for big data analytics,” *NSDI*, vol. 2, p. 4–2, 2017.

[28] “ParaPy KBE Software Development Kit,” ParaPy B.V. , 2023. [Online]. Available: https://parapy.nl/.