Linköping University | Department of Product Realization
Master Thesis 30 ECTS | Design Engineering and Product Development
Spring 2022 | ISRN: LIU-IEI-TEK-A--22/04312--SE

Automatic Creation of an Aircraft Structural Layout and Structural Analysis Model

A method for implementing design automation in an early conceptual design phase

Master Thesis Report

Philip Brånäs
Nora Enderby

Examiner: Anton Wiberg: *IEI, Linköping University*
Supervisor: Sofia Schön: *IEI, Linköping University*

External Supervisors: Gustav Ehmke, Melker Nordqvist: *Saab Aeronautics*

Linköpings universitet
SE-581 83 Linköping, Sverige
013-28 10 00, www.liu.se

LINKÖPINGS UNIVERSITET

Copyright

Abstract

Aircraft structural layout concept design at Saab Aeronautics utilize thickness optimization to evaluate a structural layout concept. The thickness values can be used to compare concepts to each other, and the best one can be further developed. Today, most of the creation and evaluation of structural layout concepts is manual work. Therefore, there is an ongoing investigation on how to implement design automation to reduce this manual and repetitive work.

The investigation aims to achieve rapid exploration of the design space to find a good base for a new aircraft development. This includes investigating how the synchronization between a structural layout model (SLM) and a global finite element model (GFEM) can be improved.

This thesis contributes to the investigation by exploring the possibilities to implement design automation in the creation of the SLM regarding the fuselage structure. Further, exploring the implementation of design automation in the creation of the GFEM to enable automatic evaluation of concepts. The thesis also explores how the synchronization between the models can be improved.

To structure the thesis work, the software development methodologies of MOKA and RAD were modified and combined. The execution of the thesis was carried out in the software of 3DEXPERIENCE, particularly using the applications CATIA and SIMULIA.

This thesis work resulted in a method for developing and evaluating aircraft structure concept designs with design automation. The new method includes two models with corresponding scripts. The first model developed is a tool for a conceptual designer that enables the creation of aircraft fuselage SLM from user defined inputs. The second model is generated by script which results in a GFEM with a direct connection to the SLM.

To conclude, the developed method enables a faster iteration work of fuselage structural concept designs compared to the current method. The detail level is lower but more consistent and uniform. The GFEM was not able to fulfil its purpose in the developed method due to time limits and software limitations. However, the synchronization between the SLM and GFEM was implemented successfully and contained all critical elements.

Acknowledgements

We would like to dedicate a special thank you our two mentors at Saab Aeronautics, Melker Nordqvist for supporting us throughout the thesis, and Gustav Ehmke for providing good energy and answering any questions that we had. Also, thank you for giving us the opportunity to do the thesis at Saab Aeronautics. We would also like to thank Erik Holmberg and Raghu Munjulury for your close cooperation and support on your expertise topics.

At Linköping University, we would like to thank our academic supervisor, Sofia Schön, for helping us through the academic challenges and providing positive energy. Also, thank you Anton Wiberg for being our examiner and providing your opinions on the work.

Joachim Neguembor, thank you for reading our report and providing good thoughts on how to improve it.

Philip Brånäs

Nora Enderby

Linköping, June 2022

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
3Dx	3DEXPERIENCE
CAD	Computer Aided Design
EKL	Engineering Knowledge Language
FE	Finite Element
FEM	Finite Element Method
FW	Floor and/or Wall
GFEM	Global Finite Element Model
KBE	Knowledge Based Engineering
KP	Knowledge Pattern
MOKA	Methodology and tools Oriented to Knowledge based engineering Applications
OML	Outer Mold Line
PC	Power Copy
RAD	Rapid Application Design
SLM	Structure Layout Model
UDF	User Defined Feature
VB	Visual Basic
VBA	Visual Basic Application

1

Introduction

This chapter introduces the thesis and why it is needed.

Saab AB, as a company, serves the global market with products, services, and solutions from military defence to civil security. The company has four main business areas which are Aeronautics, Dynamics, Surveillance, and Kockums. Saab Aeronautics is a supplier of state-of-the-art flight systems as well as both military and civil aircrafts. At the department of Airframe structural design in Linköping, Sweden, the structural parts of the military aircrafts are developed and designed. The thesis is conducted at this department.

The military aircraft market is a market of high competitiveness. Costumers, such as states, will think more than twice upon what aircraft to choose for their military defence. Therefore, price, performance, and availability must be optimized to keep a high ground on the competitive market.

The development of a military aircraft is a complex task which requires a well-defined way of working and a structured data flow to minimize costly errors. To be able to challenge the market, improvements on the process from concept to assembly are vital. This thesis will focus on implementing design automation in the creation of the structural design of an aircraft in an early stage in order to increase the conceptual design generation effectivity.

1.1 Background

In the beginning of a product development project, the design freedom is high and the product knowledge low. With further knowledge on the design, the design freedom is reduced; see the continuous lines in Figure 1. The goal of a product development process is to maintain a high design freedom as long as possible, while increasing the product knowledge, see the dotted lines in Figure 1 [1]. This may be achieved in many ways and one contributing factor is design automation of CAD models [1]. Design automation can enable a multidisciplinary design approach by easily modifying the model with a set of parameters. This makes it possible to prepare early simulations and optimizations of the product which increases the design knowledge early on.

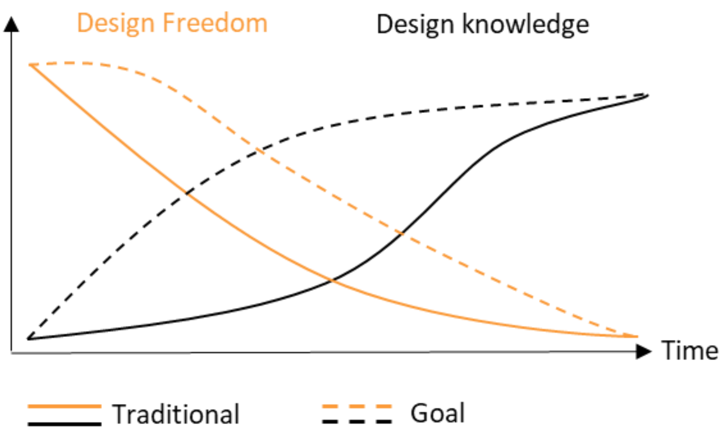

Figure 1: A figure which shows the relationship between design knowledge and design freedom over time in a project. Traditionally, the design freedom starts high but decreases fast and the design knowledge slowly increases throughout the project. The goal is to gather knowledge earlier in time. Adapted from [1].

There is currently a project at Saab Aeronautics called VUproject, which focuses on implementing new methodologies in the early development of aircrafts conceptual design. More specific, the VUproject aims to create a more automated development process regarding aircraft structure layout, simulations, and optimizations such as aerodynamic simulations, weight optimization and structural analysis. Today, a structural layout model (SLM) is created by the design team which contains the structural components in the form of surfaces, lines and points. This model is then used by the stress team as a tool to aid the creation of the Global Finite Element Model (GFEM) and keep the two disciplines synchronized.

The GFEM is used to understand how the aircraft, or larger parts of the aircraft, will distribute the load from different load cases. When the internal loads are identified, various types of optimizations can be executed e.g., minimizing weight. The results from the optimizations will then be fed back to the SLM and changes may be implemented, see loop in Figure 2.

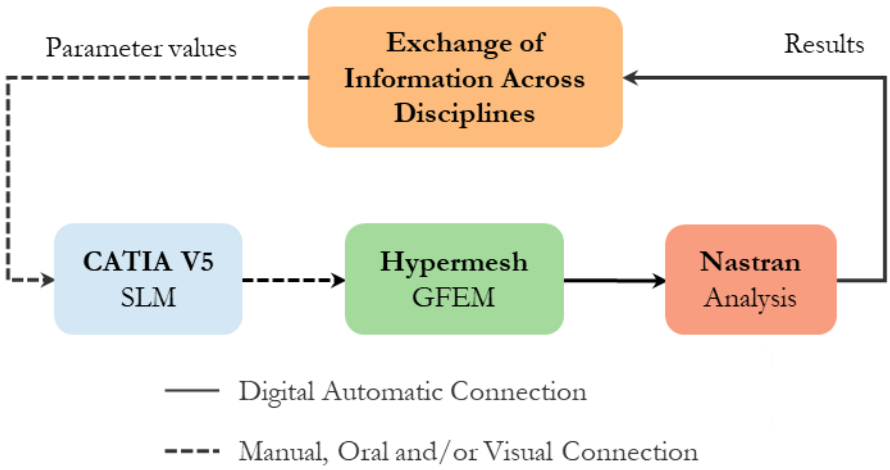

Figure 2: A simplified overview of the current loop analysing and optimizing the fuselage structure at Saab.

In the current aircraft project, Gripen, the SLM was created in a late design stage which enabled the loop but limited functionality due to the current deep nested methodology. The VUproject aims to investigate the possibilities of automating the creation of the SLM, GFEM, analysis, and feedback of the result information. This would enable analysis and optimization and increase the design knowledge early on in a project. This thesis will take part in this investigation of the problem areas presented below.

1.1.1 Problem Area SLM

Saab Aeronautics sees an opportunity of improvement at the area of design automation in an early concept design phase. Especially when developing the airframe structural layout model (SLM) in a CAD software. At this phase, the level of detail is low, and the constraints are comparably few. A concept will consist of basic geometries such as surfaces, planes, lines, and points which creates the skeleton of the aircraft. Today, the creation of the SLM is manual work which will limit the exploration of concepts as it is time consuming and restricts the number concepts that can be evaluated. Therefore, there is a need to investigate how the creation of the structure can be more automated in order to increase the design exploration.

1.1.2 Problem Area GFEM

Today in the Gripen project the SLM is mainly a model to visualize the design of the aircraft structure. It is used by the stress team as a tool to synchronize the GFEM with the design teams SLM visually. The stress team manually imports the geometry from the SLM to aid the creation of the GFEM. Analysis on the GFEM is then performed and conclusions from the analysis is then discussed with the design team and possible changes on the aircraft structure are made. This is a highly manual process and there is room for improvement by implementing design automation. Today the SLM is created in CATIA V5 and its geometries is exported to Hypermesh where the GFEM is created.

At Saab, studies have been made on how to automate this process with the existing software. This has mainly been successful but without a robust methodology behind it. Furthermore, the life span of CATIA V5 is reaching its end and is being phased out in the industry. A possible successor may be the multidisciplinary platform 3DEXPERIENCE (3Dx). If implementing 3Dx, or similar software, Saab sees an opportunity to improve and implement method such as design automation. A vision of the VUproject is to automate the process in Figure 2 by using 3Dx and its multidisciplinary properties. A proposed automation loop can be seen in Figure 3, where the creation of SLM and GFEM is made in 3Dx while the calculations and optimizations are made in the Nastran software. By integrating both the SLM and GFEM creation in 3Dx, an investigation of a stronger synchronization between the models can be made.

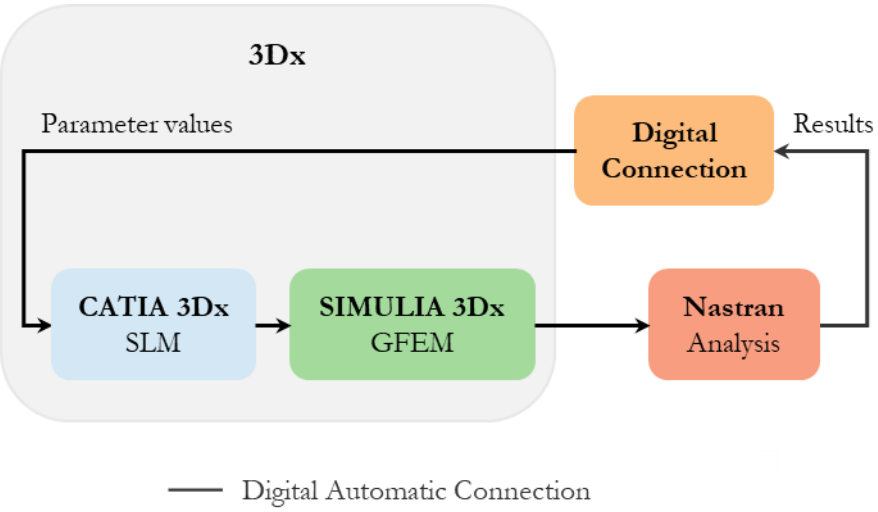

Figure 3: A simplified overview of the desired loop analysing and optimizing the fuselage structure.

1.1.3 Thesis opportunity

The thesis investigates how design automation can be implemented in the creation of a SLM in an early conceptual design phase. Thereafter, the automatic creation of GFEM and the synchronization between the SLM and GFEM will be investigated. The thesis will be conducted in 3Dx.

1.2 Purpose and Goals

The purpose of this thesis is to investigate how design automation can be implemented in 3Dx to create a SLM and GFEM in order to facilitate the concept development phase. Furthermore, to investigate the connections between the departments of design and analysis in this early stage to strengthen the collaboration and synchronization between them. The goal of the thesis is to research what information must be present in the SLM in order to create the GFEM. Also, investigate a robust method of how to create and synchronize these models in 3Dx. This is firstly achieved by creating model with automatically instantiated features to form the structure of the fuselage and secondly by creating a proof of concept of a GFEM.

1.3 Deliverables

The previously presented goals for the thesis will lead to the following deliverables:

- A script that automatically instantiates the basic shape of the fuselage structure in the SLM, containing the components:
    - Frames
    - Floors
    - Walls
- A method on how to automatically create a GFEM in 3Dx.
- An instruction on how to use the created SLM and GFEM as well as the work method needed to create such models.

1.4 Research Questions

In the competitive markets of engineered products, retaining and reusing knowledge is essential to stay on top. The time is often limited, the disciplines are many, and the departments are large. In order to improve the effectiveness in these conditions, three research questions are formulated and answered in this thesis:

RQ1: How can an automated parametric model be created to enable structural concept design?

RQ2: To what extent can design automation be implemented in 3DEXPERIENCE to create an automatic process between a Structural Layout Model and a Finite Element Model?

RQ3: How does the developed method compare to the current method in terms of time for design creation, model detail level and, synchronization between a Structural Layout Model and a Finite Element Model?

1.5 Limitations and Delimitations

The thesis is limited to 20 weeks of work by two master students. In order to satisfy Saab's confidentiality, the work is carried out purely at the office. Only company approved software is utilized in the project. Communication between the majorities of colleagues is carried out through online meetings when Covid-19 restrictions are active. The models used in the thesis are representative but not identical to Saab Aeronautics, this to avoid company confidentiality issues.

Delimitations are presented below:

- The thesis will focus on the conceptual design stage for the structural design development of future projects.
- The automation of the information flow to the SLM and from the GFEM will not be investigated.
- CATIA 3Dx and SIMULIA 3Dx will be the software apps investigated.
- The scripting languages of visual basic and engineering knowledge language will be used.

- The work is carried out in the early design conceptual stage where the level of detail of the overall aircraft design is typically low. Thus, only the fuselage outer mold line, air intake, engine, and wings are taken into consideration when creating the fuselage structure.
- The structural components that shall be instantiated are limited to frames, floors, and walls.
- The thesis will not implement attachment points and hinge joints in the SLM nor GFEM which otherwise is present.
- The structural components geometries will be represented by surfaces.
- The type of mechanical connections between components will not be investigated. The connections are assumed to be static with no degree of freedom.
- The work will be carried out in one CAD product to avoid time consuming handling of PLM references.

Due to the complexity of Saab Aeronautics design processes, more delimitations are made alongside obtaining more knowledge and executing work during the project. The additional delimitations can be found throughout the report in their respective area.

2

Theoretical Framework

This chapter presents information gathered from external sources.

2.1 Aircraft Structure

Commonly an aircraft consists of a body, called a fuselage, which holds the parts of the aircraft together and carries the payload. For example, the cockpit, tail, wings, and engines are held by the fuselage and extra fuel tanks could be carried. [2] In Figure 4, a general shape of an aircraft can be viewed.

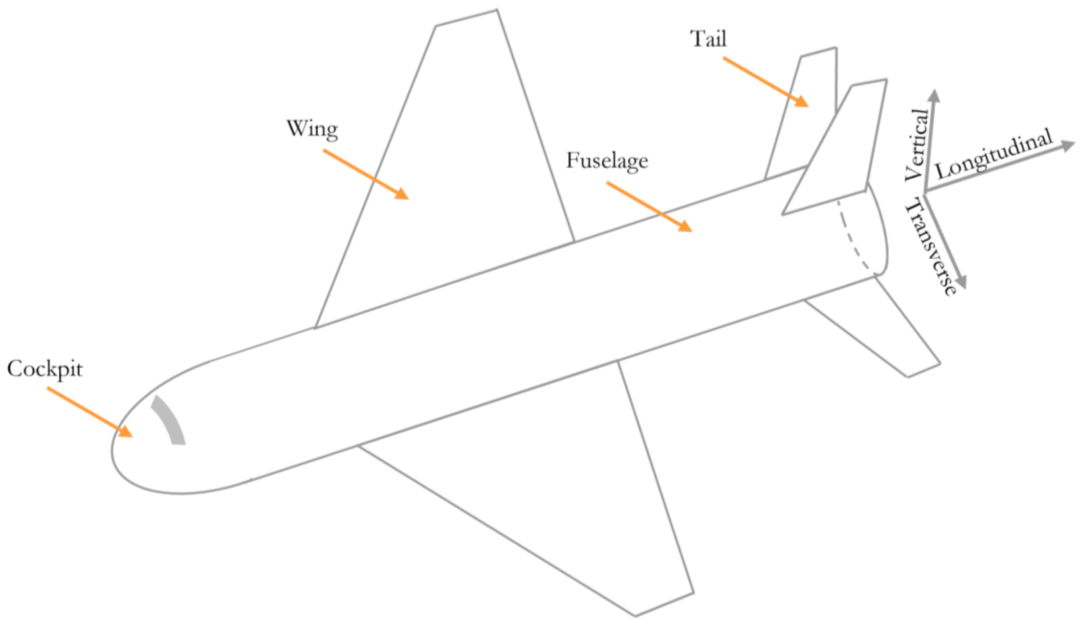

Figure 4: A general shape of an airliner aircraft depicted with some of the mayor components: cockpit, wings, fuselage, and tail. Adapted from [2].

To distribute the loads created from these parts and the payload of the aircraft, it needs a structure. The aircraft structure does not only distribute loads but also protects the passengers and equipment inside it and help establish an aerodynamic shape [3]. For most aircrafts, the outer surface, also called outer mould line (OML), is made by thin-wall structures. The OML is supported by longitudinal beams called stringers and transverse frames [3], see the simplified fuselage with representative placement of the structural components in Figure 5. The figure also shows walls and floors. These components are briefly described below.

