ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

IDEALISM
Integrated & Distributed Engineering Services Framework for MDO

D6.6: Update State-of-the-Art

Author, company:
Rudolph, Stephan – University of Stuttgart
Freund, Jürgen – University of Stuttgart
Tamm, Christopher – Fraunhofer LBF
Panzeri, Marco – Noesis Solutions
d'Ippolito, Roberto – Noesis Solutions
Beijer, Bastiaan – KE-works
Van der Elst, Stefan – KE-works
Kjell Bengtsson – Jotne
Haenisch, Jochen – Jotne
La Rocca, Gianfranco – TU Delft
Eheim, Marc – IILS
Moerland, Erwin – DLR Hamburg
Motzer, Martin – DRÄXLMAIER

Version:
1.0

Date:
January 31, 2018

Status:
Final

Confidentiality:
Public

CHANGE LOG

Vers.	Date	Author	Description
0.1	04.12.2017	Stephan Rudolph Jürgen Freund	Initial setup, TOC
0.2	15.01.2018	Christoph Tamm	Contributions of Fraunhofer LBF
0.3	15.01.2018	Marco Panzeri, Roberto d'Ippolito	Contributions of Noesis Solutions
0.4	15.01.2018	Bastiaan Beijer, Stefan van der Elst	Extended AIF section, Review and extensions throughout document
0.5	15.01.2018	Jürgen Freund	Contributions of the University of Stuttgart
0.6	15.01.2018	Jochen Haenisch Kjell Bengtsson	Contributions of Jotne
0.7	16.01.2018	Gianfranco La Rocca	Contributions of TU Delft
0.8	16.01.2018	Marc Eheim	Contributions of IILS
0.9	16.01.2018	Jürgen Freund	Improvements
0.10	18.01.2018	Erwin Moerland	Contributions of DLR
0.11	23.01.2018	Bastiaan Beijer Stefan van der Elst	Contributions of KE-works
0.12	25.01.2018	Martin Motzer	Contributions of DRÄXLMAIER
0.13	26.01.2018	Kjell Bengtsson	Contributions of Jotne Nr. 2
0.14	29.01.2018	Marco Panzeri, Roberto d'Ippolito	Contributions of Noesis Solutions Nr. 2
0.15	31.01.2018	Gianfranco La Rocca	Final review

Table of Contents

1 Introduction 5
2 State-of-the-Art before the project 6
2.1 Product development process 6
2.2 Integration frameworks 7
2.3 Design languages and standards 8
2.4 Bottlenecks 9
3 State-of-the-Art after the project 10
3.1 Product development process 10
3.1.1 Engineering services to automate repetitive, manual design tasks 11
3.1.2 Integration of a multitude of engineering services in business- and simulation workflows 12
3.1.3 Transition to a front-loaded product development process 12
3.1.4 Service-oriented architecture for product development processes 14
3.2 Integration frameworks 15
3.2.1 Hybrid workflow system 17
3.2.2 Remote service execution module 18
3.2.3 Application Programming Interface 19
3.2.4 Cloudification of MDO simulation workflow 19
3.2.5 Simulation workflow advisory and generation system 19
3.2.6 Live and distributed knowledge base 20
3.3 Design languages and standards 21
3.3.1 ISO standards 22
3.3.1 CPACS 22
3.3.2 VEC 23
3.3.3 Design Languages 23

Abbreviations

AIF .... *Advanced Integration Framework*
BPM .... *Business Process Management*
CAD .... *Computer Aided Design*
CFD .... *Computational Fluid Dynamics*
CPACS .... *Common Language for Aircraft Design*
DMU .... *Digital Mock-Ups*
DSL .... *Domain Specific Language*
ERTP .... *Exploitable Results by Third Parties*
FEM .... *Finite Element Method*
GBDL .... *Graph-based Design Language*
HPC .... *High Performance Computing*
IDEALISM .... *Integrated and Distributed Engineering Services framework for MDO*
KBE .... *Knowledge Based Engineering*
MDO .... *Multidisciplinary Design Optimization*
PDM .... *Product Data Management*
PDP .... *Product Development Process*
PLM .... *Product Lifecycle Management*
REST .... *Representational State Transfer*
SDK .... *Software Development Kit*
SotA .... *State-of-the-Art*
UML .... *Unified Modeling Language*
VEC .... *Vehicle Electric Container*
XML .... *Extensible Markup Language*

1 Introduction

This document provides an overview of the State-of-the-Art of the fields addressed by the IDEaliSM project. The IDEaliSM project is targeting to innovate three main topics, identified as:

- product development process,
- integration frameworks,
- design languages and standards.

At the start of the project a business analysis has been performed to identify the bottlenecks that negatively affect the performance of new products and the necessary development lead-time, thereby hampering the growth and competitiveness of pan-European high tech industry.

As a result of the research and development activities performed in IDEaliSM, several advancements in the state of the art have been achieved. These advancements go beyond the State-of-the-Art of technology by tackling the identified bottlenecks. This document provides a detailed discussion on the technological advancements expressed in the aforementioned main topics. Chapter 2 provides a recap of the State-of-the-Art before the project; Chapter 3 discusses the technical progress per topic of product development process, integration frameworks and design languages and standards.

2 State-of-the-Art before the project

This section gives a State-of-the-Art (SotA) analysis of the three afore mentioned objectives at the start of project in October 2014.

2.1 Product development process

The product development process is **a sequential staged-gate process structured** by major milestones to release product development deliverables. The development process is **human driven, controlled by a scattered organization** that has proven **unable to effectively reuse standard information** for new customer programs.

The status of the deliverables is managed in a Product Data Management (PDM) system, communicating the latest released versions. Besides the use of PDM systems, current product development makes use of Computer Aided Engineering/Design (CAE/CAD) solutions, of which mechanical CAD is predominant. The actual status of the deliverables is known to the actual author of the deliverable, currently working on the new deliverable release. The product development process responsibilities are separated over different departments, responsible for a subset of the deliverables. Each department manages its relevant product data and information in an independent and uncoupled way. To fill the gap of information the current status of project information is managed through regular meetings, communicating information on changes, and requesting information from others. Taking an overview of multiple development programs, each customer and program poses strict requirements and constraints on the process, procedures, methods and supporting tooling. Little to no knowledge is shared and reused within new development programs, resulting in significant losses in efficiency and productivity. This gap of information means that the actual impact of a change is not known directly and organizations are not able to rapidly respond to changes. The same holds for errors. The effects of errors will surface much later, disabling organizations to 'fail fast", provide feedback and to retrieve the actual cause of an error. Organizations are therefore inflexible to absorb changes and cannot benefit from early errors detection to improve the product. Instead, changes and errors results in rework in a late stage of the development process and therefore have a largely negative effect on the cost-efficiency.

Besides changes and errors, companies show a lack of control over their development process and its performance. Many development processes can be significantly improved by streamlining workflows and applying lean engineering practices. Non-value added activities are persistent and there is a lack of control over performance indicators such as queues, batch-size, waiting times, cycle times and process bottlenecks.