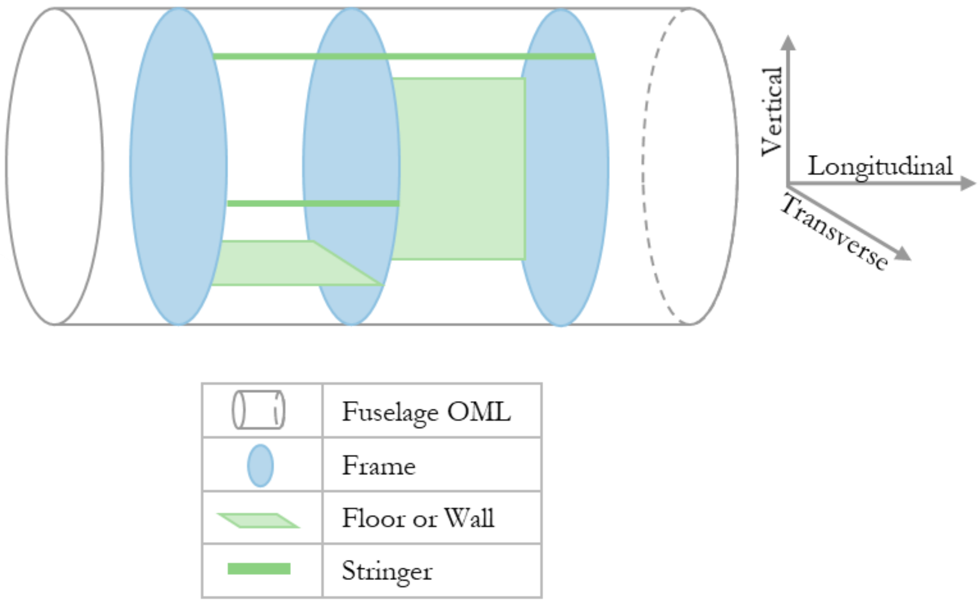

Figure 5: A simple illustration of a fuselage with representative position of the frames, stringers, floors, and walls.

2.1.1 Fuselage Outer Mould Line

The grey outline in Figure 5 represents the OML of the aircraft fuselage. The OML itself have good characteristics counteract shear and tensile loads but less good for compressive loads. Aerodynamic pressure loads are transferred to the inner structure through the OML surface. [3]

2.1.2 Frame

A frame can take different shapes, a simple example is a transverse sheet covering the cross sectional area of the fuselage OML, see the round blue surfaces in Figure 5. The main purpose of a frame is to absorb and transmit loads from systems attached externally or internally to the fuselage. Frames also increase the buckling stress that the fuselage and its structure can handle. The cross sectional shape of the fuselage OML is also ensured by the frames. [3]

2.1.3 Stringer, Floor and Wall

Stringer, floor, and wall components are placed longitudinally in the fuselage to resist the required axial and bending loads, see green rectangular shapes in Figure 5. The main floors and walls also absorb shear and torsional loads from external and internal systems attached to the fuselage. [3]

Another purpose of floor and wall components is to create enclosed volumes in the aircraft that have different environmental requirements. For example, the engine is enclosed to make a fire hazard safe environment to protect the rest of the aircraft systems. The walls and floors can also be used to attach equipment, systems, and other items to a flat surface.

2.2 Design Automation

Design automation (DA), in the context of CAD modelling, is implemented to reduce repetitive tasks of a design engineer [4]. Repetitive design features can be reused and instantiated by script and in turn allow for a faster generation of design concepts [1]. DA can generate designs which follows rules and are therefore highly related to Knowledge Based Engineering (KBE) which aims to capture engineering knowledge and implement it to the design [4]. DA can also introduce automation to an optimization context. By automatically changing design parameters, the design can be used in a design optimization [1].

A highly parametric CAD model is preferred in order to implement DA to a higher degree [1]. To enable a parametric CAD model, the geometry should be able to transform when the design changes. A parametric modification can be divided in to two categories, morphological and topological transformation [4]. These categories are explained in the sub-chapter below.

2.2.1 Morphological and Topological Transformation

Morphological transformation of a geometry refers to changes in shape. The geometric shape can be either fixed, parameterized, relation based, or script based. Topological transformation on the other hand refers to location and instantiation of geometries. This involves adding and removing geometries. A topological transformation can either be manually instantiated or automatically instantiated with script. [4]

The two transformations may also be combined to obtain a highly flexible model by instantiating a geometry which allows morphological transformation. With the correct instantiation inputs, a geometry that can change shape depending on the location of instantiation is obtained.

2.3 Structural Analysis

This chapter will give a brief introduction to structural analysis topics used in the thesis such as Finite Element (FE) model in a computational software and Global Finite Element Model (GFEM) of an aircraft structure.

2.3.1 FE Model in a computational software

There are multiple things that represents the behaviour of a physical object. These are the geometry, material property, boundary condition, initial condition, and loading condition. An engineering system can rapidly become complicated to solve by analytical means. Therefore, the Finite Element Method (FEM) in a computational software has become popular due to its reliability, practicality, and robustness. [5]

To create a FE analysis in a computational software, four entities must be present:

1. Geometry: The geometry can be represented by either 3D solids or by 1D and 2D solids. FE models of 1D or 2D solids are more efficient and computationally cheaper than a FE model of a 3D solid but causes a larger estimation of reality. [5]
2. Mesh: The mesh is the entity that discretize the geometry and divides it into finite elements. There are various techniques to mesh a geometry when it comes to the shape of the elements. For example, the elements can be triangular or quadrilateral. Here, quadrilateral elements show a higher accuracy of simulation results but may be more computationally heavy. [5]
3. Material property: The material property can be applied to the geometry and mesh in various ways depending on the user needs. A 3D solid needs only a specified material while a 2D solid may need material and thickness specifications.
4. Boundary, initial, and loading conditions: The conditions play a big role in the simulation. The object to be analysed must have a specified boundary condition and a load case to be applied to the object. Thereafter, an initial condition must be present. [5]

Entities 1 to 3 forms a FE model as they have no connection to simulation specific criterions. When the boundary, initial, and loading conditions are present, an analysis can be made on the FE model.

2.3.2 GFEM of an Aircraft Structure

A Global FE Model is a FE representation of a larger system e.g. a boat structure or an aircraft structure. The model has a low degree of fidelity and generally consist of major load carrying components. In the case of an aircraft structure, a GFEM generally contains idealized frames, walls, floors, stringers, and outer surfaces in form of bar elements and shell elements. [6] An example of a GFEM can be seen in Figure 6. The GFEM is used to perform structural analysis on the entire aircraft or larger parts of the aircraft such as a wing. This to identify internal load paths generated from external load cases.

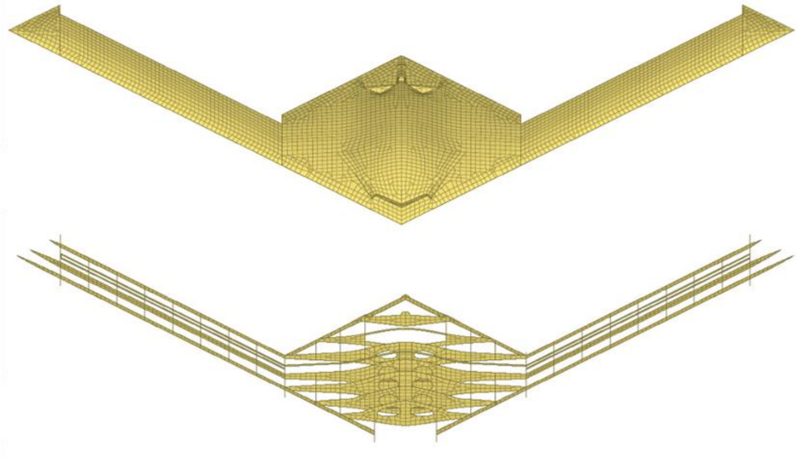

Figure 6: Example on a GFEM of an aircraft structure. The upper representation shows the outer surfaces, and the lower shows the inner structure of the aircraft.

In a GFEM, all the components have the entities; geometry, mesh, and material property, described in the previous chapter. But in such a model, the components have to be connected to each other in order to get realistic results from a simulation. There are various ways to connect meshed components to each other. One method used at Saab Aeronautics is to connect the mesh of the components in a way that the intersecting elements shares the same mesh nodes. An example is seen Figure 7 where two components with equally sized mesh elements, intersect with each other. At the left, the mesh nodes (red and yellow) are independent of each other and therefore the two components are not connected. At the right, the components share the same yellow nodes, thus, the components are connected.

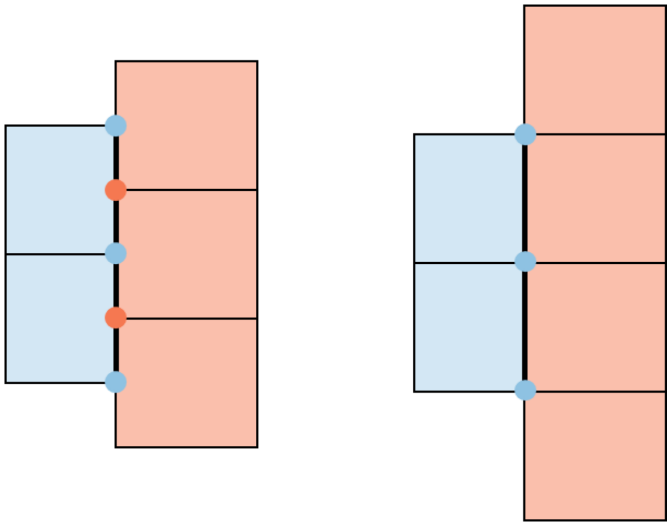

Figure 7: Two examples of two intersecting meshed components. To the left, the mesh nodes are not shared while to the right, the mesh nodes are shared between the two components.

2.4 3DEXPERIENCE

Dassault Systémes, also called 3DS, is a French company that provides the industrial sector with 3D modelling solutions, PLM systems etc. PLM stands for Product Lifecycle Management and is a system which holds all the information about the product throughout the entire lifecycle. 3DS has developed some known CAD software such as CATIA and SOLIDWORKS. [7]

In 2012, 3DS introduced a new platform called 3DEXPERIENCE, from here on referred to as 3Dx. 3DS's vision is to gather their product portfolio into this one single platform. 3Dx is intended to act as a single source of data where individuals, teams, departments, and external collaborators get their information. [8]

3Dx platform is divided into four sectors: 3D modelling applications, Simulation applications, Social and Collaborative applications, and Information Intelligence applications. This thesis will be conducted in the 3D modelling application 3Dx CATIA and the simulation application 3Dx SIMULIA.

2.5 3DEXPERIENCE CATIA

3Dx CATIA is a CAD application powered by the 3Dx platform. It is the next generation after the CAD application CATIA V5. The main difference is the PLM system application: ENOVIA. In the later version of CATIA, ENOVIA is integrated in the 3Dx platform, providing a web-based PLM. In this way, the data is stored in one central location which can be accessed from anywhere. [9] Other than that, some changes have been made in user interface and functionalities, but the principle is approximately the same.

One assumption made in this thesis is that the reader knows the basic principles of CAD applications and 3D modelling. Thus, the text will include more CATIA specific knowledge that is required to fully understand the conducted work.

To enable design automation, a 3D model must be highly controlled by parameters. This put requirements on the modelling. To achieve design automation, the thesis will incorporate morphological and topological transformations. There are multiple ways of accomplishing this in CATIA. In this thesis it will be done using Generative Shape Design, instantiation of features, Knowledge Pattern (KP) script with Engineering Knowledge Language (EKL), and Visual Basic (VB) Macros. These features are introduced in the sub-chapters bellow.

2.5.1 Instantiation of Features

In CATIA, there are a few ways of instantiating features into a model. By instantiating features onto well thought out references, one can further create a script which then automatically instantiates the feature. This is one step of achieving design automation. This chapter presents two ways of instantiating features, these are named Power Copy (PC) and User Defined Feature (UDF). For both functions there must be a set of input geometries i.e., points, lines, surfaces etc. The difference is that a PC will instantiate every geometrical element defined in the PC while a UDF will only instantiate a user defined output. See an example below.

In Figure 8, a Surface is created from two lines, Line1 and Line2. Each line is created between two points.

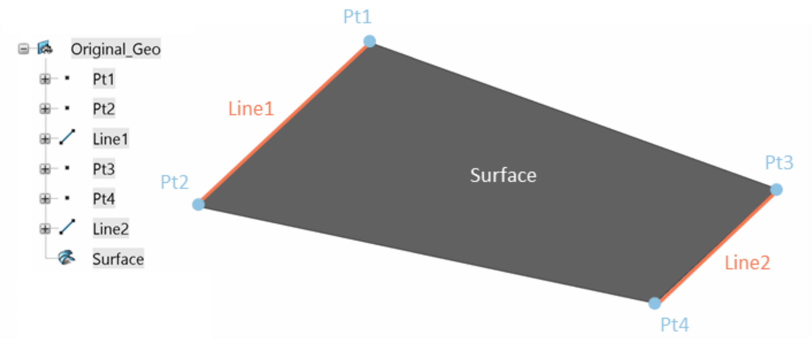

Figure 8: A surface created from two lines. Each line are created between two points.

If one would like to instantiate the Surface by selecting four points as input, a PC would also include the lines since they are needed to create the Surface, see the left in Figure 9. While a UDF would only instantiate the Surface because that was the user defined output, see the right in Figure 9. The output of a PC are regular modifiable geometrical elements. The output of a UDF is a so called "User Feature" which only displays the defined output, this may be geometries as well as parameters.

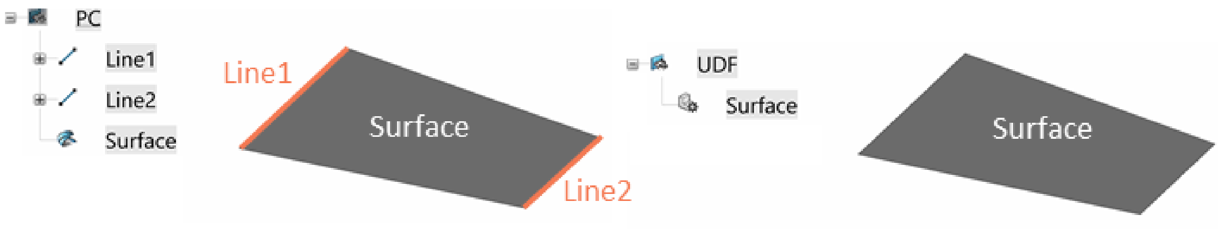

Figure 9: The figure shows the difference between PC and UDF instantiation.

2.5.2 Knowledge Pattern versus Visual Basic Macros

In CATIA there exists multiple features to automate repetitive tasks of engineering design work by script. Two examples are Knowledge Pattern (KP) and Visual Basic (VB) macros. Both features have the functions of, for example, changing parameter values, instantiating UDFs, and obtaining attributes of geometrical elements. Often a series of these functions is desired to be performed in the same sequence multiple times. This is when the KP and VB macros features can reduce manual designing by grouping these functions together in a script to be executed by one click.

Sometimes the dissimilarities of KP and VB macros make them suitable for different applications. A few examples of dissimilarities are:

- **Handling of instantiated features:** a KP saves a link of the instantiated feature in an internal list of the KP. Through this list the KP keeps track of where, how many times, and with what references the feature is instantiated. When removing an instantiated feature, either by code or manually, the KP removes all the related entities of the feature, and the list is updated. [10] VB macros on the other hand has no tracking of instantiated features. Therefore, the removal process of instantiated features quickly gets complicated as a removal function must be made from scratch.
- **Program language and its capabilities:** KP uses the programming language of EKL [10] while VB macros uses the language Visual Basic for Application (VBA) [11]. VBA is a part of Microsoft's programming language VB and is a broad programming language which contains many functionalities (such as creation of arrays and collections). EKL is somewhat limited when it comes to basic programming functionalities. Also, VBA is more documented than EKL.
- **Operational areas:** KP can only operate inside 3Dx CATIA and may only handle in-session data. It cannot directly access data from text files or Excel files. A KP script is located inside the CATIA part modelling tree. By being that close to the CAD model, KP allows to double klick on CAD features to identify them in the KP editor which is convenient. While in VBA, it takes more code to identify CAD features in the tree. However, VBA can operate both inside and outside of 3Dx, i.e., it can access data from Excel files. VBA can also run a KP but not the other way around.
- **Amount of code:** KP typically needs less code than VB macros when it comes to instantiate, create, and modify features. Much due to the reasons described above. However, KP does not support user defined functions to minimize repetitive coding. This is supported in VB macros.

There also exist a functionality in 3Dx called VB Action which allows VB code to be present in the CATIA part modelling tree just as KP. However, VB Actions has less functionality compared to VB macros in the VBA editor. For example, Collections and 2 dimensional arrays.

2.6 3DEXPERIENCE SIMULIA

3Dx SIMULIA is a simulation application that carries the capability of preparing and performing simulations. A few examples of simulations that can be performed are structural, fluids, and multi body simulations. [12] An example of preparing a surface modelled product is presented below.

The preparation starts with the designed product, for example the one in Figure 10. In the tree under the Physical Product, *oProduct*, the 3D shape representation of it is seen, *oProduct3DShape*. The surface geometry is contained within oProduct3DShape.

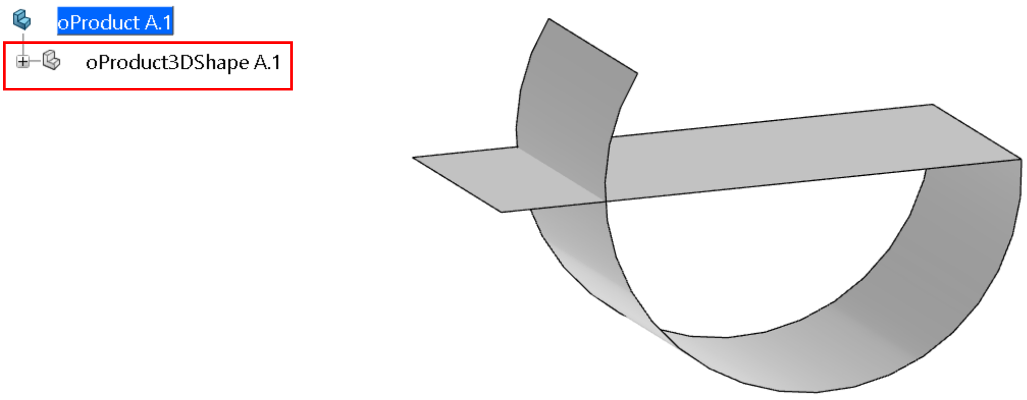

Figure 10: An example product, oProduct, and its tree structure containing the geometry in a 3D shape representation, oProduct3DShape.

By entering the 3Dx SIMULIA application, a FEM representation can be created. This FEM representation, *oProductFEM*, is also gathered in the oProduct tree as seen in Figure 11. oProductFEM uses the oProduct3DShape directly as a geometrical reference. This means that if the 3D representation would be changed, the FEM representation will follow. In Figure 11 it is seen that the oProductFEM contains two different features: Surface Mesh (under Nodes and Elements) and Shell Section (under Properties). A Surface Mesh feature is created for each surface of the product and in this feature the mesh element type and size are defined. A Shell Section feature is also created for each surface. This feature contains further properties of the products surfaces such as thickness, thickness direction and material.

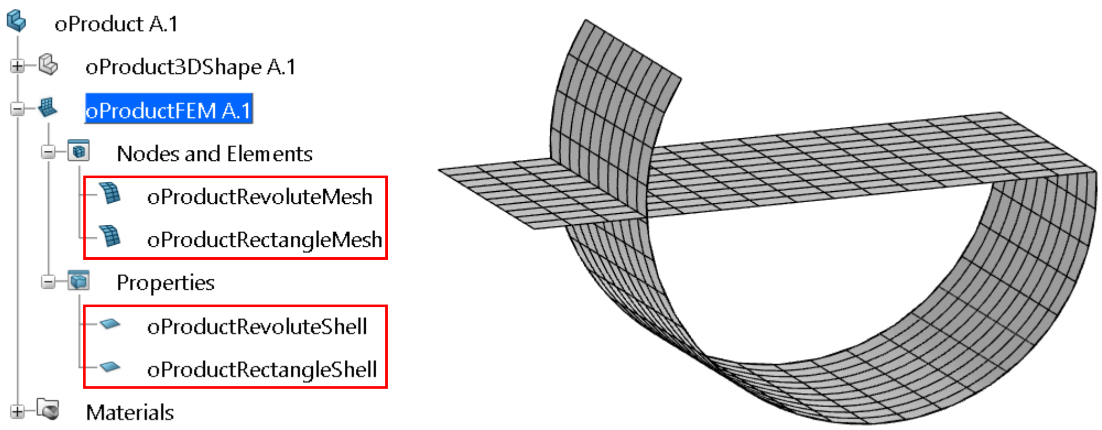

Figure 11: To the left the tree structure is shown after a FEM representation, oProductFEM, is added.