The customization of the processes to match the requirements and constraints imposed by customers' results in a **poor use of standards and standardized solutions**. Many processes are not well-defined since organizations feature no dedicated department in charge of the management of standard working procedures and related knowledge such as best-practices and lessons learned. Specifications are not always explicitly documented and communicated, making expertise and personal know-how of key importance. Currently, most of the engineering knowledge is personal know-how and experience of domain experts making it difficult to access for other engineers. For this reason, off-shoring work from critical resources to lower-cost and flexible work force is very difficult.

2.2 Integration frameworks

At the end of 2014, Product Lifecycle Management (PLM) is the most comprehensive family of IT applications supporting the complete life cycle of product development. PLM features an integration of different applications, like Product Data Management (PDM) and Computer Aided Design (CAD) and Engineering applications. PDM applications capture product development information, mainly catering to the data defined in Digital Mock-Ups (DMU). For authoring information, many different IT systems are used, of which CAD is the most prominent. As a result PDM applications focus on CAD-centric product information, and feature limited support for other information types. Many PLM applications are proprietary and closed systems, have proven to be inflexible in their interoperability with alternative third party applications and feature limited integration support for open interface standards. They also offer limited support for knowledge management disabling effective reuse of knowledge in internal processes, workflows and applications via Knowledge Based Engineering (KBE) techniques.

Back then there exist plenty of workflow management systems that serve different purposes, provided with various features and based on several workflow languages. When considering the context of engineering disciplines, two different workflow categories are identified:

1. task or business process workflows and
2. simulation or computational workflows.

The discipline Business Process Management (BPM) takes care of all tasks needed to define, document, communicate, implement, and optimize business process workflows. In the current BPM suites, support for measuring the critical performance indicators in product development processes is non-existent, as is the integration of simulation and calculation activities or a suitable library to manage the mathematical and scientific engineering rules. While BPM systems are used to streamline document based processes on a case-by-case basis they do not provide support for the dynamic nature of engineering, taking aspects into account like feedback loops induced by changes and errors, repetitive tasks, bulk data, authorization steps for the release data and data adapters to enable integration with third party systems. Simulation workflow systems are concerned with the execution of multiple tools one after another, or in parallel, to perform a complete analysis of the performance and characteristics of a product. In present-day supply chain networks, simulation workflows and tasks are lacking explicit context on their position in the supply chain and the influence of their results of the process. Simulation workflows are either not explicitly defined, or they are separately defined and documented outside the overall supply chain workflow itself. Systems that support hybrid workflows consisting of both human and simulation activities as characteristic for engineering are non-existing.

KBE tools are able to embed knowledge and logic in design applications to automate part of the decision process. Parametric CAD, and rule-based, object oriented design are the cornerstones of KBE. State-of-the-Art KBE systems provide also functionalities such as demand driven evaluation and dependency tracking[1]. However, the use of pure KBE applications is limited. Instead, modern engineers favor commercial off the shelf tools and other in-house developed applications. Often each individual tool is strongly associated with one or few experts holding tool-specific knowledge

[1] La Rocca, G.; van Tooren, M. J. L., Knowledge-based engineering to support aircraft multidisciplinary design and optimization. Proceedings of the Institution of Mechanical Engineering, Part G: Journal of Aerospace Engineering 2010, 224 (9), 1041-1055.

such as favored settings, post-processing steps and interpretation of results. The application of KBE tools is limited. Cost-effective automation can only be applied to the most time-consuming and recurring steps in the development. The existing KBE tools, show a limited longevity and are not tested nor certified to match the organizational level of certification. Knowledge and logic is often hard-coded requiring effort to reconfigure the application in order to comply with specifications of new programs and customers. As a result KBE tools show limited application and reuse and are often perceived as expensive.

2.3 Design languages and standards

The languages and standards determine the interoperability of the framework components as well as the generative nature of the individual components. In 2014, industry standards, reference and domain models try to reach interoperability by fixed terminologies and models with fixed semantics. They have proven to be not flexible and expressive enough, unable to be used by the standard engineer.

For product data this project will base its interoperability solution on ISO 10303-209:2014, the Application Protocol for Multidisciplinary analysis and design (AP209), which was published by ISO in 2014[2]. It is capable of representing in a vendor-independent manner a wide range of information elements used in analysis and design. AP209 includes AP242ed1, Managed Model Based 3D Engineering. Due to its coherent data model for product life-cycle data, ISO 10303 is a key to long-lasting knowledge representation and a mediator among the many different language and data representations schemes (design languages, UML, XML, CPACS and further domain specific languages).

One of the key elements for interoperability will be based on ontology techniques. These ontology techniques will play a decisive role in the distributed representation of specific design knowledge in several "partial" ontologies, which capture dedicated design information. While some of these ontology techniques[3] are already developed, some project specific extensions must be developed to adapt for the specific project means. The main roadblock is the lack of a coherent model of the entire engineering cycle that combines all the views of the organization from the perspective of the different stakeholders. Currently, IT does not yet provide the means to represent the needs, views, and models of a single stakeholder in a completely automated and efficient manner.

At business process level Business Process Modeling is a common language for which many standards and common practices exist, but suffers from weak semantics. This weak foundation results in severe restrictions of usage. It is not possible to query the enterprise model asking which business processes are affected by a new regulation. In contrast, each individual query must be implemented separately. It is also not possible to automatically adapt business processes to changing conditions. To overcome this problem, some initial steps were taken (e.g. FP6 Integrated Project SUPER). However, SUPER mainly focuses on executable business processes. For simulation workflows different neutral descriptions of the computational workflows are available, achieved by adopting a commonly standardized XML document representation of the workflow. Various research initiatives, like the EU project Crescendo, have been undertaken to

[2] ISO 10303-209:2014 Industrial automation systems and integration — Product data representation and exchange — Part 209: Application protocol: Multidisciplinary analysis and design

[3] Staab, S. and Studer, R. (ed.) Handbook on Ontologies. International Handbooks on Information Systems. Springer, Berlin, 2003

work out an XML representation that could be widely adopted and become a computer readable standard representation of the workflow[4] [5]. In this way as long as the inputs and the outputs of the process are kept the same the workflow can be executed on different platforms and exchanged across different organizations. A potential solution to this integration problem is available in form of so-called design languages[6].

At technology level there exist numerous engineering domain specific languages, ontologies and data standards. If interfaces between domains or integration frameworks exist at all, they are customized in-house solutions. Besides that, a standard on the modelling and simulation or experimental testing of the structural dynamics of cable harnesses does not exist. In order to consider the mechanical behavior directly during the design and development process, interfaces between design languages have to be specified (e.g. Ansys Parametric Design Language (FEM), IILS mbH routing simulation).

Languages fundamental to knowledge-based applications have not yet converged to a final judgement. The need is to establish a fair balance between the preferred use of domain specific languages (DSL) versus the use of general purpose languages such as graph-based design languages on the basis of the Unified Modelling Language (UML) in the context of engineering design, design automation and design workflow management. The roadblock is a lack of establish mapping techniques between the different language and data representations schemes design languages, UML, XML, CPACS and further DSLs) which allows to profit from the advantages offered by the different points of view.

2.4 Bottlenecks

From the above State-of-the-Art analysis' some bottlenecks were identified which lead to the development of the tools and methodologies of this project.

- **Difficult to assess and account for multidisciplinary effects in design decisions**
  Designers are poorly supported in addressing the multidisciplinary nature of new, complex, inherently integrated products.
- **Limited re-use of data, information and standard solutions**
  Limited formalizations and capture of knowledge prevent reuse of proven solutions. Dominance of ad hoc vs. standard, re-usable solutions.
- **Abundance of repetitive, non-automated design activities**
  Design process is heavily hampered by many repetitive activities, mostly based on manual, labor intensive work.
- **Too many non-value adding "design" activities**
  Waste due to data transfer, (re)formatting, pre-/post-processing among distributed and

[4] Wenzel, H., Gondhalekar, A., Balachandran, L., Guenov, M. and Nunez, M., "Automated generation of Isight-Models through a neutral workflow description", 2011 SIMULIA Customer Conference, Barcelona,17-19 May, 2011.

[5] Gondhalekar, A. C., Guenov, M. D., Wenzel, H., Balachandran, L. K. and Nunez, M., "Neutral Description and Exchange of Design Computational Workflows", 18th International Conference on Engineering Design (ICED11), 15-18 August 2011, Technical University of Denmark.

[6] Rudolph, S.: On design process modelling aspects in complex systems. 13th NASA-ESA Workshop on Product Data Exchange, May 11–12, Cypress, California, USA, 2011.

heterogeneous design and analysis tools. Waste due to rework caused by inaccurate/wrong assumptions.

- **Silos of data, no single source of truth**
  Collaborative product development is severely hampered, data dependency is not guaranteed, leading to errors and inconsistencies.

All of the above-mentioned bottlenecks have a very negative impact on product performance and development lead-time. The generation of different and individual design solutions, as well as the first design solution, which in general takes the longest, is limited because of the productive slowdown given by the bottlenecks during a given project time. Consequently, tradeoffs between feasible and optimal solutions are forced, meaning designers usually have time to generate some feasible solutions but often do not have the time and resources to look for optimal ones. Further many manual activities introduce errors that could be minimized and reduced by automation. Which in turn could make tracking and handling of requirement changes much easier. By reducing or even removing these bottlenecks budget and time overruns of projects could be prevented and therefore minimize recurring costs. Suggested solutions are shortening lead-time by automation and improve product performance by means of multidisciplinary optimization which are presented in detail in the following sections.

3 State-of-the-Art after the project

Following the State-of-the-Art before the project and the bottlenecks identified in the previous section, this section provides a comprehensive insight into the project development outcomes contributing to the State-of-the-Art of the project in 2018. The main innovations of IDEaliSM are:

1. A **framework** to create and execute **hybrid workflows**, natively integrating interactive engineering processes, simulation workflows, tools and data.
2. Adaptation of interfaces and adoption of **standard exchange formats** to enable plug-and-play integration and interoperation of heterogeneous sets of **engineering services.**
3. A set of **ontologies and graph-based design languages** to re-use knowledge and automate often-repetitive engineering tasks.
4. Process **optimization** based on data dependency tracking, management of changes, and a **single source of data.**

These four main innovations are mapped on the three main topics of the IDEALISM project in the following sections. Those readers that are interested in the exploitable results are referred to the *Exploitable Results by Third Parties (ERTP)* containing the most notable exploitable outcomes of the project. These exploitable results are based on or embody the innovations mentioned in this section. The ERTP document can be retrieved from the ITEA 3 project website (itea3.org/project/idealism.html).

3.1 Product development process

This section describes the advancements in the field of product development processes (PDP). The current-state of product development processes, a comprehensive overview of the bottlenecks in this current-state and the technical enablers to tackle these bottlenecks are discussed in detail in D2.3 *Industrial service-oriented process methodology*[7]. This document is publicly available at the ITEA 3 project website (itea3.org/project/idealism.html).

The IDEaliSM framework aims at drastically reducing the time-to-market and development cost of high-tech structures and systems, by delivering a novel product development framework being:

- distributed, flexible and service-oriented
- capable of utilizing multidisciplinary design and optimization techniques
- capable of integrating people, process and technology.

The technologies delivered in IDEaliSM enable a different way of working, opening the way towards radically new and improved PDPs. The future state PDP can at best be implemented sequentially in three major implementation steps:

1. Engineering services to automate repetitive manual design tasks;
2. Integration of a multitude of engineering services in business - and simulation workflows;
3. Transition to a front-loaded product development process.

These three implementations are ordered based on increasing solution coverage with respect to the process bottlenecks. Furthermore, the implementation steps are chosen such that the various phases incrementally improve the design process and such that the implementation of each of the individual steps already provides a significant benefit for the company.

This staged implementation strategy generally helps to gain confidence in the updated process and maximizes the chance to proceed to subsequent phases. Obviously, to gain maximum benefits and profit from all efforts, the complete implementation of all steps is required.

3.1.1 Engineering services to automate repetitive, manual design tasks

The first implementation step entails the development of engineering services to automate parts of the design process that are currently of a repetitive, manual and non-value adding nature. A key feature of this future-state PDP is the use of engineering services.

An engineering service is defined as a generically applicable software routine within the engineering domain, capable of automated handling input and output data in a standardized data format, which can be approached by other services via standard web or network technologies and ideally allows for batch execution without requiring any intervention of the user.

Within IDEaliSM a conceptual framework (explained in more detail in section 3.1.4) and enabling technologies have been developed to enable the structured capturing, formalization and automatic execution of company-specific engineering knowledge by using software technology. This principle is referred to as Knowledge Based Engineering (KBE). KBE enables the re-usage of knowledge accumulated over the years by automatically executing (mono-disciplinary) design tasks.

The main innovation brought forward by adding interfaces to standard data formats for the exchange of process- and product information enables running a KBE application without user intervention as an engineering services. The automated exchange of data through interfaces to standard data formats has multiple benefits:

[7] The first ERTP: IDEaliSM framework architecture application

1. it further standardizes the execution of a KBE routine and allows saving input-output data according to standardized schemes interpretable by multiple engineers.
2. it enables the connection to other engineering services allowing for the standardization of complete parts of the PDP.

The technical enablers to develop such engineering services are further described in section 3.2, where about a key IDEaliSM innovation: the Engineering Language Workbench.

3.1.2 Integration of a multitude of engineering services in business- and simulation workflows

This second implementation step covers the development and integration of multiple engineering services within business- and simulation workflows. This largely widens the scope of automation applied to the design process and characteristically allows for the adoption of MDO techniques.

Within business and simulation workflows, multiple manual- and automated engineering competences are integrated into a single process, allowing for project performance monitoring, embedding requirements and change management, and optimization principles to be included.