2.7 Knowledge Based Engineering Methodologies

In the aerospace industry there is a constant challenge to keep down the cost and still minimize the lead times. Optimization and Design Automation are suitable tools that can obtain a successful outcome for these challenges. [13] Knowledge Based Engineering (KBE) is an enabler of Design Automation and thus, a structured methodology that focuses on KBE applications is needed. A common methodology for KBE applications is MOKA.

2.7.1 MOKA

Methodology and tools Oriented to Knowledge based engineering Applications (MOKA) was originally a European project to catch up with the fast-evolving KBE technology. [14] MOKA was created to describe a standardized way of developing KBE applications and focuses on how to capture the engineering knowledge to do so. The six phases of MOKA can be seen in Figure 12.

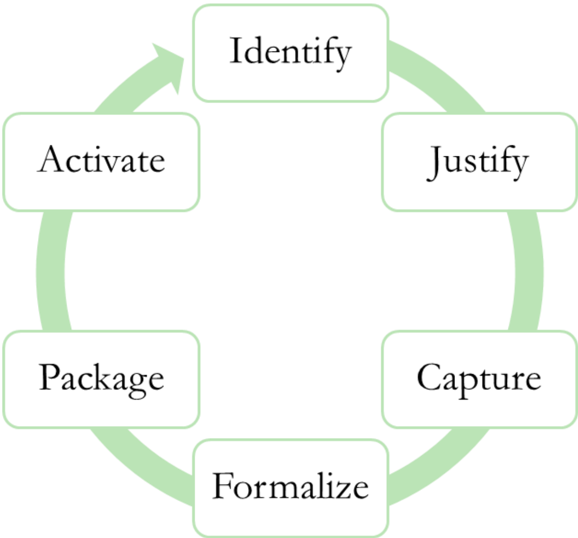

Figure 12: The iterative cycle of MOKA, starting with the phase Identify and ending with Activate.

Stokes [15] describes the six phases as following:

- **Identify:** The Identify phase focuses on getting an understanding of the company needs and limitations. It also focuses on specifying the type of KBE application.
- **Justify:** In the Justify phase, the recourses available for development will be identified and a project plan is created to validate the scope. The assessment of risks is also an important task.
- **Capture:** In this phase, the method for collecting raw knowledge is explained. Collection of raw knowledge such as explicit and tacit knowledge. Explicit knowledge focuses on standardized knowledge which can be found in reports, standards, etc. Tacit knowledge is more personalized knowledge such as know-how and skill. The raw knowledge is gathered in an informal model.
- **Formalize:** The phase of Formalize focuses on translating the raw knowledge from the informal model to a formal model. This is done by implementing the MOKA Modelling Language (MLL)**.**
- **Package:** The KBE application is developed by translating the formal model to programming code.
- **Activate:** In the Activate phase, the implementation of the KBE application is proceeded. Testing and evaluation are executed in this phase. If improvements can be made, a new cycle start is triggered and the Identify phase becomes active.

According to both Curran et.al [13] and Sandberg [14], MOKA focuses more on the CAPTURE and FORMALIZE phase rather than the actual development and implementation of the KBE application.

2.8 Software Development Methodology

When developing a KBE application, the need of a software development methodology may be of complimentary use. The software development methodology investigated in this project is Rapid Application Development which is presented in the upcoming section.

2.8.1 Rapid Application Development

Rapid Application Development (RAD) is a methodology for developing software applications. The aim is to enable a faster development process and a lower development cost [16]. Berger et.al. [16] mentions that the RAD methodology focuses on prototyping and user involvement, using short iterative sequences of building and testing. They continue by stating that the process enables an early visibility of the application and an early validation from the user. An overview of the process can be seen in Figure 13. According to [17], there are four phases of RAD, these are presented below:

- **Outlining Requirements:** The objective of the project is decided among relevant members and agreed upon. Project needs and challenges are identified, and a project plan is conducted.
- **User Design and Input:** This phase is applied in short sequences that enables an iterative process. The sequence consists of three steps which are Prototype, Test, and Refine. It is a continuous phase where the application is developed in the Prototype step, while repeatedly having input from the user in the Test step. The application is refined until a final product is approved.
- **Construction:** This phase act as an output from each sequence. The feedback from the user is checked whether it needs further development iterations or if it is approved. If the application is approved it continues to phase 4, Finalization.
- **Finalization:** The developers round off the project by applying final tests. An interface is polished and user training is conducted.

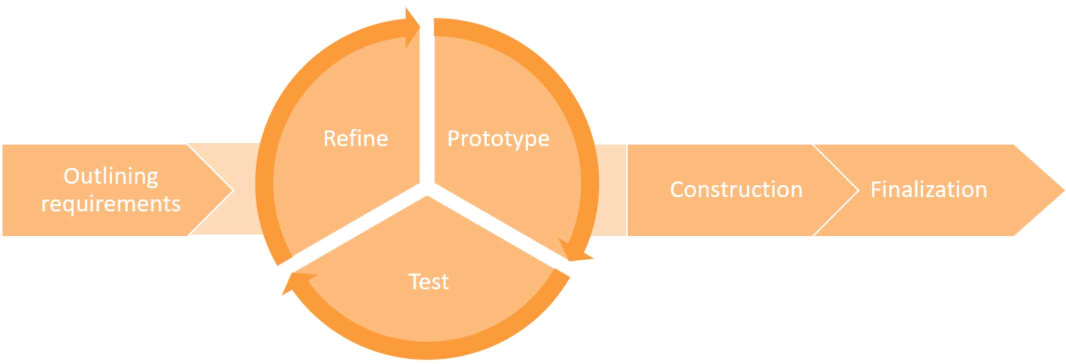

Figure 13: The four stages of RAD are Outlining Requirements, User Design and Input (interpreted as the loop), Construction and Finalization.

In [17], the advantages of RAD is, for instance, flexibility and adaptability to new inputs. It also takes advantage to the user input early in the process which avoids large amount of feedback later in the project when the design freedom is low. The disadvantages explained by [17] are for example that the method is highly dependent on a reliable and helpful user group. If the feedback from the user is not helpful, the project could appear in a standstill. They also mention that RAD is not suitable for large-scale teams where planning and control is highly required. RAD requires a highly modular project.

3

Methodology

This chapter presents the methodology of the thesis work with the corresponding five phases.

To provide the deliverables and answer the research questions two models are created, that is the SLM and GFEM. The methodology for the thesis work is a combination of MOKA and RAD methodologies with applied modifications to fit the current project. The project process consists of five phases, phases 1, 2, 3 and 5 are imported from MOKA while phase 4 is imported from RAD, see Figure 14. The process is linear with the exception of the RAD Loop which will be iterated several times until continuing to Activate. This layout has been chosen due to MOKA's descriptive knowledge capture and RAD's effective way of implementing a software application. Here, the RAD methodology will be used to create scripts to enhance design automation in the models rather than developing an application. The three first phases will focus on project planning, gathering knowledge and information, and structure the findings. The two last phases focus on developing and implementing the models. The aim of the process is to fulfil the deliverables and further answer the thesis research questions. The phases Identify and Justify, Capture, Formalize, RAD Loop, and Activate are explained in the following subchapters.

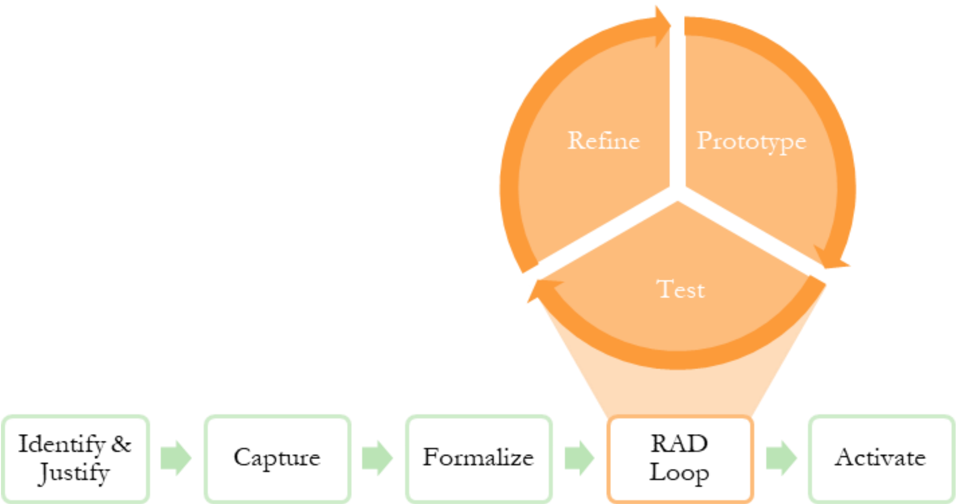

Figure 14: The methodology of the thesis project based on MOKA with inspiration from the RAD methodology.

3.1 Identify and Justify

The first phase of the thesis work focuses on identifying and validating the project scope, goal, as well as a project plan. To identify the scope, several meetings and discussions with relevant co-workers and managers are held. Relevant people for potential knowledge capture are identified. After identifying the scope, several limitations and delimitations are stated in order to justify the project. The primary findings from this phase are presented in chapter *1.1 Background*.

3.2 Capture

To investigate and develop a KBE influenced model data, information, and knowledge has to be gathered. The initial information from the Identify & Justify phase are used to understand what knowledge needs to be captured. By collecting knowledge, the ability to create, evaluate and improve a KBE model is increased. According to MOKA, see chapter *2.7.1 MOKA*, knowledge is gathered in explicit form and tacit form. In order to capture both explicit and tacit knowledge, literature study and interviews are conducted. Information from the literature study is needed to have relevant interviews while interviews and meetings are needed to identify relevant theory areas for the literature study. Therefore, the interviews and literature study are performed simultaneously.

3.2.1 Literature Study

The explicit knowledge is gathered in the literature study. Old thesis reports and interviews are used to identify possible theoretical areas relevant to the thesis. The literature is gathered from the university library, internet sources, Saab internal documents, and old thesis works in the topic of KBE. Relevant literature is read and analysed to get a specified knowledge for different theory areas.

The result of the literature study is specified knowledge on the areas affecting the thesis work. The explicit knowledge is used together with the tacit knowledge to create an informal model to structure the findings. The literature study may be divided in to two areas, external knowledge, and internal knowledge. The external knowledge are presented in chapter *2 Theoretical Framework*, while the internal knowledge are presented in chapter *4.1 Capture and Formalize*.

3.2.2 Interviews

Interviews are done to capture the tacit knowledge. It is also done to identify the company needs. In order to ask relevant questions, the findings from literature studies are used. The interviews are mainly done in a semi structured manner and by traditional meetings. An important requirement for a successful interview is that the person who are being interviewed has the desired engineering knowledge [15]. These persons are identified in the Identify and Justify phase.

In a semi structured interview, the interviewer has prepared relevant questions and use them as a guideline to lead the conversation instead of strictly asking the question in a survey manner [18]. In [18], Adams mentions that semi structures interviews are a good choice if you are examining uncharted territory, then the interview may result in leads to further pursue the work. Interviews are conducted multiple times with each person. As the knowledge of the work becomes greater, more detailed questions can be prepared. Notes are taken as the interviews proceed.

The result of the interviews is gathered, analysed, and validated tacit knowledge. Finally, the knowledge is gathered and structured in an informal model. Further, a list of identified needs is created to keep track of the desired characteristics of the deliverables. The identification of needs is inspired by Ulrich and Eppinger [19], where the needs are listed in a table and labelled required and desired. The identification of need is important to ensure no aspects are missed when conducting the thesis.

3.3 Formalize

The Formalize phase is inspired from MOKA, however, implementing the MOKA Modelling Language seemed unsuitable for the thesis work. This because MOKA focuses more on engineering products with multiple engineering rules, while the thesis is to be conducted in a conceptual phase where the design freedom is high. Still, a model to structure the findings from the Capture phase are desired. The informal model from the capture phase is used together with a need identification list to create a formal model where needs are transformed into requirements. Thus, the model becomes more specific which will aid the development process in the RAD Loop.

3.4 RAD Loop

The fourth phase is inspired from the RAD methodology which is explained in chapter *2.8.1 Rapid Application Development*. The element from the RAD methodology that is implemented in this thesis is the iterative design loop User Design and Input, in the thesis called RAD Loop. Due to the thesis time limitation, a fast way to develop and evaluate the solution is required. The RAD Loop consist of three steps; first Prototype, then Test followed by Refine and then back to Prototype, see Figure 14. The loop is iterated in short intervals of a couple of days. The steps in the loop are explained below.

Prototype
In the prototype step, the models are developed. This includes creation of parametric CAD-models and knowledge templates in 3Dx CATIA, and scripting in EKL and VB. The engineering knowledge that where captured and formalized are used in this step to create the models. The first iterations of the loop are manual instantiation of components and manual creation of the GFEM. By further iterations, the automation is implemented in form of scripts.

Test
The Test step is used to test the models and to identify errors, improvements, and successful parts. The testing is done for each loop iteration. The models are examined in a critical way and notes are taken of improvements and errors. Further, evaluation sessions are held with co-workers when necessary to test the models and provide input on further improvements. The improvements are noted and brought into the Refine step.

Refine
In the refine step, the errors are investigated and corrected. Further, the suggested improvements are evaluated and decided which ones to implement. The decided improvements are implemented in this step and the models proceeds to further development in the prototype step.

The RAD Loop iterates until a well working model has been achieved or the time of the thesis comes to an end. The decision when the models are done is made together with the project members of the VUproject. When the models are done, the methodology can be evaluated to answer the research questions.

3.5 Activate

The Activate step focuses on the hand over process of the developed models so that they can be used and studied by other people. This includes cleaning up the created models and scripts as well as creating an instruction manual which describes the models. Thereafter, a final verification is conducted by the thesis workers and relevant co-workers.

The verification aim is to identify the model's limitations, areas of failure, and representativeness of reality. Two types of tests are carried out in the verification: explorative testing and extremity testing. The explorative tests use DOEs to change the design attributes of the models to explore a wide area of the design space, types of failures and run times are noted. The extremity tests are done by manually changing the design attributes and using the models' functionalities in different ways. Discovered unwanted behaviours, types of failures, and limitations of the models are noted. The verification notes are evaluated and used to answer the research questions in the discussion and conclusion.

4

Execution

This chapter contains the result from the capture and formalize stage as well as the execution of the project.

4.1 Capture and Formalize

A literature study was conducted with both internal and external literature. The result from the external literature is presented in chapter *2 Theoretical Framework,* while the result from the internal study is presented below. The external study is mainly done by reading literature regarding aircraft structure, design automation, KBE, structural analysis, 3Dx and more. The internal study is done by reading internal documents about GFEM, SLM, and the overall Saab processes. Further, the main source of knowledge is gathered from semi-structured interviews with experts in relevant areas such as design automation, concept design, structural analysis, and optimization. A list of conducted interviews and the person's expertise are gathered in Table 1. The questions asked in the interviews are tailored to the person's profession. The result from the interviews and readings are presented in the following chapters and summarized as a list of needs and a formal model.

Table 1: List of conducted interviews and what expertise the interviewed person had.

Interview person	Experience
Conceptual Design Engineer 1	Conceptual design and start model knowledge
Conceptual Design Engineer 2	Conceptual design knowledge
Global FE Analysis Engineer 1	Structural analysis and FEM automation
Global FE Analysis Engineer 2	Structural analysis working with GFEM
Design Automation Engineer 1	Design automation regarding aircraft design

The goal of the literature study and the interviews are to gather relevant information and knowledge in order to facilitate the development of the SLM and GFEM. The knowledge and information are used to identify needs and requirements on the models.

4.1.1 Concept development process

To understand the needs and requirements on the SLM to be created, a study on the concept development process at Saab Aeronautics had to be done. The study was conducted to identify at which development phase the thesis would focus on. This would determine the level of detail which are to be implemented in the SLM.

The main purpose for a concept development process is to introduce confidence in stakeholder needs, requirements, and expectations when business decisions are made. The development process for integrated product creation at Saab can be described in a simple way with six steps, see Figure 15. Initial Studies, Concept, Development, Production, Utilization and Support, and Retirement. The new method that the thesis will investigate may cause a change to this process. Therefore, the thesis will focus on both Initial Studies and Concept. This is because an automated SLM may be used in an early stage as well as in later stages.

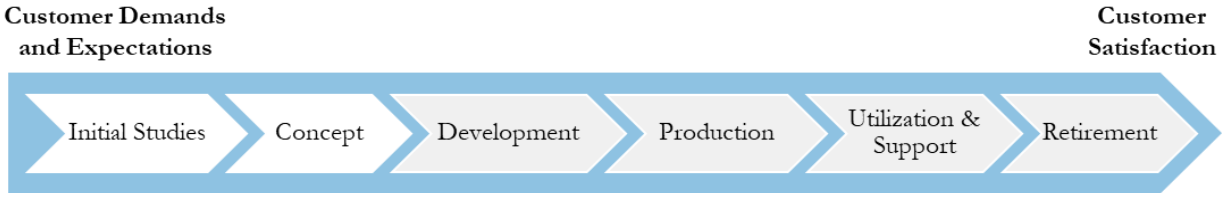

Figure 15: Overview of the product development process at Saab Aeronautics.

Before the conceptual design of the structural layout is started, some decisions on the aircraft have been made. This could be, for example:

- Preliminary design of the fuselage OML.
- Preliminary design of the air intake.
- Preliminary positioning of the landing gear, engine, fuel tank, cockpit, and other major systems.

Once these decisions have been done, the creation of the SLM can begin. This indicates that the level of detail in this stage is expected to be low. Therefore, the SLM is created with respect to the fuselage OML, air intake, and the existing major systems such as the engine, landing gear, cockpit, and some sensors.

A low detailed structural concept generally consist of the most important structural components as 2D shapes. The most important systems, such as the ones mentioned above, has a preliminary position. A higher detail level would be introducing more systems to the structure. Also, introducing more structural components such as stringers. Then later, details to the components would be introduced, such as stiffeners and holes.

4.1.2 Concept Design and Analysis - Today

To get a general idea of how Saab operates today regarding organization, software, and design process, internal documents were examined and compiled. The current method of the design and analysis loop was identified, the process found is simplified in Figure 16. Firstly, the design is made in CATIA V5 and later exported to Hypermesh where the components are redrawn, assembled, and meshed to create the GFEM. The prepared file is then exported to Nastran and structural analysis is performed. Lastly, the result from the analysis is fed back into CATIA V5 through written or oral communication between the Analysis group and Concept Group. The thesis found out that this loop contains a lot of manual work and dead links as the files are exported between multiple software.

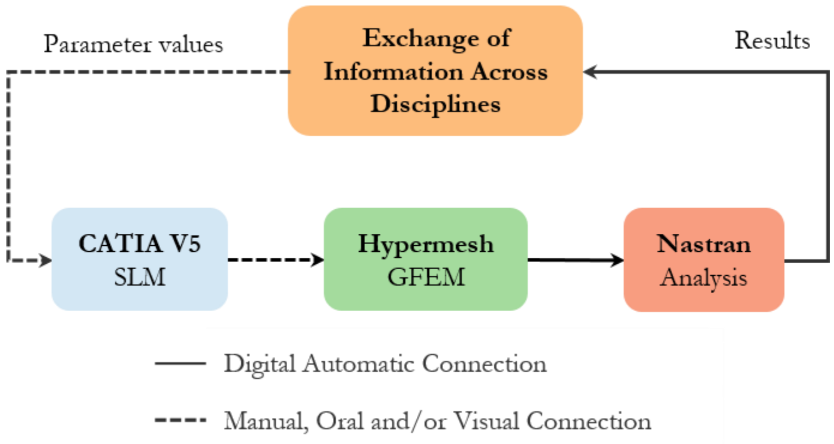

Figure 16: Design and analysis loop for the fuselage structure where the dashed lines represent manual connections.

4.1.3 Concept Design and Analysis - VUproject

After obtaining a general idea of how Saab is operating today, project specific information from the VUproject was gathered. In an early concept design phase, the aircraft structure is evaluated by performing thickness optimization on the structure during various load cases. The mass can then be calculated; thus, concepts can be compared to each other. A lower mass means a better structure. In the conceptual phase today, Saab jumps from a low fidelity aircraft model to a high-fidelity model. A need of an in-between medium fidelity model that can be integrated in an optimization framework was identified by Saab, see simplified framework for in Figure 17. A pre-requisite for an optimization framework is that all involved models and simulations needs to be fully automated to be executed.

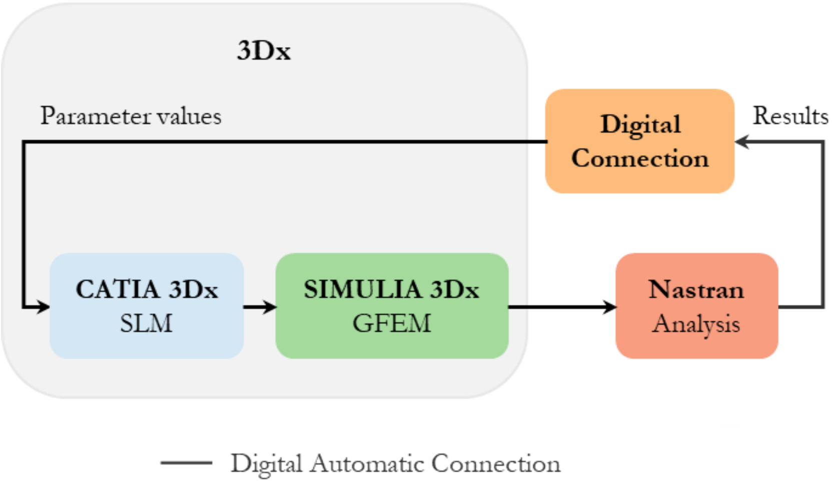

Figure 17: Framework of the VUproject's vision. All connections are automated in the loop.

Furthermore, the VUproject wants to use this opportunity to investigate the new software 3Dx, partly because CATIA V5 is reaching the end of its life span and partly because 3Dx potential of integration. The automation of the SLM, GFEM, aerodynamic models, and analysis are the one part of the area of interest for the VUproject. Another part is to investigate the integration of these models and simulations into 3Dx, specifically the SLM, GFEM, aerodynamic model, and creation of the optimization framework.

The thesis project identified the delimitation of working in the software of 3Dx with the SLM and GFEM. Requirements of these models is gathered in chapter *4.1.6 Identification of Needs* and 4.1.7 *Formal Model* divided in to the components of frames, walls and floors, and GFEM.

4.1.4 Start Model

Saab uses a so-called start model in the conceptual stage. In this model, most components in a general fighter airplane exists so that different configurations can be created, and a layout of the aircraft can be made. The purpose of the model is to enable easy manual generation of different concepts to explore the design space. This model has been used for the development of multiple aircrafts. It was created with the intent to be modified manually, therefore it is parametrized but does not include any automation. Thus, the model can undergo morphological transformations but not topological transformations without further manual work.

Through reading Saab internal documentation of the start model a simplified tree structure, see Figure 18, is extracted with information relevant to the thesis project. In the Fix Part the aircraft coordinate system is defined permanently. The model is built of skeleton models i.e. Master Data Surfaces (MDS) and Master Data Files (MDF). The MDS model contains the outer shapes of the aircraft called OML, and the air duct surfaces. Planes defining the placement of them, and all mayor components are placed in the MDF model. The structural components, such as frames, floors, and walls, are created in the SLM model based on the features of MDS, MDF and Other Systems. Other systems like engine, fuel tank, cock pit, and landing gear is dependent of all the previous mentioned models and gathered separately.

More information on the start model is gathered in a comparison table between the start model method and the developed method in chapter *5.3 Method Comparison*.

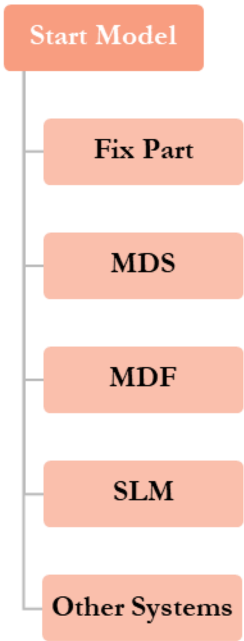

Figure 18: Simplified tree structure of the current start model. The structure is based on the hierarchy of the information, for example the global coordinate system is kept at the top in Fix Part.

4.1.5 Global FE Model

By reading Saab internal documents regarding global FE model (GFEM) and applying various semi structured interviews with persons working in the stress team, fundamental knowledge of the GFEM have been captured. The latest developed aircraft at Saab is Gripen, therefore the knowledge that have been captured is originated from the development of Gripen. The fundamentals of the GFEM are presented below.

A global FE model (GFEM) at Saab Aeronautics serves the purpose to perform structural analysis on the load carrying components in the aircraft. The GFEM analysis provides internal loads which is used to run stress analysis of detailed 3D-components. The GFEM is used to understand how the aircraft, or larger parts of the aircraft, will distribute the load from different load cases. The GFEM consists of surface elements, lines, and points, see Figure 6.Structural components such as frames, walls, floors, stringers, and the OML are represented in surface elements. System components such as radar, engine, etc. are represented in points where a mass is connected (not shown in figure). Hinge joints and other attachments between components are represented with lines and points (not shown in figure).

The thesis investigates the possibility to create the GFEM directly from the SLM, therefore, the study of the GFEM resulted in multiple needs and requirements on both the GFEM and SLM. These needs and requirements are gathered in chapter *4.1.6 Identification of Needs* and *4.1.7 Formal Model.*

4.1.6 Identification of Needs

The creation of design concept can naturally be divided into three major areas which are frame instantiation in SLM, floors and walls instantiation in SLM, and GFEM creation. Each area has their own needs in order to satisfy the thesis deliverables. Therefore, three separate needs identification tables where constructed, see Table 2 to Table 4. The identification of needs where mainly accomplished through semi-structured interviews with several people in the project. This was done to get a wide range of needs and opinions. The needs could then be appended to a table and then labelled either required or desired. Further, they were ranked from 0 – 5 where 0 is highly relevant and 5 is less relevant. The required needs got the rank 0 and the desired were ranked from 1-5 to identify which desired needs that are more important. The labelling and ranking were done by taking the interviews and the needs of the thesis into consideration. In this way, both the thesis and the company project will have beneficial results.

For frame instantiation, needs 1-7 are labelled required, see Table 2. Automation and SLM GFEM synchronization are of big interest to answer the research questions. Thus, need 1, 2, and 7 are labelled required. Need 6 are also labelled required as it were a requirement from the VUproject. Need 3-5 focuses more on the geometrical boundaries to consider when creating a frame. These are labelled required because it is mainly these geometries which governs the shape of an aircraft structure. Further, need 8-14 are labelled desired. These are considered "nice to have"-features but not crucial in order to test the principles of automation and SLM to GFEM synchronization. Needs that are common for the frame, floor and wall instantiation is that they should be automatically instantiated yet easily modifiable manually. However, in the thesis the autonomous part is considered more important. Therefore, need 10 and 12 are labelled desired. Need 8 is ranked 1 as the need were stated in multiple interviews and is also important in a programming perspective. Another need worth to mention is need 11 which came from the stress team. A frame is often preferred to align with a wing beam so that the loads generated from the wing can be distributed through the structure.

Table 2: Identified needs regarding frame instantiation in the SLM. The rank value 0 is highly relevant and 5 is less relevant.

FRAME INSTANTIATION			
No.	**Need**	**Importance**	**Rank**
1	Should be automatically instantiated	Required	0
2	Position should be modifiable	Required	0
3	Should consider engine surface	Required	0
4	Should consider fuselage skin surface	Required	0
5	Should consider air duct surface	Required	0
6	Should be included in optimization	Required	0
7	Should lay foundation for GFEM	Required	0
8	Should follow a name convention	Desired	1
9	Should update the model fast	Desired	2
10	Position should be easily modified	Desired	2
11	Should consider wing beam position	Desired	3
12	Position should be visualized clearly	Desired	3
13	Tilt should be modifiable	Desired	5
14	Should consider cockpit volume	Desired	5

The identified needs regarding floor and wall instantiation are presented in Table 3. There are some similarities when comparing the needs for instantiating frames regarding to walls and floors. Need 1, 3, 6, 8, and 10 are the same as for frames, thus the same motivation can be applied here. Although need 10 has a higher rank for walls and floors as it is more common that such components are tilted in an aircraft. Need 2 and 7 has a close connection and are labelled required. They focus on the placement on the walls and floors and that they shall move if a system is moved. Example of systems that can require closed volumes are engine, landing gear, cockpit, etc. Need 3 and 4 are also labelled required and focuses on the behaviour of walls and floors when there is a morphological or topological transformation of frames. A desired need was to automatically change the position of walls and floors.

Table 3: Identified needs regarding floor and wall instantiation in the SLM. The rank value 0 is highly relevant and 5 is less relevant.

FLOOR & WALL INSTANTIATION			
No.	**Need**	**Importance**	**Rank**
1	Should be automatically instantiated	Required	0
2	Position should follow an engineered system	Required	0
3	Should be included in optimization	Required	0
4	Should be able to handle changes of frames position	Required	0
5	Should be able to handle changes of frames presence	Required	0
6	Should lay foundation for GFEM	Required	0
7	Should create closed volumes at certain engineered systems	Required	0
8	Should follow a name convention	Desired	1
9	Position should be modifiable automatically	Desired	1
10	Tilt should be modifiable	Desired	1
11	Should update the model fast	Desired	2

The identified needs regarding the creation of GFEM is presented in Table 4. Eight general requirements were identified through interviews and through a literature study of global FE models. GFEM is a proven method for conducting structural analysis at Saab Aeronautics. Therefore, the needs on such a model are known in the company. Thus, all the needs achieved the label Required. The GFEM should be created from the SLM and therefore automatically synchronize with it. The GFEM should also be created automatically. In turn, this will allow the SLM to GFEM process to be automatically driven and thus be implemented in an optimization loop. In this thesis, the GFEM shall include the components: frame, wall, floor, and fuselage OML. This is a need which is active in the current conceptual design stage. To perform analysis on the GFEM, the geometry has to be meshed. The needs regarding thickness, material and thickness direction are required for structural analysis and allows for thickness and material optimization.

Table 4: Identified needs regarding GFEM creation. The rank value 0 is highly relevant and 5 is less relevant.

GFEM CREATION			
No.	**Need**	**Importance**	**Rank**
1	Should create GFEM automatically	Required	0
2	Should automatically synchronize with SLM	Required	0
3	Should include Frame, Wall & Floor, Fuselage OML	Required	0
4	Should be included in an optimization	Required	0
5	Should carry thickness	Required	0
6	Should carry thickness direction	Required	0
7	Should carry material	Required	0
8	Should create mesh automatically	Required	0

4.1.7 Formal Model

From the identified needs a formal model is formed to aid structure and guidance with prioritization of work in the thesis project. First the needs are translated into concrete characteristics of the models, called *Formal Entity* in the tables below. Often multiple needs share the same entity among others to be fulfilled. For example, entity 1 for Frame Instantiation: *Shall be created through script* in Table 5 is required to fulfil both the need 1: *Should be automatically instantiated* and need 6: *Should be included in optimization* from Table 2. Secondly the entities are ranked with both the thesis project and VUproject in mind. The ranking shows important focus areas regarding the Frame Instantiation such as:

- The parametric and automatic creation and modification of the SLM model.
- The creation of the frames to reference systems inside the fuselage OML. This to evaluate how the different systems can be handled in an automatic way.

Some entities turned out lower in priority as they are not necessary for answering the research questions. Two examples of this in Frame Instantiation are need 14: *Efficient script and modelling* and entity 16: *Frame tilt is parametrical driven*. Some entities as number 6 and 7: *Frame shall carry thickness direction and material* are prioritized lower. If appending the material and thickness direction automatically becomes too complicated to achieve in the thesis time frame, they can be added manually or have a standard setting.

Table 5: Formal entities together with the reference to the corresponding needs of Frame Instantiation in the SLM. The prioritization of each entity is stated in the far right column. The rank value 0 is highly relevant and 5 is less relevant.

FRAME INSTANTIATION			
No.	**Ref.**	**Formal Entity**	**Rank**
1	1, 6	Shall be created through script	1
2	2, 6, 10	Frame position is parametrical driven	1
3	6	Can add frame automatically	1
4	6	Can remove frame automatically	1
5	7	Frame shall carry thickness	1
6	7	Frame shall carry thickness direction	3
7	7	Frame shall carry material	2
8	3	Consider engine	1
9	4	Consider fuselage OML	1
10	5	Consider air duct surface	1
11	11	Consider wing beam position	4
12	14	Consider cockpit volume	1
13	6, 8	Have name convention	3
14	6, 9	Efficient script and modelling	3
15	12	Position is visualized clearly	4
16	13	Frame tilt is parametrical driven	5

The two focus areas of frame instantiation are also relevant for the walls and floors. While frames, floors, and walls shall take systems into consideration, the floors and walls must also take the frames into consideration. This because floors and walls must be connected to frames so that they do not float in space. Therefore, the formal entity 12 are of big interest as well.

Table 6: Formal entities together with the reference to the corresponding needs of Floor & Wall Instantiation in the SLM. The prioritization of each entity is stated in the far right column. The rank value 0 is highly relevant and 5 is less relevant.

FLOOR & WALL INSTANTIATION			
No.	**Ref.**	**Formal Entity**	**Rank**
1	1, 3	Shall be created through script	1
2	2, 3	Floor and wall position is parametrically connected	1
3	3, 4, 5	Can add floor and wall automatically	1
4	3, 4, 5	Can remove floor and wall automatically	1
5	6	Floor and wall shall carry thickness	1
6	6	Floor and wall shall carry material	2
7	6	Floor and wall shall carry thickness direction	3
8	2, 7	Consider engineered systems	1
9	9	Tilt is modifiable	2
10	3, 11	Efficient script and modelling	3
11	3, 8	Have name convention	4
12	4, 5	Consider morphological and topological transformation of frames	1

Regarding GFEM Creation the important areas were found to revolve around:

- The automatic creation of GFEM to gather the knowledge about how FEM representations can be made in 3Dx.
- The direct link between the SLM and GFEM. This to have up to date information and less man hours in the creation of GFEM.

The entities of number 2 and 4: *Attach material and thickness direction* is again prioritized lower with the motivation that they can be added manually or have a standard setting if the automatic process is too complicated for the thesis time frame.

Table 7: Formal entities together with the reference to the corresponding needs of GFEM Creation. The prioritization of each entity is stated in the far right column. The rank value 0 is highly relevant and 5 is less relevant.

GFEM CREATION			
No.	**Ref.**	**Formal Entity**	**Rank**
1	6	Mesh nodes shall coincide at connections	1
2	7	Attach material from CAD part	2
3	5	Attach thickness from CAD part	1
4	5	Attach thickness direction from CAD part	3
5	3	Include Walls & Floors	1
6	3	Include Frames	1
7	3	Include Fuselage OML	1
8	6, 9	Mesh shall be created through script	2
9	4, 9	GFEM shall be created through script	2
10	1	Instructions how to create GFEM manually	1
11	2	Is created with the SLM geometry	1

4.2 RAD Loop

When initializing working in the RAD loop a general procedure of first implementing all features and functions in a simple way was used. This to test the principals and after refining the results. The decision was made to use a CAD product that can hold both the 3D Shape and FEM representations. Since the focus of the thesis work was on the generation of the SLM and GFEM, no automation was considered in regard to importing references to the SLM, such as the fuselage OML geometry, air intake geometry, or other systems. References to relevant datums and surrounding systems were imported to the 3D Shape as external references manually. The starting point of the thesis work can be seen in Figure 19. A surface of a generic aircraft including OML, air intake, wings, datum planes and representative placements of some systems where present.

Figure 19: The start off point for the thesis including the fuselage OML, air intake with engine, landing gears, etc.

4.2.1 Automatic Instantiation of the Structural Layout Model

The Structural Layout Model was created in the 3Dx CATIA environment. All the component geometry was decided to be created in one 3D Shape instead of having one separate 3D shape for each instantiated component which makes the data handling complicated. In 3Dx there exist a VBA editor and KP editor to handle the two different languages. The decision was made to use the languages where they were most suited in the project, but if both could perform a task equally as well then KP would be used to keep the information as close to the model as possible.

4.2.2 Frame Instantiation

The first type of component to be instantiated in the SLM was chosen to be the frame, since the floor and wall geometries depends on them. To create the frame, it was chosen to use two separate UDFs. The first UDF, called *framePlaneUDF*, will create a plane representing the placement, and the second UDF, called *frameUDF*, will create the frame geometry.

The framePlaneUDF instantiates a plane with two parameters, *offset* and *rotationAngle*. The frame plane is defined by a distance from the *originPoint*, see Figure 20, this distance is governed by the offset parameter. The frame plane may also rotate around the y axis, the angle is controlled by the rotationAngle.

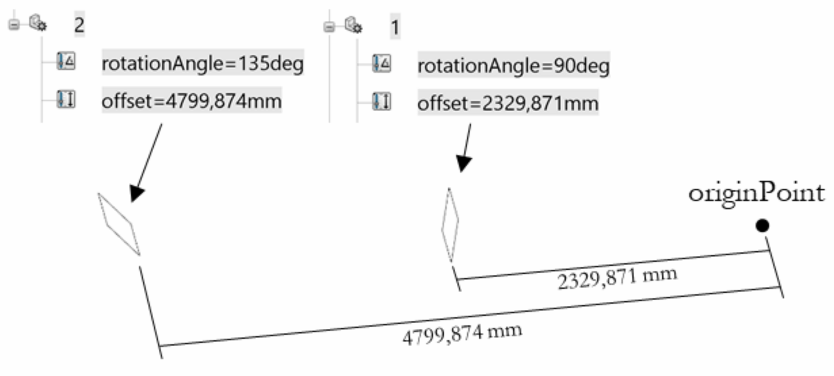

Figure 20: Two instantiated framePlaneUDFs with different values on the offset and rotationAngle parameters. The UDFs can be seen in the CAD tree named 1 and 2. The offset is measured from the originPoint.

The frameUDF instantiates the frame geometry with three parameters: thickness, thicknessDirection, and material, here on called the design attributes, see top of Figure 24. These design attributes are required to create the GFEM, requirement 5-7 in Table 5. In Figure 24 frameUDF is instantiated on the frame plane from Figure 20. According to the identified requirements 8-10 in Table 5, the boundaries of the frame geometry will be the fuselage OML, the air intake, and engine.