Integrating engineering services and performing MDO enables the (semi) automated exploration of design solution spaces. Pushing MDO and numerical post processing techniques into the early design phases will support the decision-making process considerably through computerized generation of reliable physical information within the bounds of the explored design spaces. When applied to the conceptual design phases of the product development process, this allows for more substantiated design decisions at moments when the effect of design decisions and design freedom is still relatively high and the cost of proposing changes relatively low. Through this capability, the reduction of non-recurring costs due to the broader scope of design automation in early design phases represents a well-wanted secondary effect.

The technical enablers to allow for the integration of multiple engineering services in both the business and simulation workflows, is further described in section 3.2. This section describes a key IDEaliSM innovation: the Advanced Integration Framework. In this innovation business processes and simulation workflows are integrated into a single ‘hybrid’ workflow architecture.

3.1.3 Transition to a front-loaded product development process

Front loading is described by Thomke and Fujimoto[8] as “a strategy that seeks to increase development performance by shifting the identification and solving of design problems to earlier phases of a product development process”.

However, this strategy can be applied to shift the problem identification and solving phase even further forward, by developing engineering knowledge before the actual design process starts. Within IDEaliSM, therefore the following definition of front-loading is introduced.

A front-loaded PDP is defined as a strategy in which increased performance and reduced time-to-market is sought by shifting the identification and resolution of design problems to earlier phases, or even in front of the actual product development process.

[8] Thomke, S. and Fujimoto, T., 2000. The Effect of “Front-Loading” Problem-Solving on Product Development Performance. Journal of product innovation management, 17(2), pp.128-142

The principle of a front-loaded PDP, and how it offsets the traditional sequential and concurrent engineering development processes, is depicted in Figure 1. In the front-loaded scenario, product- and engineering knowledge from earlier projects is captured, re-used and standardized to:

- Enable rapid evaluation of man design variants whilst covering different requirements sets
- Provide the ability to rapidly switch to alternative concepts when design requirements from the customer change or become more mature
- Development setbacks can be identified when changes in the product are still allowed, allowing for a better response to changing requirements.

To enable a front-loaded PDP, a knowledge library including the storage and re-use of validated standard solutions a service-oriented process is proposed, discussed in section 3.1.4. This section introduces a key IDEaliSM innovation; the Engineering Library.

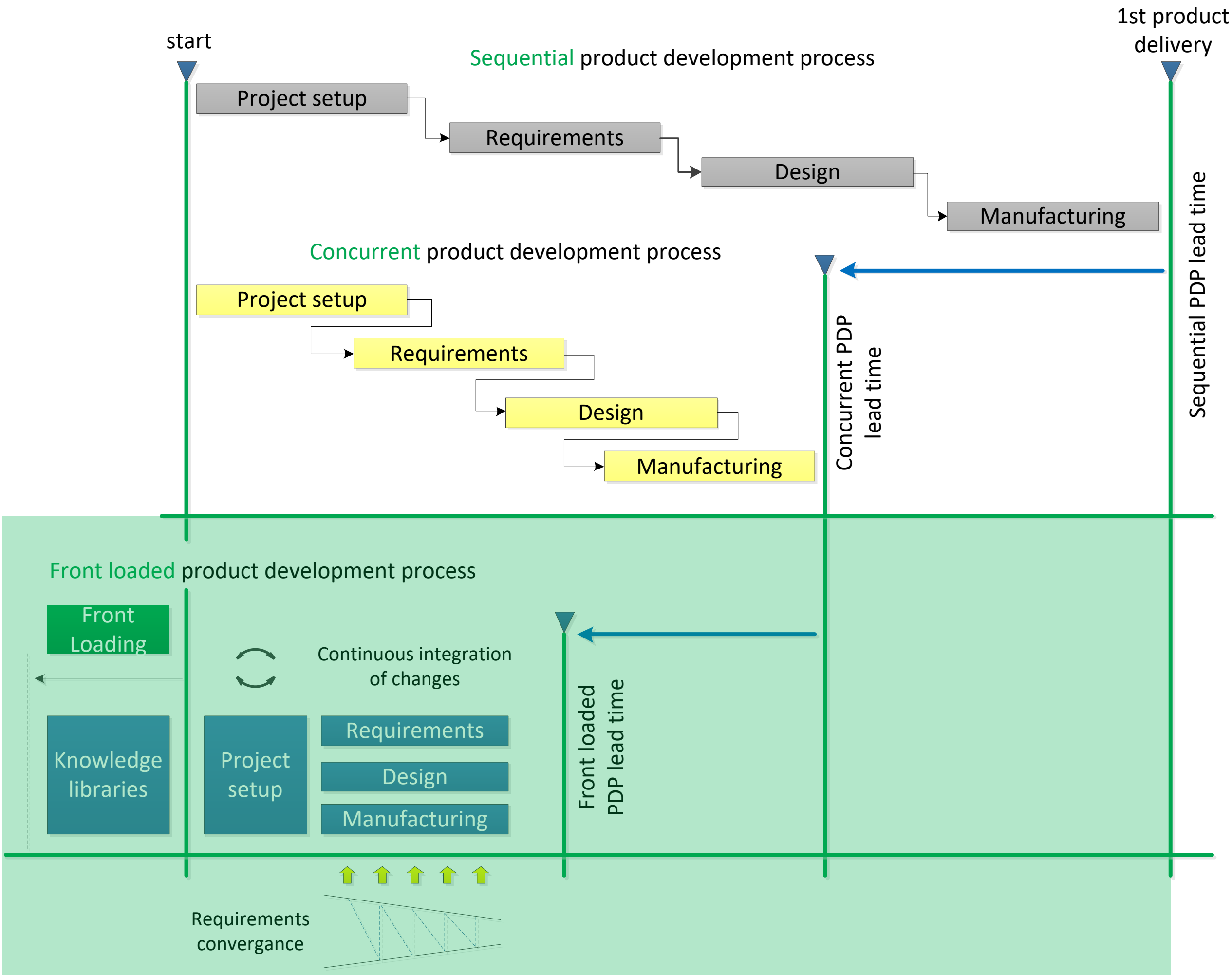

Figure 1: Innovative, front-loaded product development process

3.1.4 Service-oriented architecture for product development processes

The main innovations presented in the previous sections describe the innovations brought by IDEaliSM to solve identified current process bottlenecks. Obviously, a more radical future process also requires more resources for adequate implementation of both a technical- as well as organizational nature.

To support the realization of the future-state PDP enabling the above-mentioned innovations, a service-oriented architecture is developed (called the IDEaliSM framework). The IDEaliSM framework is a novel product development framework for multidisciplinary design optimization. *Figure 2* provides a schematic overview of the main components and their role within the overall IDEaliSM framework architecture. The architecture consists of three major components:

1. The **Engineering Library (EL)** is at the core of the framework. It is a repository in which the knowledge, tools and services of all partners involved in a project are made available;
2. The **Engineering Language Workbench (ELW)** is the environment for creating and adjusting the capabilities within the Engineering Library;
3. The **Advanced Integration Framework (AIF)** allows for logically arranging the available engineering library contents and executing the analyses within the product development process.