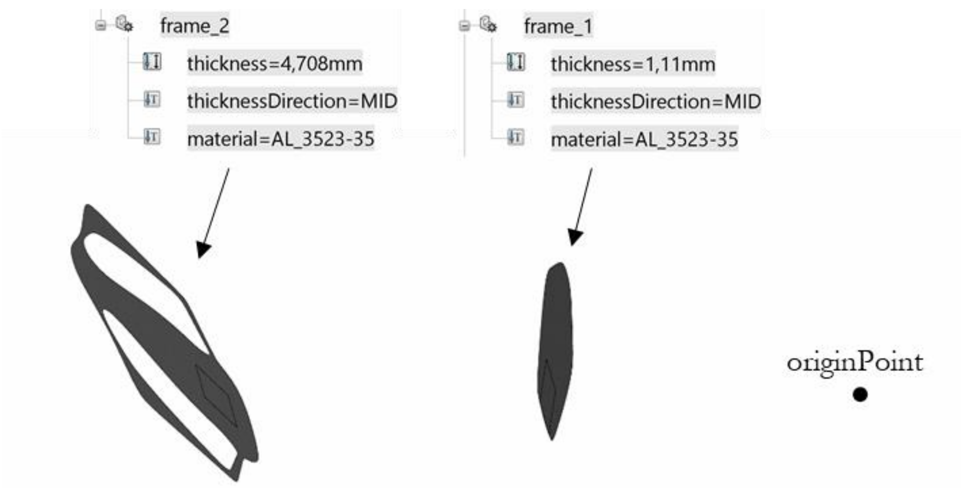

Figure 21: Two instantiated frameUDFs on the previously instantiated frame planes. The frames geometrical boundaries are the fuselage OML and air intake. The UDFs can be seen in the CAD tree named frame_1 and frame_2, and have different values on the design attributes.

The amount of frames will be equal to the amount of instantiated frame planes. To control how many frame planes which are to be instantiated, both an automatic and manual methods was developed. The manual method was developed to enable user defined positions of frames. In an aircraft, certain systems will govern the position of some frames to transmit the system loads to the fuselage, see chapter *2.1 Aircraft Structure*. Example of such systems are wing beams, cockpit, engine attachment points, etc. These systems are defined by planes in other models developed on Saab. By creating planes or importing system planes to a geometrical set in the SLM, called *framePlanesManual*, they will be used to instantiate a frame plane upon.

When the user does not know an exact position where the frame shall be positioned, it can control the amount of frames by a parameter called *frameAmount*. Thus, the total amount of frame planes which will be instantiated in the SLM are the value of the frameAmount parameter plus the amount of manually defined planes in framePlanesManual. See example in Figure 22. Each frame plane instantiated by the frameAmount parameter will get random values to the offset parameter. These can then be changed manually or automatically by script.

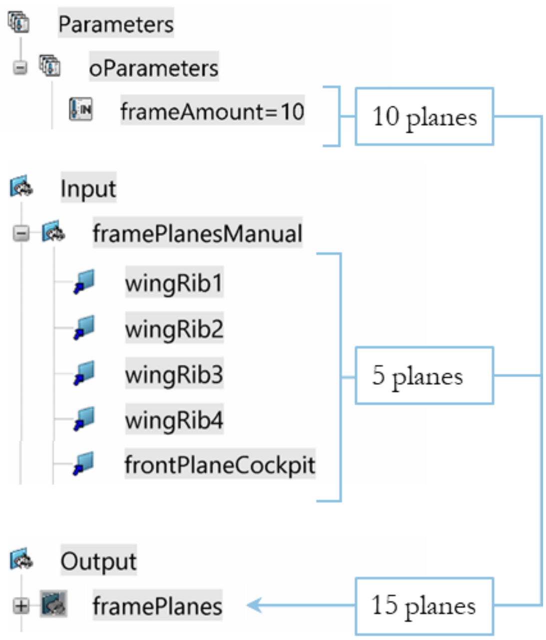

Figure 22: The parameter frameAmount and the number of planes in framePlanesManual adds up to the total number of frame planes.

This approach of separating the frame creation into the framePlaneUDF and the frameUDF was to introduce a general way to define the frame position (by offset and rotation angle) that both a script and a user can handle.

To fulfil the requirement of a naming convention, a logic that would benefit the scripting was used. The naming convention of the frame planes was a number that corresponds to the placement of it. The plane that was the closest to the nose got the name "1" and so on. The frame geometry created from plane "1" got the corresponding name "frame_1". This naming convention was chosen to allow the script to find the frame planes in their placement order instead of creation order. The placement order of the frame planes was later used in the creation of the floors and walls.

The instantiation of frame planes is handled by a KP called aFramePlanes, and the frames by aFrames.

4.2.3 Floor and Wall Instantiation

From the conducted interviews in the Capture phase of the method, useful information of the floors and walls (FWs) where found. When a concept designer creates a frame in an early concept stage, they will decide its position in x-axis (and possibly a tilt angle). The frame geometrical boundaries will then be the fuselage OML, air intake, engine, and possibly some system such as landing gear. However, when creating a FW, it may be positioned and sized in x-, y-, and z-axis within the aircraft. This means that a FW can take a great number of configurations. A major purpose of the FWs is to create closed volumes around certain systems as described before, but also to act as attachment surfaces for all smaller systems in the aircraft. The positioning of smaller systems may be different for each aircraft and placement may also change many times before the concept stage is complete. Therefore, the placement of FWs has less logic rules suitable for automation and depends more on the "know how" of the designers. Thus, a method to instantiate FWs for systems automatically using manually predefined positions where suggested and will be presented below.

Similarly, to the frame creation, FWs are also positioned by planes and contains the same design attribute parameters. Planes that define the FW's position and boundaries are manually placed in a geometric set under *FWPlanesManual*, see *top* in Figure 23. This means that each geometrical set under FWPlanesManual will result in a FW.

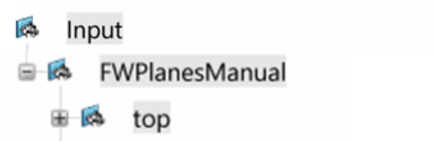

Figure 23: The geometrical set FWPlanesManual

It was found that four different UDFs could create a wide range of FW configurations, see Figure 24. The first UDF, *FWFusFusUDF,* will create a FW that extends across the whole fuselage cross section (from Fus to Fus) in the XY plane. The *FWFusPlanePosUDF* will create a FW which starts from the fuselage and ends at a plane (Fus to Plane). The difference between *Pos* and *Neg* is the direction of the wall starting from the fuselage. The *FWPlanePlaneUDF* will create a FW that starts from a plane and ends at a plane without any fuselage contact (Plane to Plane).

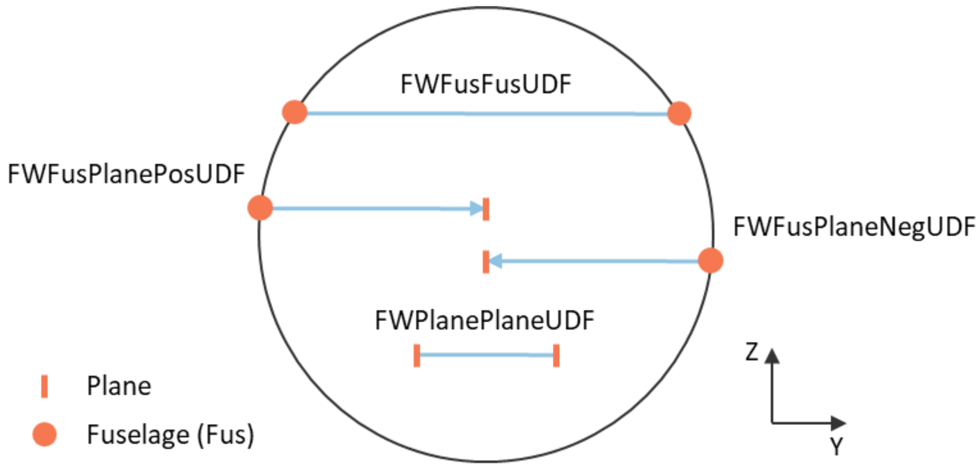

Figure 24: The four identified floor and wall configurations and their UDF.

The UDFs have three input planes in common to define the FW's position and boundaries, start and end planes in the x direction and a plane to define its position. What differentiates the four FW UDFs is the additional amount of planes needed to define it:

- **FWFusFusUDF** do not need any additional planes over the common ones as input. Resulting in three input planes.
- **FWFusPlanePos/NegUDF** needs one additional input plane to define the boundary in the Y or Z direction. Resulting in four input planes.
- **FWPlanePlaneUDF** needs two additional input planes to define the boundaries in the Y or Z direction. Resulting in in five input planes.

The KP that instantiates the FWs, called *aFWs*, will loop through all the geometrical sets under framePlanesManual and check if there are three, four, or five planes present. Depending on the number of planes the correct FW UDF will be instantiated. To decide the direction of the FWFusPlanePos/NegUDF the name geometric set must end with either *POS* or *NEG*. An example of each FW UDF is shown in Figure 25.

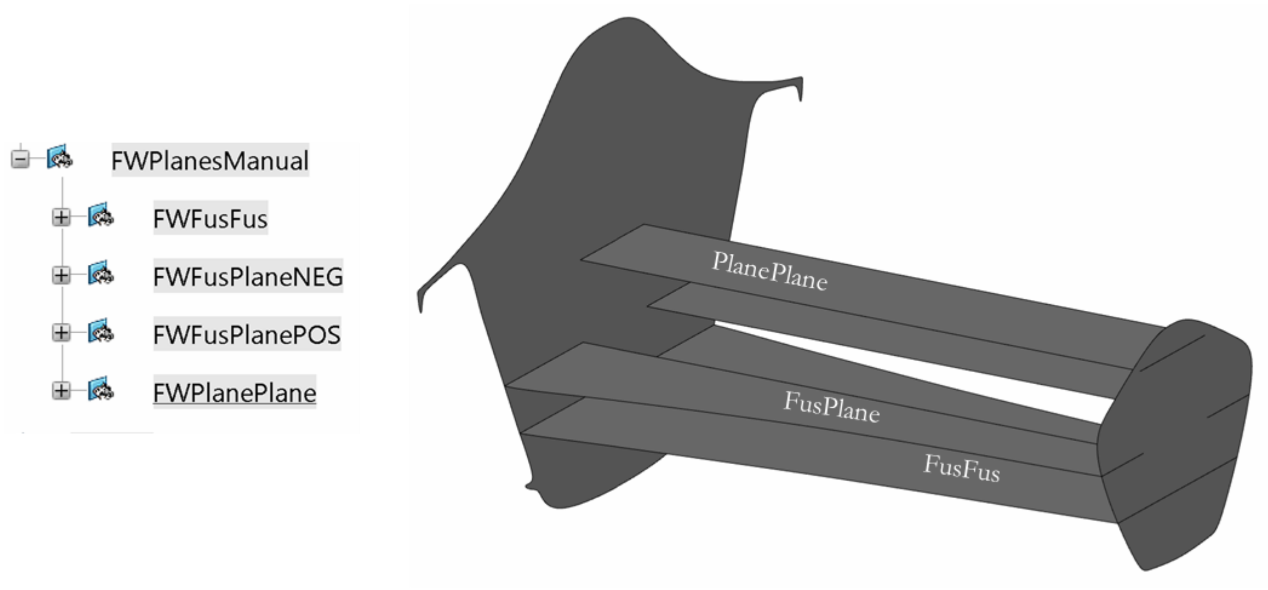

Figure 25: The FW created from the input planes in the geometrical sets shown under FWPlanesManual.

In Figure 26 *planeX1* and *planeX2* represents the start and end planes of a FW. They do not coincide with the frames, yet the FW geometry starts and ends at the frames. This is because the KP automatically finds the closest attachment point for the FW. The KP will search for the frame in front of planeX1 and the plane after planeX2. This way the KP will automatically find suitable frames to attach them on to obtain an uninterrupted load path.

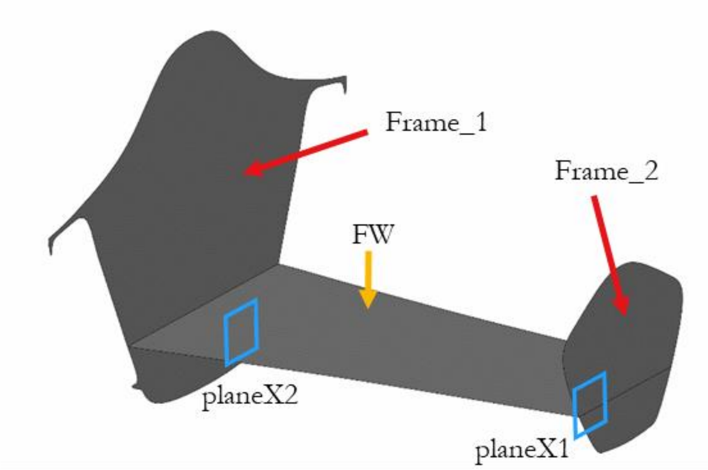

Figure 26: A floor that connects with Frame 1 and Frame 2 by having defined planeX1 and planeX2.

If a FW would reach through multiple frames, see Figure 27, it will be divided into pieces at the intersections of the frames. The script which instantiates FWs will identify relevant frame planes such as start, end and in between planes. Each FW piece will thereafter get their own design attributes. This is to give each volume in between the frames the possibility to acquire different structural properties. This search is made by knowing the position of each frame plane. The position is identified by measuring between the originPoint in front of the aircraft nose, and its projection onto the frame planes.

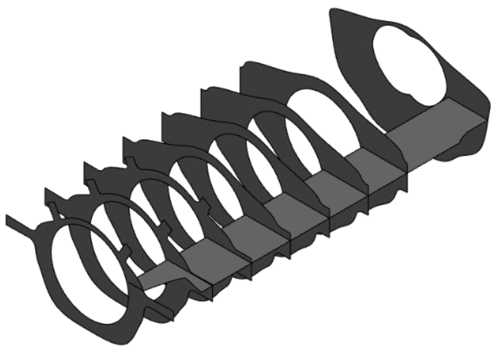

Figure 27: FW reaching through multiple frames, dividing them into multiple pieces.

The naming convention of the FWs follows the order of the geometrical sets. The first geometrical set under FWPlanesManual will create the first FW "FW_1". If the FW is divided into pieces by frames, then the piece closest to the originPoint will be named "FW_1_1" and the second "FW_1_2" etc.

Long Floor and Wall

During the Capture phase, it was identified that aircrafts can have long FWs which extends throughout the entire aircraft in the longitude direction, see an example in Figure 28. The walls are positioned vertically on both sides of the engine/engines to create closed volumes. The floors are positioned horizontally and goes from the long wall to the OML, in this case, it creates a closed volume for the landing gear. They will also distribute load between the frames in the aircraft. During interviews, it was mentioned that these walls and floors where present in today's start model and should be present in the proposed model as well.

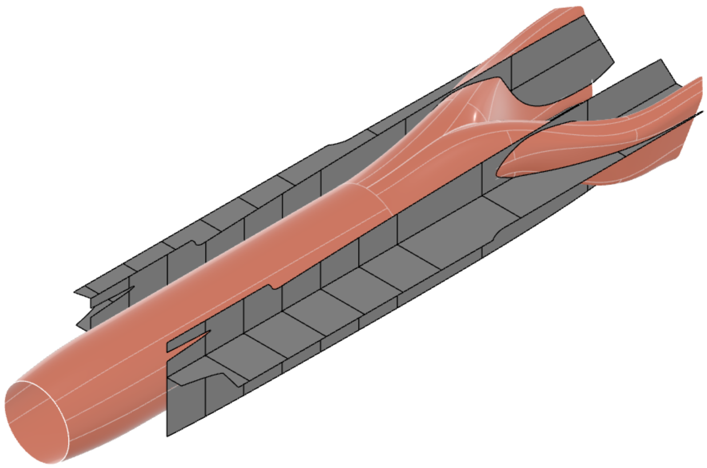

Figure 28: Two Long FWs on the left and right side if the air intake.

As the long FWs should be present, close to always, these FWs was not included in the manual FW plane method described above. The long walls were created by using the FWFusFusUDF and the walls by using the FWFusPlanePosUDF and FWFusPlaneNegUDF. The script which instantiates the long FWs will start and end at a frame plane to avoid the FW hanging in the air. The long FWs are also divided when intersecting with frames similar to the FWs explained above.

4.2.4 SLM Concepts Creation

To create a SLM, there is a process that must be followed. The process can be divided in to two categories, manual preparation, and VB macro, see Figure 29. The manual preparation has three steps, first, define the manual frame planes by creating or importing planes into the geometrical set framePlanesManual. Then define the manual FW planes by creating or importing planes into the geometrical set FWPlanesManual. Thereafter, the frameAmount parameter is set which defines the additional frame planes to be instantiated. This step may also be controlled by a script if running a DOE for example.

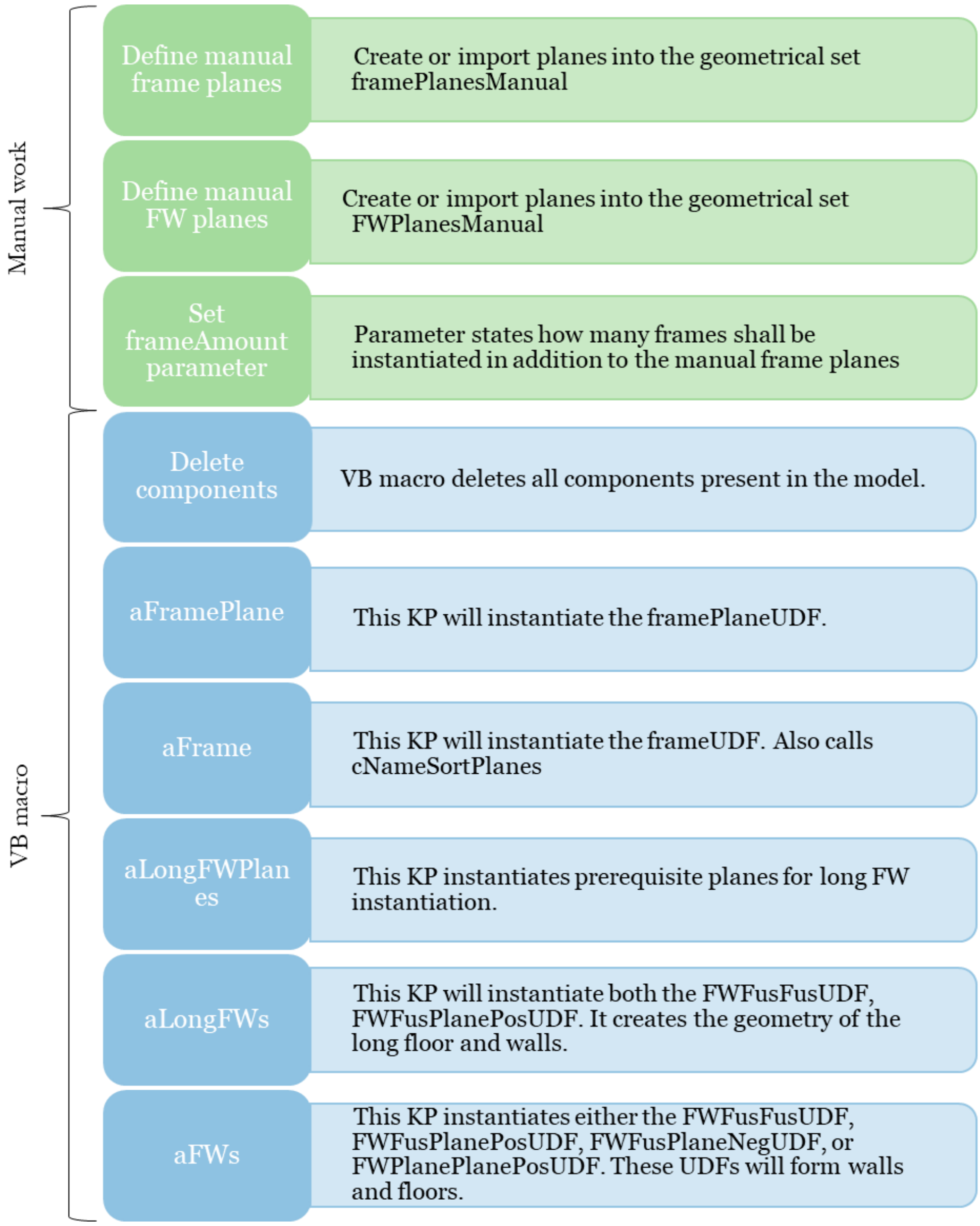

Figure 29: A flow chart of the process for creating an SLM using the developed model.

The next category is VB macro. The main VB macro will do necessary tasks and run all KPs, and VB actions used to create the SLM. The first step of the main macro is to delete all components present in the model. If the components are not deleted, it can cause errors as the modified UDFs will not update their input references. The next step is to run a KP called aFramePlane which instantiates the frame planes. Thereafter, the KP aFrame is called which instantiates the frameUDF on every frame plane. The aFrame KP also calls the VB Action cNameSortPlanes which changes the name of the frame planes depending on their position. Further, a KP called aLongFWPlanes which instantiates start and end planes that are a prerequisite for creating the long FWs. Then, the long FWs are instantiated by calling the KP called aLongFWs. And lastly, the KP called aFWs is called which instantiates the manually predefined FWs.

When a SLM is created, it can be modified in certain ways. The position of frames can be modified by changing the offset parameter for the desired frame. Adding or removing frames are done by manually instantiating frame planes or deleting the instantiated frame planes, then run the script again. The same goes for FWs, but to modify the position, the plane which defines the wall must be moved manually.

The developed method of creating a SLM concept was compared to the current method.

4.2.5 Automatic Creation of GFEM

This section will present a proposal of how to automatically create a GFEM from the previously generated SLM, and present and motivate decisions made along the way.

Overall method

The GFEM is created in the 3Dx SIMULIA environment with direct link to the SLM in the 3Dx CATIA environment. The programming language chosen for this part of the thesis where VBA. VBA supports meshing and creation of other FE model properties, and a lot of example code can be found in the 3DEXPERIENCE Automation Help file. The built in Visual Basic editor in 3Dx were used for all programming in this chapter. An advantage for using the VB editor is that the script file will be saved onto the PLM system for colleagues to use.

With the formal model of the GFEM creation in mind various attempts on creating the GFEM was first made manually to understand the process. A general manual process to prepare a single component for analysis is presented in Figure 30. A component from the SLM is meshed and its material parameter is read, the correct material is found, and applied to the component geometry. Thereafter, a property-carrying feature called Shell Section is created on the mesh. There are other property-carrying features such as beam element, however, frames, floors, walls, and fuselage OML are thin structure components, and thus shell elements are suitable. When the Shell Section is created on the same component as the material are applied on, the material is automatically set in the Shell Section. Finally, the value from the thickness and thickness direction parameters from the SLM is read and set in the Shell Section.

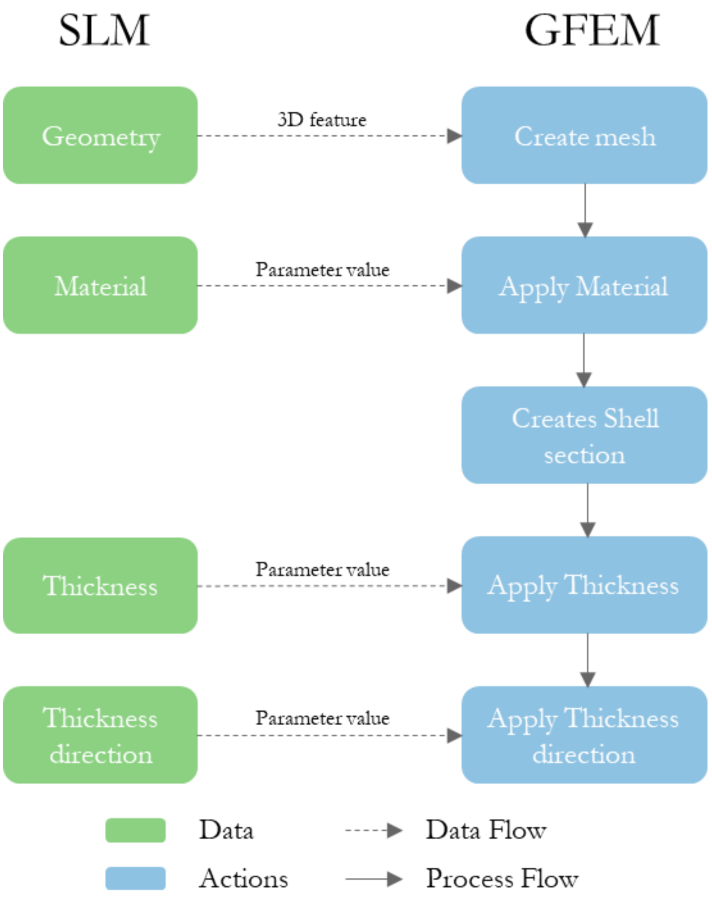

Figure 30: Flow chart of manually creating a mesh with material properties on one component from the SLM.

Meshing of the Structural Layout Geometry

One important requirement of the GFEM was "Mesh nodes shall coincide at connections", see Table 7. To connect multiple meshed components, their edge nodes must share the same node, recall from chapter *2.3.2 GFEM of an Aircraft Structure*. Two different concepts for connecting the edge nodes of each component were made in the thesis. The two concepts are presented below.

The idea of concept 1 were to mesh every component separately i.e., the number of mesh features equal the number of components. To enable nearby components to share the same mesh edge node, intersection lines between the contacting components had to be present. The lines would then be used when creating the mesh such that nodes will be attached and distributed on the lines. A feature in the mesh creation tool in 3Dx called Mesh Capture can enable this behaviour. In Figure 31, three components are meshed together. By using the proposed method, two intersection lines were created, see the highlighted lines. Each mesh must take relevant intersection lines into consideration when creating the Mesh Capture. For example, the large square must consider both intersections while the circle must only consider one intersection.

This method was proven to work manually, as seen in the Figure 31, and the functionalities used were also assisted by VBA. However, the concept required that the components, which are in contact with each other, must be identified. Then, an intersection line must be created. This method may be suitable for smaller constitutions rather than larger as the association between the components and intersections could be difficult to handle script wise. The script would have to find the intersecting components, create the intersection lines, and keep track of what lines belong to what components. A suitable feature to find intersecting surfaces was not discovered in 3Dx and automatic creation of such lines could be difficult to accomplish as they must be created in different ways depending on the geometrical situation.

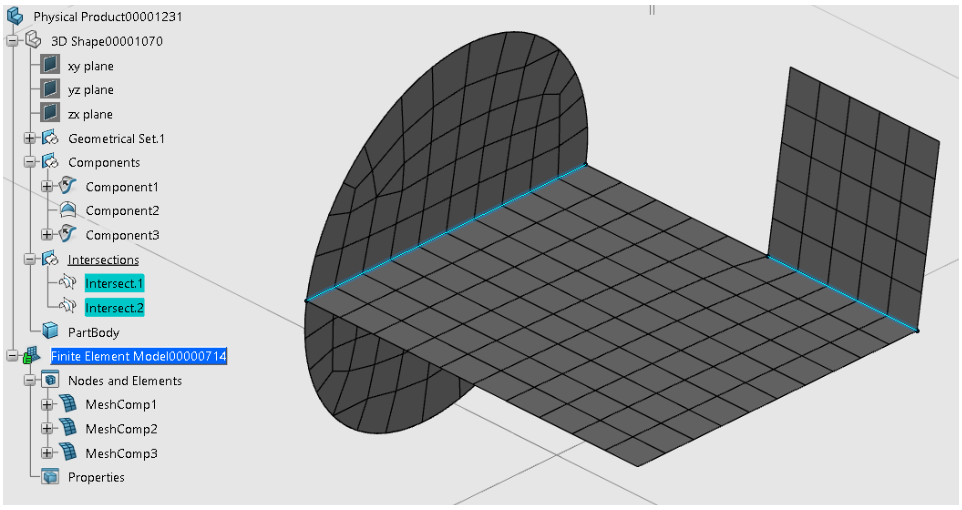

Figure 31: A model which connects mesh nodes at the intersections by meshing each component separately and attach the intersection nodes to the intersection lines.

In concept 2, the entire fuselage structure was meshed as one single component. If doing this, all components had to be integrated into one single 3D feature. To do so, a modelling function in 3Dx CATIA called Trim where used. The Trim function can be used in a certain way that multiple features are joined as one. If the Trim feature is meshed it will automatically connect the nodes at the intersections. See Figure 32 where the mesh is created by the "Trim feature" which in turn is created by all three components. In the figure we also see that the edge nodes are all connected in a correct way.

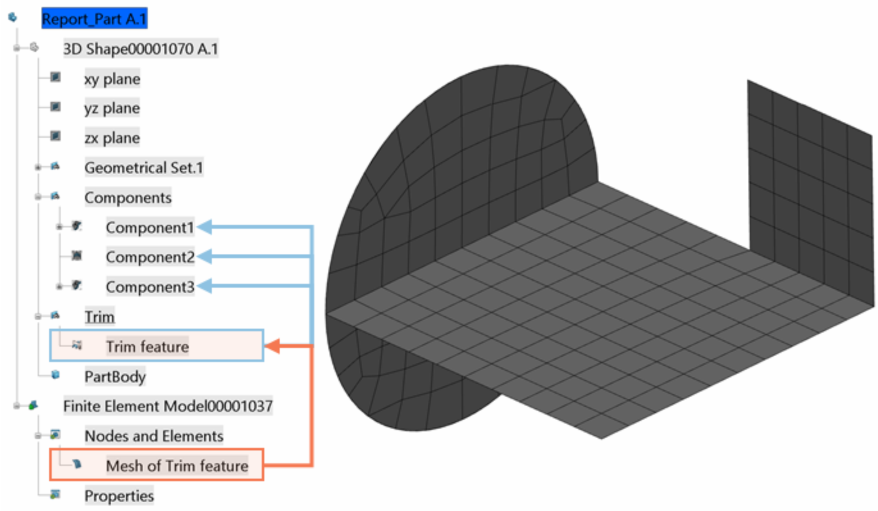

Figure 32: A model which connects mesh nodes at the intersections by meshing trim feature containing all components.

All the functionalities to achieve the proposed concept 2 are supported in VBA scripting. Concepts 1 and 2 were compared to each other. Concept 2 were considered to be easier to implement as it did not require to handle each components intersection lines. However, the mesh element size could only be of one size for the whole trimmed feature. Concepts 1 on the other hand can have unique element sizes on the different components. Having the same mesh size for all components were considered to be acceptable in an early conceptual stage after advice from experts in the topic. Although it could be desired to change the mesh element size from one component to another in a later design stage. Concept 2 were used in further development.

Appending the Design Attributes

The property-carrying feature Shell Section connects design attributes: material, thickness, thickness direction to the elements of the mesh. In concept 2, when a Shell Section is put on the "Mesh of Trim feature" in Figure 32, all the elements will get the same design attributes. This is because a Shell Section can only be applied on a mesh. To do thickness optimization in the future, the design attributes must be specific for each component.

To solve that problem the tool in 3Dx CATIA called *Proximity Group* was used. A Proximity Group identifies mesh elements within a user defined distance of a 3D feature. By creating a Proximity Group for each component 1-3, a connection is established between the components and its corresponding mesh elements. A Shell Section for each component can then be applied to the Proximity Groups to give the component their own design attributes.

After satisfying the requirement 9 and 10 in Table 7 regarding the automatic creation of GFEM, a flow chart of the script needed to generate the GFEM where created, see Figure 33. First off, all components which shall be meshed are appended to the Trim feature, which is all frames, all walls and floors and the fuselage OML. Thereafter, the material parameters are read by the script and the correct material is searched in the PLM system and stored in a list. A prerequisite is that Saab standard materials must already be created as a material and saved to the PLM system. When these preparations are complete, the FEM-representation is created which contains the SIMULIA part of 3Dx. Inside the FEM-representation the mesh of the Trim feature is created. Different mesh element type and size can be set in the script. The component specific steps are looped such as one component are handled at a time. A proximity group is created referring to the current component, thereafter a Shell Section is created on the Proximity Group. Then, the correct material from the previously created material list is applied on the Shell Section. And lastly, the thickness and thickness direction parameter of the current component are read and set in the Shell Section.

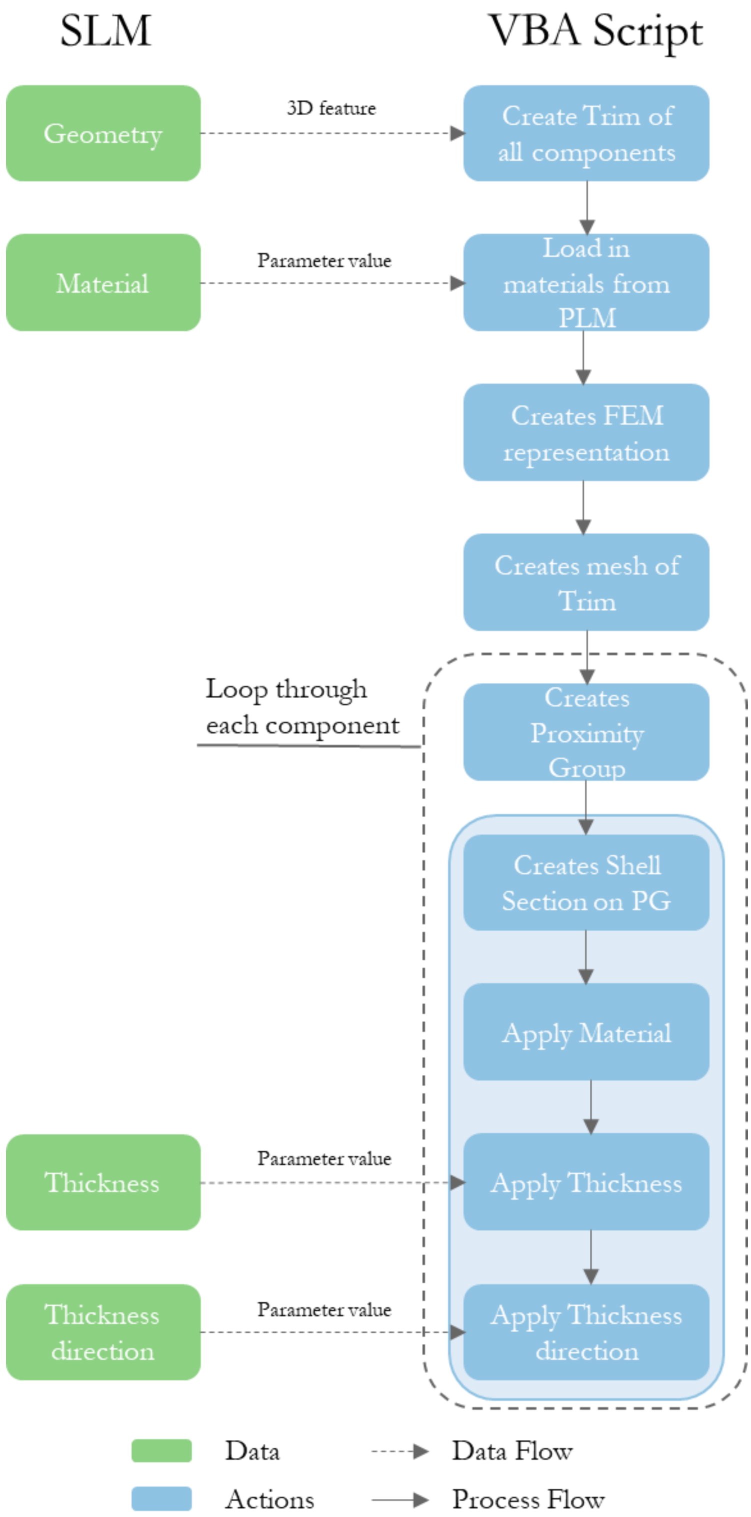

Figure 33: Flow chart of the VBA script which creates the GFEM with data from the SLM.

4.3 Activate

When the RAD loop was finished, the models created were verified. The SLM verification were done by two tests, explorative and extremity, as described below. The GFEM was manually tested by creating a GFEM from various SLM configurations, but also by modifying the mesh element type and size. The activate step also focuses on the hand over process of the developed models. Therefore a manual of the created models was written. The manual stores all model specific information.

4.3.1 Explorative test

The explorative verification test was done to identify problem areas in the design space of SLM. That is, testing the positioning of the frames (offset parameter) and the number of frames (frameAmount parameter) in the SLM. All the verification tests imported their variable parameter values from a DOE created with uniform latin hypercube sampling. The DOE was of the size *number of variable parameter x 50 runs*. To have number of runs equal to 50 for analysing the robustness of a model was deemed sufficient by [4].The offset value varied in the range from the start of the aircraft to the end. All tests also included 11 active FWs in the model. Each run the SLM was cleared and regenerated, time was logged of start and finish, and a screenshot was taken. The screenshot was then analysed to identify if any unwanted behaviours occurred. The result of the test can be seen in chapter *5.1.2 Limitations*.

Some of the verification tests had manual frame planes present from: a system in the start of the nose, cockpit area divider, and the wing beams. This means that the total amount of frames resulted in the number of manual frame planes added to the frameAmount number. Three different types of explorative verification tests where formulated:

Type A: **Constant** frameAmount parameter, manual frame planes present.

Type B: **Variable** frameAmount parameter, manual frame planes present.

Type C: **Variable** frameAmount parameter, **no** manual frame planes present.

Information regarding test data for the three test types can be seen in *0*.

4.3.2 Extremity test

The extremity test was conducted manually by testing the functionalities of all UDF's and parameters. The UDFs was instantiated manually at positions where the tester thought could be a sensitive area. Then the parameter such as frame offset and rotationAngle where given certain values that was thought to be sensitive values. Any unwanted behaviour was noted and the source to the behaviour was identified. The result of the test can be seen in chapter *5.1.2 Limitations*.

5

Result

Here the result of the execution is presented.

This thesis work resulted in a method for developing and evaluating aircraft structure concept designs with design automation. The new method includes two models with corresponding scripts.

The first model developed is a tool for a concept developer that enables the creation of aircraft fuselage structure from user defined inputs. The second model is generated by script which results in a structural analysis model with a direct connection to the SLM. The results of the method's models are presented below.

5.1 SLM

In Figure 34, the fuselage OML, air intake, engine, and wings have a shape, and the systems have a placement. This means that the Structural Layout Model is ready to be created automatically from a VBA script.

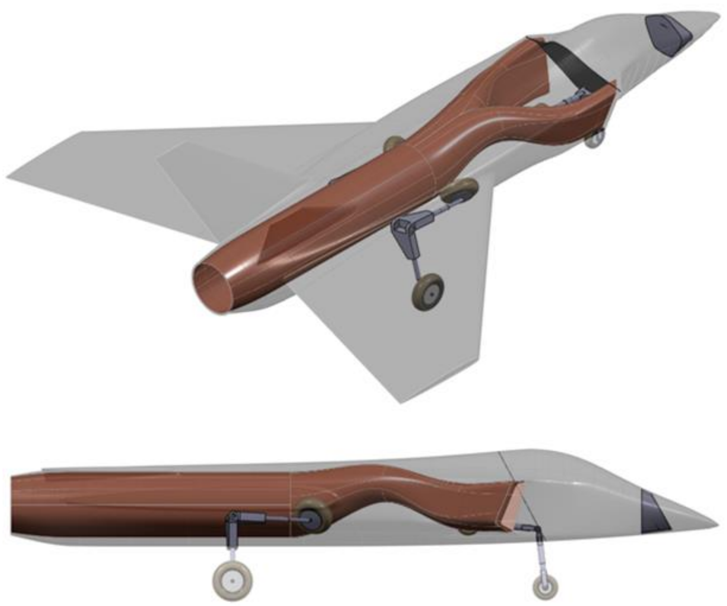

Figure 34: The start condition of the SLM where the OML, air intake, engine surface, landing gear, and radar is present.