These three components are developed to support typical user scenarios encountered in the future of PDPs: support front-loading and multi-disciplinary design and optimization in a reusable and integrated framework. The IDEaliSM framework is distributed, flexible and service-oriented. The distributed nature is required to integrate heterogeneous and distributed sets of people, processes and technologies. The IDEaliSM framework architecture is industry independent. Moreover, it enables[9]:

- **35%-90%** reduction of design activities
- **50%-90%** time reduction to incorporate changes
- Hundreds of design iterations possible versus a few w.r.t. the current state of the art
- Product performance improvements through the application of MDO

The technical details and the innovations brought forward by IDEaliSM to accommodate for an overall integration framework to support future PDPs per component are presented in the following chapter.

[9] IDEaliSM, Deliverable 5.1.3 Integration Framework Validation - Final, 2017.

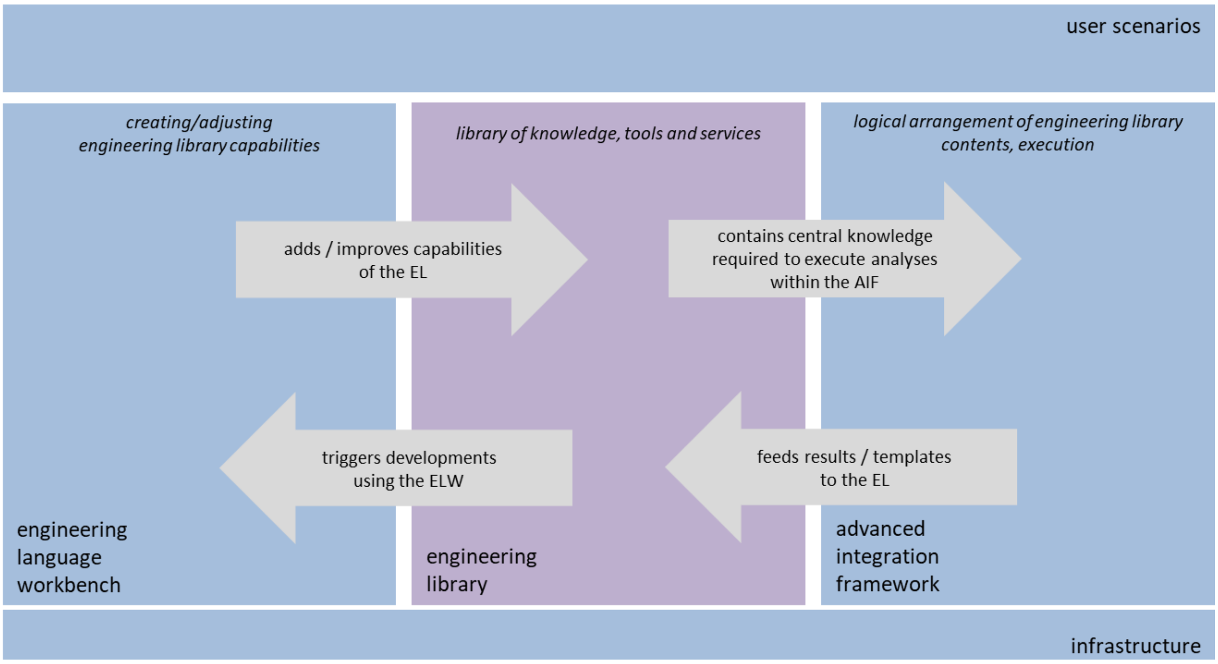

Figure 2: Basic view of a service-oriented architecture to realize future product development processes, showing the main components and their interactions

3.2 Integration frameworks

As introduced in section 3.1 one of the key enablers in IDEaliSM's key innovations is a service-oriented architecture to realize future product development processes: **the IDEaliSM framework architecture**. The main technical components of the IDEaliSM framework architecture are the Engineering Library (EL), the Engineering Language Workbench (ELW) and the Advanced Integration Framework (AIF). These main technical components are shown in Figure 3 in more detail.

IDEaliSM contributes the AIF to the SotA of integration frameworks in general. This AIF facilitates the collaboration among distributed development teams and allows the (re)use of the pre-existing solutions and engineering services available in the EL. It enables design activities requiring human interaction (e.g. by using business workflow templates from the EL) as well as automated computation-intensive simulation workflows (using the engineering services and workflow templates stored in the EL). Furthermore, it is responsible for storing (intermediate) results, to enable sharing a single source of data and ensure data consistency.

Finally, as part of the IDEaliSM framework architecture, the ELW is a dedicated environment for the development of engineering services. In IDEaliSM a set of standardized data exchange formats have been developed based on which engineering services were developed. Using standardized interfaces between engineering services, interconnection of multiple engineering services becomes possible. Finally ontology and graph-based design languages have been developed to enable quick generation of services and the sharing of general automation capabilities. Data standards developed within the context of IDEaliSM that add to the state-of-the-art are explained in more detail in section 3.3.

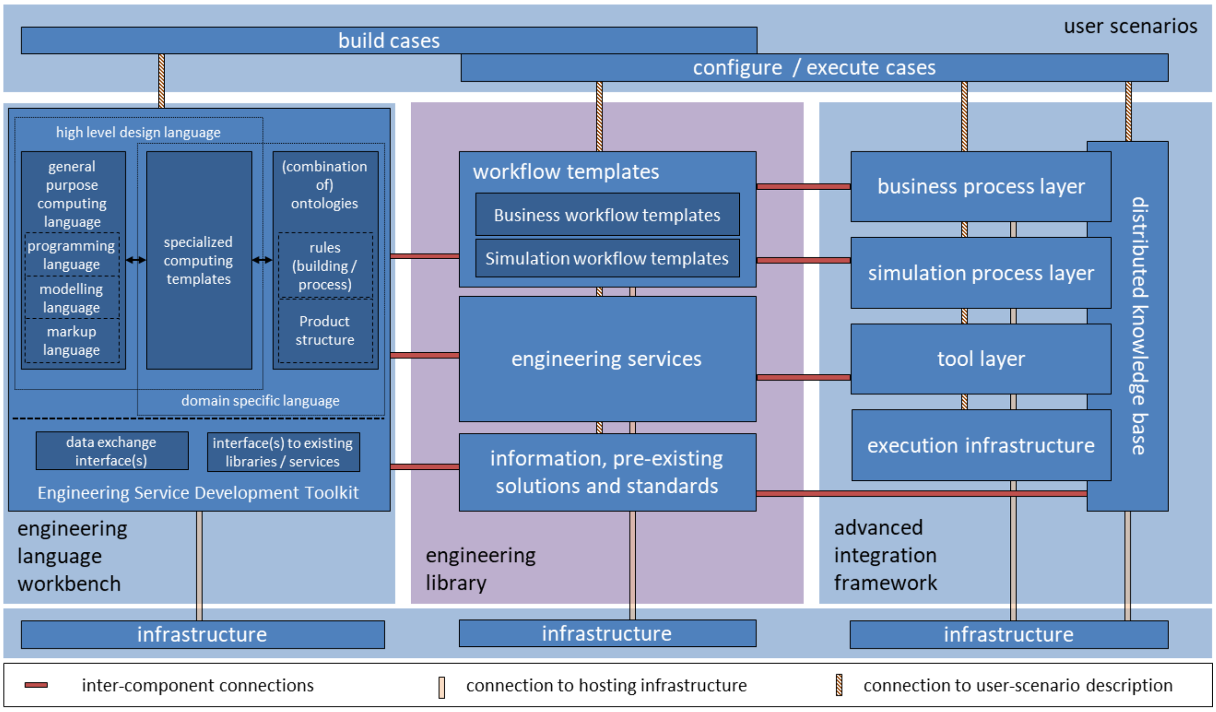

Figure 3: Detailed overview of the IDEaliSM framework architecture

The main innovations related to the three main technical components of the IDEaliSM framework architecture related to integration frameworks can be summarized as follows:

- **A hybrid workflow system:** The AIF provides a unique environment to setup and execute hybrid workflows[10]: it provides seamless integration of manual activities from any business process with automated simulation and optimization workflows and as such strengthens the integration between engineers and tools in MDO problems.
- **Remote service execution module:** To support in the integration of distributed engineering services, on local secured computers systems, in a cloudified hybrid workflow environment as part of the AIF.
- **Open-source Application Programming Interfaces (APIs):** To support in the development of engineering services that can be transferred in reusable workflow templates a new set of open-source APIs have been developed.
- **Cloudification of MDO simulation workflows:** To support the on-demand and flexible delivery of the resources required to deploy and execute (MDO) simulation workflows on hybrid cloud environments, including both private as well as public cloud infrastructures
- **Simulation workflow advisory and generation system:** A set of *optimization algorithms and a supporting advisor* to generate feasible and efficient optimization architectures and processes for Multidisciplinary Design and Optimization
- **Live and distributed knowledge base:** A centralized data server is used to share product information across the different domains of the AIF.

These main innovations are explained in more detail in the following sections.

[10] The second ERTP: Hybrid workflow system (Optimus - KE-Chain)

3.2.1 Hybrid workflow system

As can be seen from Figure 3 the AIF is composed of a business process layer, simulation process layer, tool layer, execution infrastructure and a distributed knowledge base. The first three describe what can be referred to as a hybrid workflow system: an environment for hybrid workflows is an innovation providing a perfect mix of business process aspects such as tasks, user assignment and user-interactions and the power of fully automated analysis and optimization workflows that can be operated in batch, and a single and standards based (ISO 10303) repository for all data created throughout one or many processes. The hybrid workflow system is highlighted in Figure 4.

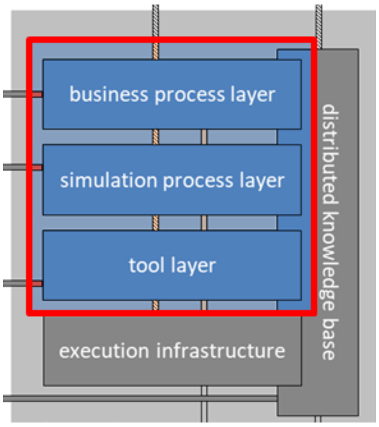

Figure 4: Hybrid workflow system highlighted in the overall IDEaliSM framework architecture.

This hybrid workflow system makes use of new technologies (described in sections 3.2.2, 3.2.3 and 3.2.6) which enable cloudification of the business process to steer the underlying layers. Using workflow cloudification the threshold to perform collaborative design in distributed teams of experts in cross-organizational networks is drastically reduced. The web-based business process layer enables distributed teams and team members from different locations to collaborate together real-time.

The back-bone of the cloudified business workflow and database is a unique standardized data structure (ISO 10303-209) that links product data to the business process. This data structure enables easy, flexible and reusable configuration of the centralized product data model and business process activities. This flexibility and reusability ensures, that the AIF can be used to setup, manage and execute various engineering problems such as electric redesign of a large commercial aircraft or structural sizing of a vertical tail plane of a novel fighter aircraft as demonstrated during the IDEaliSM project.

Another advantage of using cloudified business processes and a live and up-to-date database is the ability to have increased insight in the overall progress throughout the design chain through real-time and transparent progress and status updates. Due to formalization of business process activities, integration of experts and disciplinary tools in a single process work can be performed concurrently. Hereby overall development process lead-time can be reduced.

3.2.2 Remote service execution module

As stated in the sections before, the AIF also enables the configuration and execution of workflows with a distributed nature: the business workflow is cloud-based whereas the analysis and optimization workflows run on local environments and computers with different operating systems. Within the integration framework, the business process layer provides the portal for the users to use such local engineering applications as-a-service via remote procedure calls. To this purpose a remote service execution module has been developed.

The remote service execution module aims to integrate existing applications which do not offer web services using an agent-based shell. This service execution module has been developed based on the engineering platform KE-chain. This service execution module offers a flexible middleware component to enable remote design and analysis capabilities to communicate with interactive workflows modelled in the cloud-based business processes of KE-chain. In this way, the local design and analysis capabilities can be consumed over web-based communication protocols and are used as such as-a-service. Specific roadblocks that have been addressed are IP protection and export compliance of confidential data and the realization of a secure and approved communication protocol using secure WebSocket connections (like https and VPN connections) to access computers and tools from external sites such as the KE-chain cloud environment and the automatic registration of new capabilities that are to be coupled to the integration framework. A schematic overview of the remote execution module is presented in Figure 5. The remote execution module has been adjusted to communicate with the APIs of the business process layer on the one hand and the simulation process and tool layers on the other. The resulting execution module forms the basis for the integration with remote applications as demonstrated in the various use cases throughout the project.

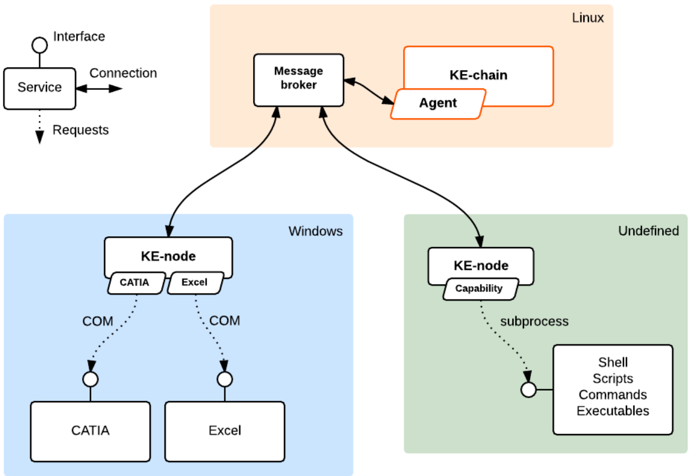

Figure 5: Schematic representation of the integration between KE-chain and KE-node.

This "servitization" of engineering applications realized by the remote execution module provides the ability for non-expert users to operate more advanced tools and analysis workflows through the higher-level user interface offered by the business process layer. In other words, the integration of the analysis workflow layer and business process layer realized by the integration framework lowers the required knowledge and threshold to use engineering applications enabling even non-expert users to make use of advanced tools. In effect, this largely increases the level of automation of product development processes.