The configuration shown in Figure 35 is a generic example of a SLM. Ten frames are made from referencing wing beams, one frame is placed on the tilting cock pit rear wall, and three frames are placed by script.

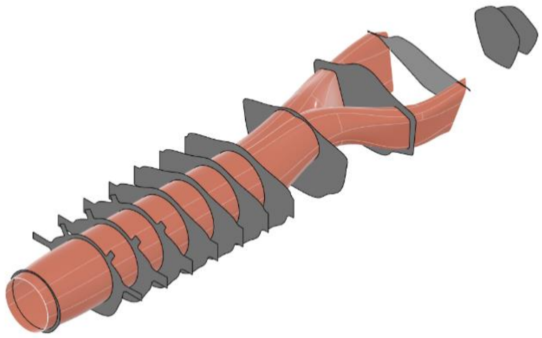

Figure 35: Result of the frame geometries after executing the script. Shown together with the air intake and engine geometry.

Two long walls are placed at the engine surface and two floors are made to encapsulate the main landing gear. To enclose the nose landing gear, three floor and walls are instantiated. Figure 36 shows these components below.

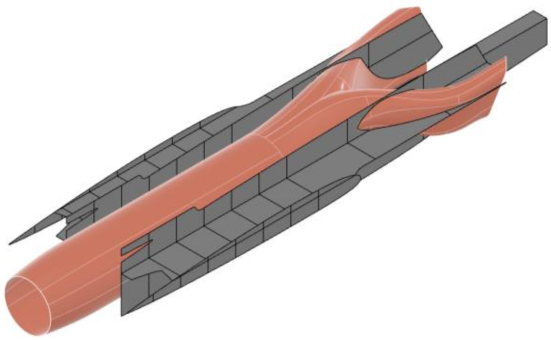

Figure 36: Result of the floor and wall geometries after executing the script. Shown together with the air intake and engine geometry.

The whole configuration can be seen in Figure 37, all the frames, floors, and walls has individually been assigned the design attributes required for creating the GFEM: thickness, thickness direction, and material.

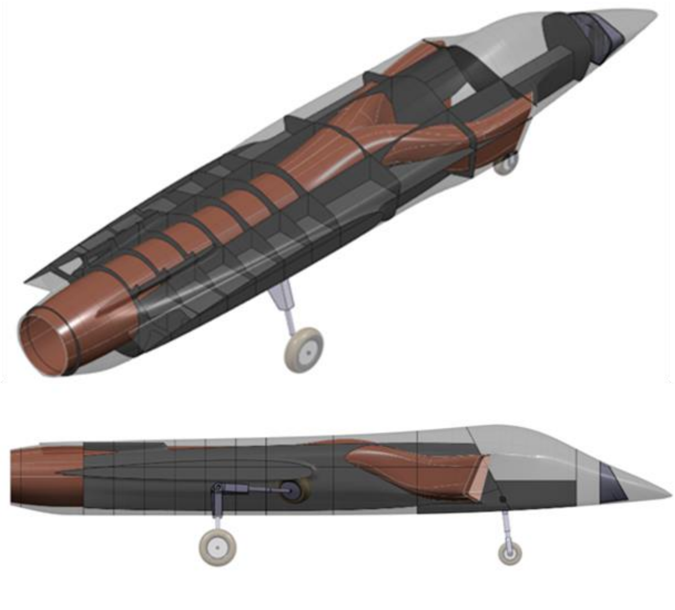

Figure 37: A configuration of the SLM where the walls, floors, and frames are shown together with the OML, air intake, engine surface, landing gear, and radar.

5.1.1 Formal Model Evaluation

In this section, the formal model is evaluated by looking at each requirement and state if they are implemented in the SLM or not.

Regarding the instantiation of frames, the formal model can be seen in Table 8 with the column stating if the requirements are implemented or not. The frames are automatically instantiated by script which adds, removes, and parametrically changes the position of frames. Further, the design attributes are linked to each component using parameters. The frame geometry considers the engine volume, fuselage OML, and air duct surface. The manual preparation of frame position allow the model to consider the wing beams positions. However, the cockpit volume was not implemented in the model as no geometry of the volume was present. Regarding the name convention, the framework allows for implementation of a desired naming convention.

Table 8: The formal model for frame instantiation which states if the requirements is implemented in the SLM.

FRAME INSTANTIATION				
No.	**Ref.**	**Formal Entity**	**Rank**	**Implemented**
1	1, 6	Shall be created through script	1	Yes
2	2, 6, 10	Frame position is parametrical driven	1	Yes
3	6	Can add frame automatically	1	Yes
4	6	Can remove frame automatically	1	Yes
5	7	Frame shall carry thickness	1	Yes
6	7	Frame shall carry thickness direction	3	Yes
7	7	Frame shall carry material	2	Yes
8	3	Consider engine	1	Yes
9	4	Consider fuselage OML	1	Yes
10	5	Consider air duct surface	1	Yes
11	11	Consider wing beam position	4	Yes
12	14	Consider cockpit volume	1	No
13	6, 8	Have name convention	3	Partly
14	6, 9	Efficient script and modelling	3	Yes
15	12	Position is visualized clearly	4	Yes
16	13	Frame tilt is parametrical driven	5	Yes

Regarding the instantiation of FW, the formal model can be seen in Table 9 with the column stating if the requirements are implemented or not. Here the FWs are created by script and also contains the design attributes similar to the frames. Positioning, adding, and removing FWs are considered partly implemented since there is manual preparation for defining and positioning the FWs, while the geometry is automatically created. The requirement which states that the FWs should consider engineered systems and that tilt should be modifiable are considered implemented as the manual preparation enables this. The FWs also consider morphological and topological transformations of frames as the script automatically searches for nearby frames.

Table 9: The formal model for frame instantiation which states if the requirements is implemented in the SLM.

FLOOR & WALL INSTANTIATION				
No.	**Ref.**	**Formal Entity**	**Rank**	**Implemented**
1	1, 3	Shall be created through script	1	Yes
2	2, 3	Floor and wall position is parametrically connected	1	Partly
3	3, 4, 5	Can add floor and wall automatically	1	Partly
4	3, 4, 5	Can remove floor and wall automatically	1	Partly
5	6	Floor and wall shall carry thickness	1	Yes
6	6	Floor and wall shall carry material	2	Yes
7	6	Floor and wall shall carry thickness direction	3	Yes
8	2, 7	Consider engineered systems	1	Yes
9	9	Tilt is modifiable	2	Yes
10	3, 11	Efficient script and modelling	3	Yes
11	3, 8	Have name convention	4	Partly
12	4, 5	Consider morphological and topological transformation of frames	1	Yes

5.1.2 Limitations

From the way the model, the KPs and the VB macros are built, it introduces some limitations to what is possible and what is not. These are known limitations, then there are unknown limitations where some of them were identified from the explorative and extremity tests. The most important limitations are explained below.

During the explorative test, it was a problem with rotated frames were noted. For example, when creating a FW, the KP searches for frames in between the start and end of the FW to divide the FW into smaller sections. But if some of the frame planes are intersecting with each other, the script does not know the correct order of them. In the example in Figure 38, a frame plane called A is rotated in such a way that it intersects with plane B. The scripts measure offsets from the origin point to a projected point on the plane, point A1 and B2 into the left in Figure 38. In the red area plane B would be considered closest to the origin point, but measured value shows that plane A is closest. This causes problems since the script will try to instantiate a FW from the first to the second plane, which depends on where you measure from. Thus, in the right side of Figure 38 where the planes are not intersecting, the order of them is the same.

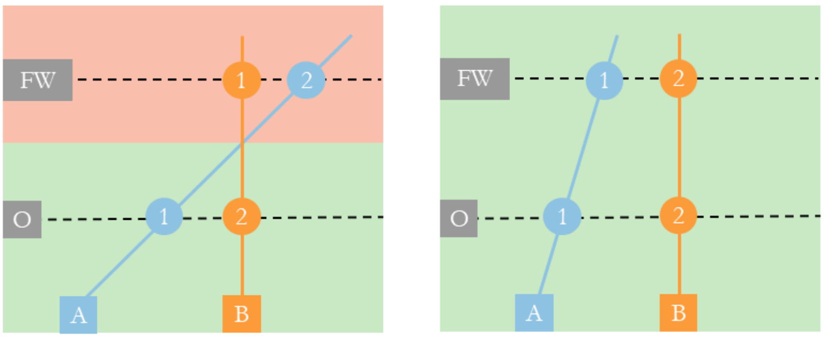

Figure 38: Problem with measuring the position order of rotated frames. In the left figure, at the line O, plane A is first. While at the line FW, plane B is first. In the right figure, plane A is always first.

Another limitation with the model is that when modifying or creating a new SLM, all components need to be deleted to instantiate the new components. This is done in the beginning of the VBA scripts. If the components are not deleted, it can cause errors as the modified UDFs will not update their input references.

There are also limitations regarding how many frames that must be present and the position on some of them. A FW must have a frame in front of the manually defined start plane and after the manually defined end plane. The long FW must have two frames in-between the start and end planes.

Further, the number of frames is known before running the script, but the final number of FWs are not known at this stage, as they will be created between frames. This could cause problems when creating DOEs as the number of components are not known.

Lastly, a limitation of the model is that frames will not be cut out when intersecting with systems other than the engine and air intake. Such systems could be cockpit or landing gear.

5.2 GFEM

When the VB macro, CreateGFEM, is run, it will generate a meshed model of the structural components from the SLM, see GFEM in Figure 39. In the figure we can see the SLM and its corresponding GFEM both with and without the OML. The GFEM is connected to the SLM which means that if the structure is modified, the GFEM will be modified as well. Each component attains the design attributes defined in the SLM.

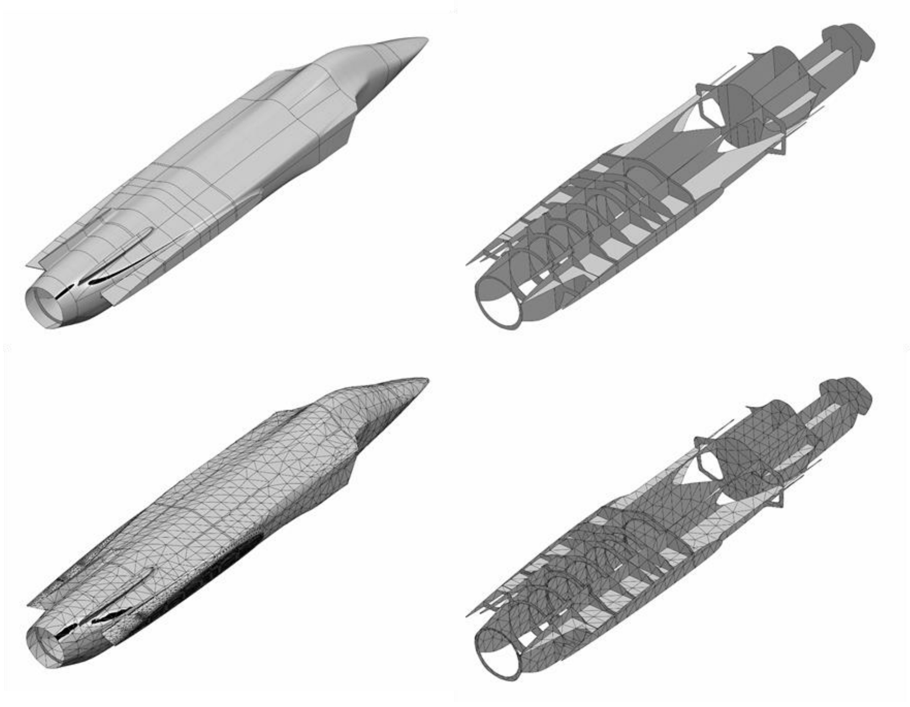

Figure 39: A comparison between the SLM and the GFEM.

To the left in Figure 40, the CAD-tree of the SLM is zoomed in on the component frame_4 with its design attribute parameters. To the right in the figure one can see the created GFEM. The Shell Section of frame_4 is highlighted, and the design attributes can be seen in the Shell Section window. Thus, requirements 2-7 and 11 in Table 7 are fulfilled.

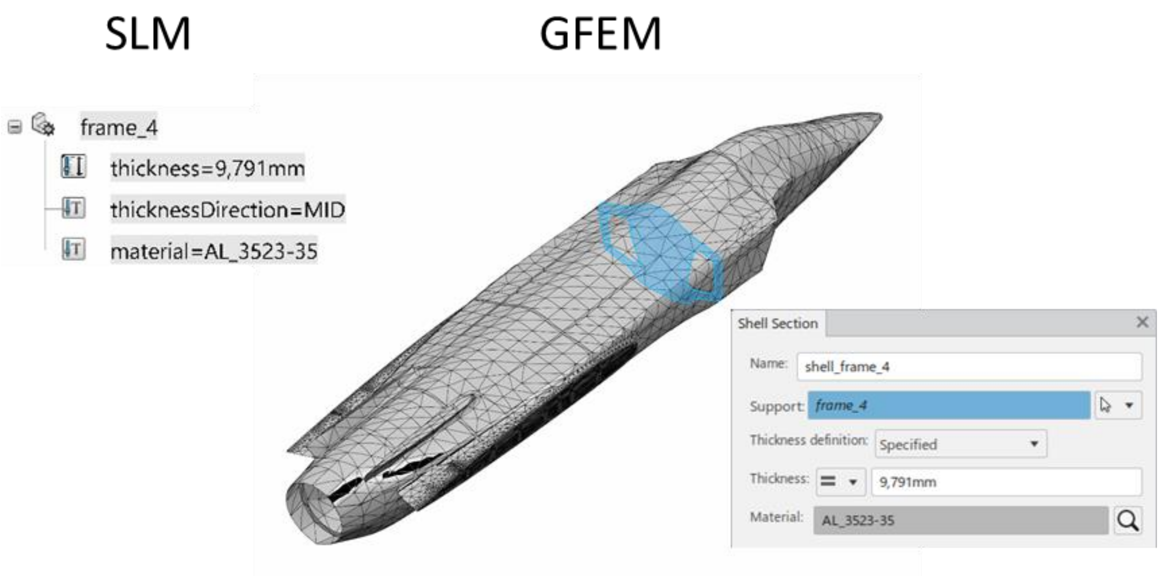

Figure 40: The result of a GFEM when the VBA script is run. The figure shows the connection between the material property parameters in the SLM and the applied material properties in the GFEM.

5.2.1 Formal Model Evaluation

In this section, the formal model is evaluated by looking at each requirement and state if they are implemented in the GFEM or not.

Regarding the GFEM creation, the formal model can be seen in Table 10 with the column stating if the requirements are implemented or not. The GFEM is automatically created through script which also meshes the components and connects the edge nodes at intersections. It is created through the geometries and design attributes of the SLM which includes frames, walls, floors, and fuselage OML. Design attributes where only partly attached to the meshed components in the GFEM, further insight on this is presented in the next chapter.

Table 10: The formal model for GFEM creation which states if the requirements is implemented in the GFEM.

GFEM CREATION				
No.	**Ref.**	**Formal Entity**	**Rank**	**Implemented**
1	6	Mesh nodes shall coincide at connections	1	Yes
2	7	Attach material from CAD part	2	Partly
3	5	Attach thickness from CAD part	1	Partly
4	5	Attach thickness direction from CAD part	3	Partly
5	3	Include Walls & Floors	1	Yes
6	3	Include Frames	1	Yes
7	3	Include Fuselage OML	1	Yes
8	6, 9	Mesh shall be created through script	2	Yes
9	4, 9	GFEM shall be created through script	2	Yes
10	1	Instructions how to create GFEM manually	1	Yes
11	2	Is created with the SLM geometry	1	Yes

5.2.2 Limitations

When creating the GFEM using one mesh and identifying the components mesh elements using Proximity Group, it showed to have a major drawback. As explained in chapter *4.2.5 Automatic Creation of GFEM,* a Proximity Group will identify mesh elements within a distance of a user defined surface. This distance is called tolerance. When meshing a sharp curved surface, the mesh elements will experience sag which is a distance between the element and the actual surface, see Figure 41. The red lines are elements of a mesh, and the black curve is a components surface. The blue dotted line will symbolise the Proximity group tolerance. As seen in the figure, the sag of one mesh element is larger than the tolerance of the Proximity Group. This will result in the element not being identified by the Proximity Group, and in turn not getting a Shell Section containing design attributes. This will disable the model to be used in analysis as all elements must have an assigned thickness and material. This is a problem in the case of the fuselage OML, since it has multiple sharp curves.

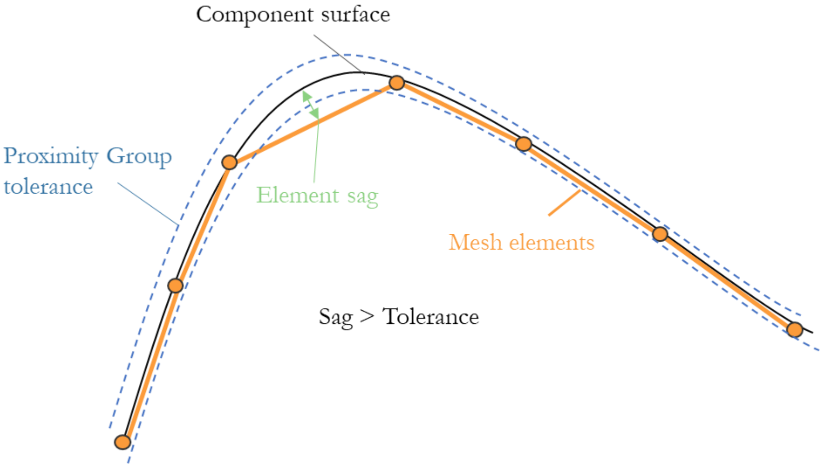

Figure 41: A schematic figure of a component surface and its mesh elements. The one element experiences a large sag which is larger than the tolerance of the Proximity Group. This result in the element not being identified by the Proximity group.

A solution could be to use a greater tolerance, but then the Proximity group would not only identify all elements on the component surface but also identify elements on neighbouring surfaces. Another solution could be to decrease the mesh element size so that they fit within the tolerance. However, neighbouring elements would also become smaller (due to one common mesh) which means that elements at intersections would be identified by both the intersecting components Proximity Groups.

That leads to the next limitation which is all the components will have the same mesh element size. This is due to meshing every component in one single mesh. Although, the study cannot exclude that local mesh specification may be done by script. This is not investigated in the thesis. Another limitation regarding the mesh is that the script may only adjust the mesh settings: mesh type, element order and element size. More settings are possible to integrate but the three available settings were considered sufficient.

Another limitation is that the script will only apply Shell Section as a property carrying feature.

There are also some limitations regarding the design attributes parameters from the SLM. Only thickness, thickness direction, and material can be applied to the GFEM as it is the data available in the SLM. The direction of the thickness may only be set from the top surface, bottom surface, or mid surface. Although, more settings can be integrated if necessary. The material parameter value must be equal to a material which has been uploaded to the PLM system in 3Dx.