3.2.3 Application Programming Interface

Extending the developed remote service execution module is a newly developed open-source Python-based Software Development Kit (SDK)[11]. The SDK enables live and secured exchange of data between registered engineering services and a centralized database. This centralized database ensures a single source of truth of data between engineering services operating in the distributed workflow. With respect to current state-of-the-art solutions the SDK provides flexibility, easy integration of engineering services in a secured way as an enabler for collaborative product development.

A REST web service was defined and implemented to expose the key functionalities of the simulation workflow platform, Optimus, as web services. This web application, called Optimus Workflow Manager (OWM) extends the desktop functionalities of Optimus by enabling the possibility to apply a wide set of operations related to simulation workflows (deployment, execution, retrieval of results) from a remote location and by adding extra features, among others workflow lifecycle management and workflow representation in XML format. A user interface was also developed to expose all these functionalities in common web browser applications.

3.2.4 Cloudification of MDO simulation workflow

The execution infrastructure is a key component of the AIF developed in IDEaliSM. Running MDO workflows characterized by a large number of disciplines can require considerable computational resources due to the highly dimensional design space to be explored and also due to the diversified requirements of the entailed disciplinary tools. The current state-of-the-art approaches, consisting on executing these workflows on either desktop environments or HPC clusters, have several limitations that can be ascribed, for example, to poor scalability, lack of deployment flexibility and limited capacity. To cope with these limitations, a cloudification architecture[12] has been developed to support the deployment and execution of simulation workflows (together with other aspects such as resources allocation and monitoring) on virtualized, hybrid (private, public, or both) computational cloud infrastructures. In this way, the execution infrastructure can be tailored to the specific needs of the simulation workflow used to solve the MDO problem at hand. Hereby a more affordable, easier and earlier access to high performance computing infrastructure is enabled, which improves the adoption of true MDO in industry and answers one of the major bottlenecks described in section 2.4.

3.2.5 Simulation workflow advisory and generation system

Finally, within IDEaliSM an ontology based system is developed to help formulating, formalizing and executing MDO problems. This system of ontologies was defined to encompass and capture the knowledge at different levels of granularity, from the MDO problem formulation and solution strategies to the executable MDO simulation workflow definition to be materialized within the target PIDO platform. The basic principle behind the so-called InFoRMA[13] is depicted in Figure 6. This MDO advisory system enables the engineering of simulation workflow templates, executable through the AIF, which can be stored in the EL for reuse. The MDO advisory system tackles the

---

[11] The eleventh ERTP: Open Source KE-chain Python API (pykechain)

[12] The fourth ERTP: Optimus simulation workflow cloudification

[13] The eight ERTP: InFoRMA (MDO advisor)

issue of limited MDO used in modern PDP by lowering the accessibility level of MDO technology. This is enabled by advising non-expert MDO users on the selection of the most appropriate MDO architectures for a problem at hand. MDO problem formulation is based on the (de facto) standard XDSM, short for eXtended Design Structure Matrix[14]. The generation of executable MDO workflows through an automation of simulation workflow generation can lead to a significant time reduction, estimated in 90% in the IDEaliSM use case, due to the possibility of eliminating the need of manual, repetitive and error-prone human tasks.

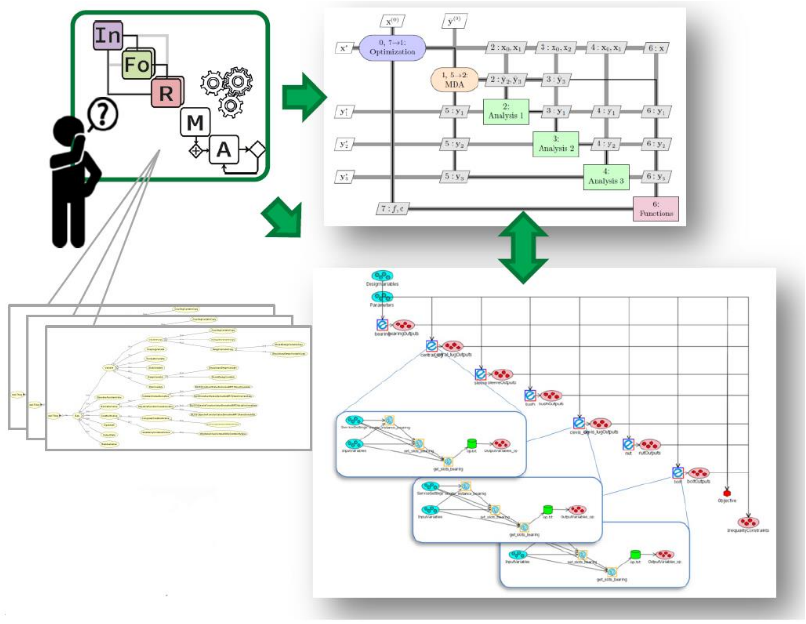

Figure 6: InFoRMA (MDO advisor) basic principles: Advising non-expert users on the formulation of MDO problems using the (de facto) standard XDSM before deploying a simulation workflow in a PIDO application.

3.2.6 Live and distributed knowledge base

One of the major bottlenecks reported in Section 2.4 is related to the way the data is made accessible to the list of users involved in the product development process and to aspects such as data dependency tracking and long-term archival. In IDEaliSM, these limitations have been tackled by adopting a distributed knowledge base (i.e., the EDMopenSimDM solution provided by Jotne) with the following key capabilities:

- a centralized place to store, manage, keep track and archive the data required and generated by the virtual simulation tools involved in the product development process
- an authentication layer to ensure that the stored data is properly managed according to the (often complex) security requirements of the stakeholders involved in the product development process

[14] Lambe, A. B., & Martins, J. R. (2012). Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. Structural and Multidisciplinary Optimization, 2012(46), 273-284.

- full support for the internationally-recognized ISO STEP standardized format (specifically ISO 10303-209 and the emerging edition 2 (AP209e2)) in terms of validation, visualization, export and querying
- a web interface providing mechanisms for engineering programs to remotely access the knowledge base, interact with the product model repository structure and download / upload the required data

To cope with problems related to data dependencies and inconsistencies, specific interfaces have been developed that connect the knowledge base with the business and simulation process components of the AIF. These interfaces enable the storage and retrieval of virtual prototyping data in real time and in a completely automatic fashion together with the possibility of exchanging of data across the different actors of the product supply chain (OEM, Tier 1 and Tier 2 suppliers), therefore reducing drastically the occurrence of errors and inconsistencies.

By using this concept one achieved the objective of single source of truth for product data and processes. Both users and software application could access the information, either by the easy-to-use web browser or by web-services developed in the project. Further, a complete supply chain integration responding to the requirements in data exchange, sharing and archiving and at the same time, providing access control lists meeting partners need in confidentially was documented.

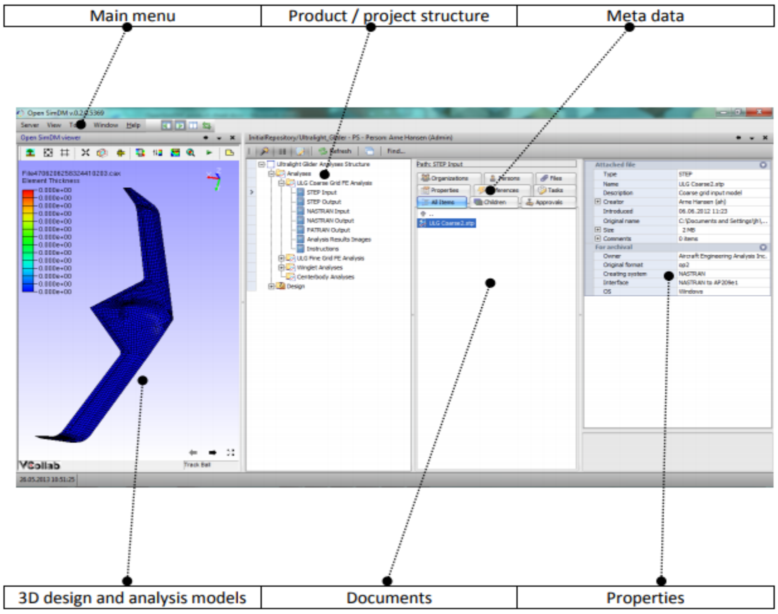

Figure 7: User interface from Cloud based Server (Jotne's EDMOpenSimDM)

3.3 Design languages and standards

The sections below give an update of the SotA of Graph-based Design Languages and Standards.

3.3.1 ISO standards

The main international standard that is applicable to the scope of the project is ISO 10303-209:2014, Multidisciplinary analysis and design (AP209), as mentioned above. The project applied AP209 to ensure interoperability in exchange and sharing of engineering data. The standard lived up to project expectations and resolved the interoperability requirements. However, the validation of AP209 by the consortium revealed about ten issues, such as

1) Updates to the specification documents
2) Use in isogeometric analysis
3) Representation of feedback data in engineering optimization processes
4) Multi-graded material for additive manufacturing
5) Aerodynamic load analysis and use in a multi-disciplinary analysis context.

These were reported to the ISO committee responsible for AP209, ISO/TC 184/SC 4, together with solution proposals. In several meetings with the committee agreement was achieved on the validity of the issues and their solutions. Results are already incorporated in a subset of AP209, that is, the emerging edition 2 of ISO 10303-242, Managed model-based 3D engineering. The remaining updates will be published in edition 3 of AP209, which the project took the initiative to get started in November 2017.

3.3.1 CPACS

The IDEaliSM project contributed an extension to the CPACS[15] (Common Parametric Aircraft Configuration Schema) standard. CPACS is a parameterized data exchange format fitting conceptual and preliminary aircraft design purposes. Its main goal is to standardize the interfaces between engineering services, leading to a reduction in both the effort required for manual data re-formatting as well as the amount of conversion errors. The explicit data exchanged between the different engineering services through CPACS forms a basis for communication between the heterogeneous experts involved in the design process. Due to its applicability to aircraft design problems in academic as well as industrial settings, CPACS is becoming a de-facto standard for data exchange within the MDO community of aerospace design.

Within the IDEaliSM project, major extensions and adjustments of the data exchange format have been developed to extend the utilization of the data exchange format from civil design towards military aircraft design purposes. As major achievement, the complete re-adjustment of the aircrafts' mission definition and performance requirements can be stated. The definition now allows the flexible description of all kinds of design missions and performance requirements, fitting civil, military, manned and unmanned aircraft design. In connection to this, references to the engine and aerodynamic performance databases have been updated to allow studying the effects of combined engine concepts (e.g.: gas turbine and electric propulsion) as well as changing the aerodynamic constellation of the aircraft during the mission analysis.

The resulting, updated schema definition of CPACS is included in the upcoming major release and available to the aerospace design community as an open-source download from the CPACS portal www.cpacs.de.

[15] The third ERTP: CPACS data schema for streamlining data exchange within conceptual and preliminary aircraft design

3.3.2 VEC

Within the project, a possible extension to the VEC (Vehicle Electric Container) is proposed[16]. Additional information on the mechanical properties of harness segments can be stored and reused in subsequent laying simulations. For the determination of the tensile, torsional, and bending stiffness of cable harness segments an approach for the modelling and a methodology for the numerical calculation are developed. The extension provides the following capabilities:

- Integration of mechanical properties into analysis and design tools used during the cable harness design process
- Interfaces to the engineering language used during routing simulations
- Integration of experimentally determined data for validation of the simulated results

3.3.3 Design Languages

The (graph-based) Design Languages (GBDL) were extended by “template” ontologies of abstract geometry, abstract physics and automated wire routing[17]. Under abstract geometry the program independent specification of geometry is summarized, while abstract physics describes the program independent specification of structural mechanics simulation properties. The use of abstract geometry and physics enable easy and reusable construction of design language features in UML to define concrete geometry objects for use in CAD applications. Similarly the definition of FEM or CFD simulation models allow the straightforward computational simulation and analysis without the need to know the vendor-specific data formats of the underlying numerical solvers.

The automated wire harness ontology uses the geometry ontology to describe a wire harness design problem not only by the needed electrical schematics but to also describe the situation of the geometrical environment in order to execute algorithms for automated wire harness design.

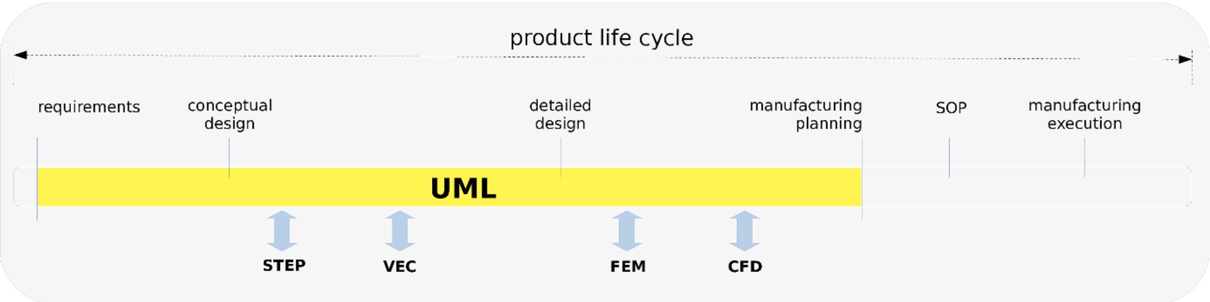

Figure 8: GBDL supported product life cycle

Based on these abstract ontologies engineering services have been successfully build for automated wire harness generation and automated FEM analysis. The figure above on the product life cycle illustrates the consistent use of graph-based design languages in UML throughout the conceptual and detailed design phase. Via several plugins the abstract geometry can be transformed into STEP format, the harness into VEC format and the abstract physics into vendor-specific FEM- and CFD-data formats.

[16] The tenth ERTP: Stiffness calculation of wire harnesses

[17] The fifth ERTP: Wire harness modularized framework