From the testing of the GFEM, it was shown that the functionalities of the script worked well in most cases. However, in some cases, where the geometry is much smaller than the mesh size, the mesh may experience distorted elements in these areas. Also, the proximity group had problems identifying the desired elements when two surfaces were positioned within 10 mm from each other.

5.3 Method Comparison

A comparison between the current method and the developed method was made and can be seen in Table 11. The comparison was focused on the topics of research question 3: time, detail level, and SLM to GFEM synchronization.

Table 11: Comparison between the current and the new method. The different categories that were compared can be seen to the left. To the right, comments about the methods can be seen.

Category	Current Method	New Method
	General	
Software for design work and analysis preparation	CATIA V5, Hypermesh.	3Dx CATIA, 3Dx SIMULIA.
Naming convention	Depends on if the user follows the naming convention.	One general way that is decided in the script.
	Structural Layout Model and Script	
CAD model structure	MDF MDS from the start.	Sorted in what the model has as input and output.
Design Attributes	Depends on if the user follows convention	One general setup for every component.
Topological (can add and remove components without breaking)	Yes, but only geometries with no children.	Yes, all.
Design approach	Depends on the user's design approach.	General approach of creating geometries from inputs
Design freedom of components	Freedom in the geometric modelling.	Freedom of the general concept design. No design freedom within the logical rules creating the geometries.
Manual steps	The whole process of creating the components. From placement to defining the geometry and adding the correct design attributes.	Input of predefined frame planes. Input of FW planes.
Concept Creation	Limited by engineer's design speed.	Freedom in the number of concepts to create in a period of time.
Level of detail	The level of detail which the designer desires.	Level of detail is restricted by design automation due to complexity. It has potential.
Systems	So far just a few important systems introduced	Many introduced systems, both small and large.
	Global Finite Element Model and Script	
Settings	Individual mesh size for each component, mesh type, etc.	One mesh size of the SLM, mesh type, mesh order.
Connection with the SLM	No connection, it is lost when the SLM model is imported to Hypermesh. When the SLM changes, nothing changes with the GFEM.	Direct connection. When the SLM changes, the GFEM changes.
Manual steps	Export the SLM model, import the SLM model, choose settings of the mesh, and create mesh.	Choose settings of the GFEM mesh. Running the script for GFEM creation.
Level of detail	The level of detail which the designer desires.	Level of detail is restricted by design automation due to complexity. It has potential.

6 Discussion

In this chapter, the results and improvements of the thesis are discussed.

A method for conceptual design has been developed as well as a connection between the SLM and GFEM established. The automatic creation of the SLM in a conceptual phase have been successful in certain areas but showed to be problematic in others. The rapid generation of the new SLM will allow the conceptual designer to explore a wider design space in less amount of time than the current SLM. Although, because the complex nature of an aircraft design, there are limitations of the model which reduces the design freedom in some cases.

By using the multidisciplinary software 3Dx, the GFEM could be created with a direct connection to the SLM in an automatic way. Both the geometric data but also the design attributes was connected in a successful way. However, the method used to create the mesh showed to disable the model from its purpose. Although the level of automation integrated in the creation indicated good potential.

A comparison between the developed method and the current method was done regarding time, detail level and synchronization. The comparison indicated that there are differences induced by the implementation of design automation. The developed method is considered more time efficient when creating a SLM and GFEM. It also shows a better synchronization between the SLM and GFEM. However, the developed method shows a lower detail level compared to the current method. Mainly to the lower design freedom regarding the geometrical aspect when implementing design automation of a complex design. But also the amount of introduced systems.

The execution and result of the thesis are considered suitable to answer the research questions. As a reminder, the research questions are repeated and discussed in the following chapters.

6.1 Research Question 1

RQ1: How can an automated parametric model be created to enable structural concept design?

Design automation is most suitable for a product that follows simple design rules. In this thesis, the research was to identify a method to implement design automation on a complex aircraft structure. First off, requirements and general design rules can be identified which can be translated into code. But due to the complex design of an aircraft's structural layout, there are many rules which interfere with each other. An aircraft structure consists of compromises between disciplines everywhere. Compromises are difficult to interpret into code as they can be unpredictable, the result of a compromise can vary. This was the most difficult aspect when implement design automation to an aircraft fuselage SLM.

Even in an early conceptual design stage, the structural layout is complex with requirements from many disciplines. To implement design automation to a high extent, some limitations were made when creating the model. For example, the position of FWs cannot be modified by the script. Such limitations would constraint the design space for the structural layout which is an undesired property.

The developed method for creating the SLM have been explained in chapter *4 Execution*. A conclusion can be drawn that the SLM can be created automatically with some limitations due to the way of modelling and coding. But also due to the complexity of the aircraft design. The balance between the manual preparation and the design automation allowed for a wide range of SLM concepts to be generated rapidly. The manual preparation allowed the designers "know how" knowledge to be implemented in the SLM. These properties of the SLM creation are considered suitable in a conceptual design stage.

In chapter *1.1 Background*, the design paradox was presented, also see Figure 42. The developed method for creating the SLM have in some ways reached the goal of the design paradox in the conceptual design stage. The design freedom is kept at a high level while the design knowledge is increased due to a high design exploration. Although, the design paradox can be even more realized if limitations to the model are decreased.

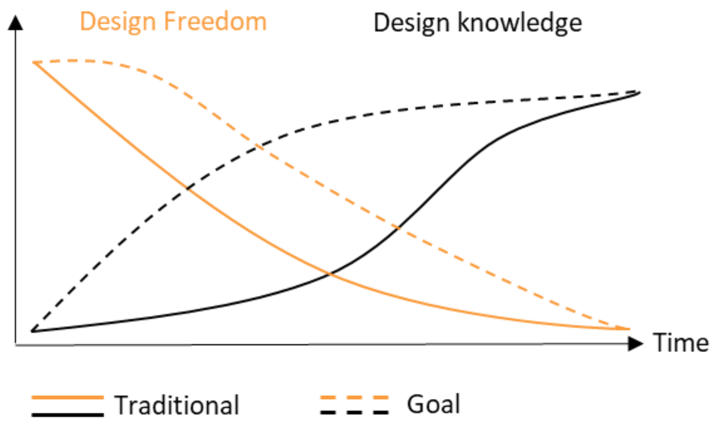

Figure 42: The design paradox.

An example of a limitation to the model which limits the design freedom is the long floors and walls. These are always active in the aircraft and may not be modified or removed if not entering the script and changing the code. This part of the model can be improved, for example applying the method for creating the other FWs with manual plane definition. This would let the user modify the SLM to a higher extent.

The decision of creating the SLM using KP, VB Action, and VB macro as programming tools had both positive and negative consequences. An improvement that could be made is to integrate all the VB macros to VB Actions in the 3D Shape to keep all scripts in the same place. KP and VB have different capabilities, and it was suitable to use both KP and VB as they could complement each other. However, instantiating UDFs using KP resulted in having to delete all components upon new instantiation due to reference issues regarding KP functionalities. One feature of KP is to handle the deletion and instantiation of UDFs. This feature partly failed in this thesis, it caused existing UDFs to experience incorrect references. However, it cannot be concluded that it does not work as no larger investigation was made on the topic.

The overall thesis method described in chapter *3 Methodology*, is considered suitable for the work. The phases integrated from MOKA gave a good structure to the thesis and resulted in a wide pre study with gathering relevant information. The RAD loop was appropriate for the creation of the models. To start with simple tasks and refining the models as knowledge is gained where a good way of working. However, when it comes to the test phase of the RAD loop, there could have been a more structured plan with clear goals regarding testing of the models. Multiple limitations of the model could have been noticed earlier and remedied if more structured tests had been conducted.

6.2 Research Question 2

RQ2: To what extent can design automation be implemented in 3DEXPERIENCE to create an automatic process between a Structural Layout Model and a Finite Element Model?

The vision of the thesis was to integrate the structural design and analysis preparation into 3Dx with direct connection and automatic synchronization between them, see Figure 43.

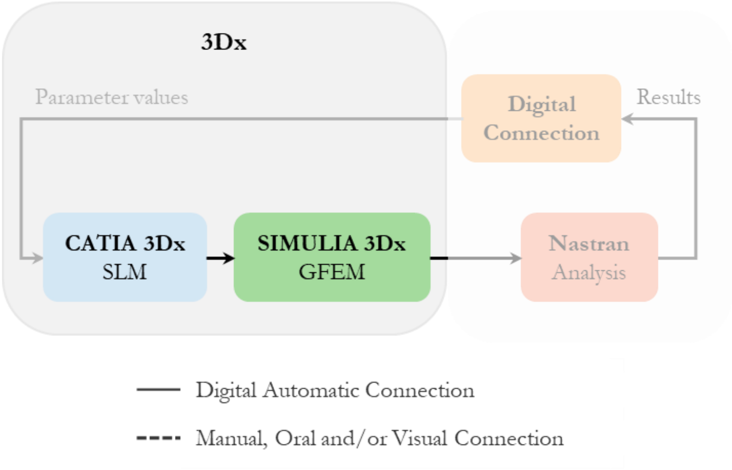

Figure 43: The VUProjects vision, the thesis focus area is highlighted.

From the execution of the thesis, it is shown that the geometric and design attributes aspect of connecting the SLM and GFEM can be and is successfully automated. The method chosen for creating the mesh in the GFEM is not working. The other identified method regarding meshing complex structures using intersection lines, see *Meshing of the Structural Layout Model* under chapter *4.2.5 Automatic Creation of GFEM,* has potential of being successful but is more time consuming to investigate and to construct the logic for. This thesis did not find a given method from 3Dx of meshing a complex assembly of components that is supported by script. If an appropriate function is developed in 3Dx and supported by script, it would be ideal for complex structure applications.

An alternative solution for creating the mesh is to keep the current software, see Figure 44, use the thesis developed SLM (and script), but automate the connections with a universal scripting language. If possible, a tool to handle the versions of the models that tells the engineers when they are working an outdated model could be used or created.

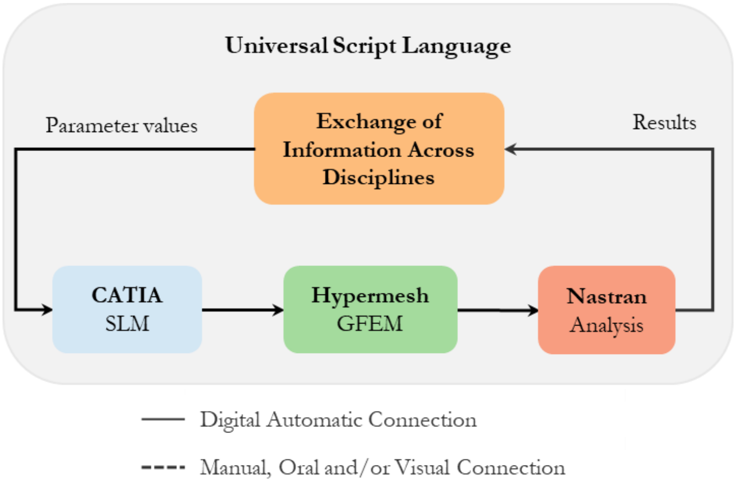

Figure 44: Alternative solution using a combination of the current software and the method developed in the thesis.

6.3 Research Question 3

RQ3: How does the developed method compare to the current method in terms of time for design creation, model detail level and, synchronization between a Structural Layout Model and a Finite Element Model?

The developed method compared to the current method showed differences in the topics of time, detail level and synchronization. It is mainly design automation that induce these differences. The three topics will be discussed below.

Time for Design Creation
Regarding time needed to create both SLM and GFEM, the new method is considered to be less time consuming. The time it takes for a script to perform the repetitive tasks in this thesis is less than if a human would do the same tasks. That is the obvious difference, now if the method requires many preparation tasks before executing the script, the comparison would be different. The developed method also reduces the number of repetitive tasks for the designer, such as floor and wall creation. The method also reduces the number of manual tasks, this is favourable in time. Also, the current method uses skeleton models such as MDF, and MDS to create the SLM, while the developed method does not. It can save time for the script to have all relevant information gathered in one model, but this can reduce the understanding of the overall information hierarchy.

Detail Level of Overall Aircraft Design
The developed method may be less time consuming, but the cost can be that the aircraft contains a lower level of detail. As discussed before, the complexity of an aircraft induced challenges when integrating design automation. The more detail in a design, the more complex it gets. Therefore, to test the implementation of design automation in the creation of SLM and GFEM, the level of detail had to be reduced. Thus, the number of introduced systems is less in the developed models. The lower detail level is also a consequence of the lower geometrical design freedom in the developed method. While creating the models manually, the level of detail is not a problem. Although, no attempt on implementing a higher level of detail was made in the thesis, therefore, the discussion above is mainly speculations based on experience.

Synchronization
A part of the thesis goal where to improve the synchronization of the SLM and GFEM. The new method is considered to do so. As the result shows, the geometry from the SLM is used directly in the GFEM with the design attributes from each component. All the data is stored in the SLM and are used to create the GFEM. Therefore, the SLM is the controlling model and decides the outcome of the GFEM. It has also shown that if the SLM geometry changes, the GFEM is also changed. 3Dx allowed the models to be in the same software compared to the current method where two separate software are used to create the models. Thus, the automatic synchronization is lost in the current method. Although, as discussed above, the new method encountered difficulties in meshing the geometries and therefore is not a functional GFEM.

6.4 Improvement Areas

Possible improvement areas that exist naturally from the thesis limitations and that were discovered throughout the thesis work will be presented below.

Information
The information implemented in the SLM are simple geometries, thickness, thickness direction, and material. Further information that is required for more accurate analysis of the structure are the couplings, degrees of freedom, weight, and inertia of the different components. This information would also have to be integrated in the GFEM by script.

The configuration of this information was not considered in the thesis due to delimitations. However, the overall approach of presenting and storing information in the model can be used for this application as well.

Optimization
The thesis assumed that the system placements were given and the information of their placement where already imported to the SLM. To carry out an optimization, an automatic connection between the SLM and relevant system models need to exist. The SLM is configured to allow the future implementation of copying reference system planes into specific input geometrical sets in the SLM.

To avoid the problem with the measuring of rotated frames, a new method for identifying a frames position must be developed.

Detail Level
The three types of components that were deemed the most important in the early concept phase were chosen to be created in this thesis: frames, floors, and walls. Next, stringers can be added with similar method.

The geometries created in the model is simple representations of the final design of the component. Simplifications were made, and all components created in the thesis needs to be updated to be able to be cut out by each other. More systems than the fuselage OML, air intake, and engine should be considered by the components geometries.

The components geometry detail level is low and can be further developed in the model. The details that are important for the decision making in the early concept stages can be implemented in the SLM.

Re-Evaluation
The issue of meshing a complex structure in 3Dx needs to be addressed. An evaluation needs to be carried out regarding if it is worth spending the time either to get the available tools in 3Dx complying to the method or if the resources should be spent on finding other meshing solutions outside of 3Dx.

7

Conclusion

From the results and discussion, conclusions are drawn and presented in this chapter.

The goal of the thesis was to investigate how design automation could be implemented in an early conceptual design phase regarding the creation of a SLM. Also, to investigate a more synchronized connection between disciplines regarding the SLM and GFEM.

Implementation of Design Automation
Implementation of design automation in the creation process of the SLM has been achieved. Instantiating components using UDFs together with mixture of KP and VB macros was found to be a suitable approach. Although, implementing design rules and compromises in form of code showed to be complicated, as the detail level of the aircraft gets higher and higher. Therefore, the suitable stage for design automation on a large scale is when the detail level is kept low. However, manual preparation steps for creating the SLM was introduced which allowed implementation of design automation in a more detailed design.

The Design Paradox
It was found important to identify what information that is necessary to make design decisions in this early concept phase. This way it can be derived what information and characteristics the models shall take. The thesis also found, in line with the design paradox, that the models should hold the possibility to efficiently be changed as long as possible in the concept phase in order to save money in time.

Global Finite Element Model
Design automation was found to automate the process from creating the SLM to importing geometric references and information into the GFEM successfully. However, difficulties were discovered when meshing a large structure using design automation in 3Dx, which disabled the model from its purpose. The thesis argued that 3Dx might not be the suitable software for such actions, although, it cannot be completely rejected as there might be more functions to be investigated.

The Developed Method
The developed method enables a faster iteration work of fuselage structural concept designs compared to the current method. The detail level is lower but more consistent and uniform. By using 3Dx, the synchronization between the SLM and GFEM could be made automatic. However, the GFEM was not fulfilling its purpose in the developed method.

7.1 Future Research

The thesis opened doors for further research on the topic of design automation. Below are some examples of such research.

- Investigate other methods of meshing complex structures, inside and outside of 3Dx which enables a synchronization between the SLM and GFEM.
- Explore other ways of referencing structural components to each other, regarding CAD geometry and automation.
- Investigate other model configurations other than a SLM and GFEM.

7.2 The Work in a Wider Context

The thesis work has contributed to the scientific topic of design automation in the field of structural aircraft design and structural analysis. This method can be applied on any larger scale project. The investigation of automatic creation of a SLM and GFEM has provided a foundation for implementing design automation in these models. It also provided a base for further research on the topic. The thesis also provided suggestions on how to implement design automation in 3Dx.

This thesis challenged the traditional way of working in a conceptual design stage by exploring possibilities and limitations of a more autonomous approach.

References

[1] M. Tarkian, “Design Reuse and Automation: On High Level CAD Modeling for Multidisciplinary Design and Optimization,” Linköpings universitet, Linköping, 2009.

[2] N. Hall, “Airplane Parts and Function,” NASA, [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/airplane.html. [Accessed 05 05 2022].

[3] T. H. G. Megson, Aircraft Structures for engineering students, Oxford: Butterworth-Heinemann, 2013.

[4] K. Amadori, M. Tarkian, J. Ölvander and P. Krus, “Flexible and robust CAD models for design automation,” *Advanced Engineering Informatics,* vol. 26, no. 2, pp. 180-195, 2012.

[5] G. R. Liu and S. S. Quek, The Finite Element Method : A Practical Course, Amsterdam: Elsevier Science & Technology, 2014.

[6] A. Dharmasaroja, C. G. Armstrong, A. Murphy, T. T. Robinson, S. H. McGuinness, N. L. Iorga and J. R. Barron, “Load Case Characterization for the Aircraft Structural Design Process,” *American Institute of Aeronautics and Astronautics,* vol. 55, no. 8, 2017.

[7] Dassault Systémes, “3DEXPERIENCE® 3D Software Products - Dassault Systèmes®,” Dassault Systémes, 2022. [Online]. Available: https://www.3ds.com/products-services/. [Accessed 16 Febuary 2022].

[8] Dassault Systémes, “The 3DEXPERIENCE Platform: Sustainable, Collaborative Innovation,” 2021. [Online]. Available: https://www.3ds.com/sites/default/files/2021-11/3dexperience-ebook-final.pdf. [Accessed 16 Febuary 2022].

[9] Technia Addnode Group, “CATIA V5, V6, 3DEXPERIENCE... is it all still CATIA? | TECHNIA (US),” Technia Addnode Group, 2021. [Online]. Available: https://www.technia.us/blog/CATIA-v5-v6-3dexperience-is-it-all-still-CATIA/#section-1. [Accessed 16 Febuary 2022].

[10] Dassault Systèmes, “About the Knowledge Pattern Feature,” 2009. [Online]. Available: http://CATIAdoc.free.fr/online/pktug_C2/pktugat0019.htm. [Accessed 21 04 2022].

[11] Dassault Systèmes, “Automating Repetitive Tasks Using Macros,” 2009. [Online]. Available: http://CATIAdoc.free.fr/online/basug_C2/basugbt1901.htm. [Accessed 21 04 2022].

[12] Dassault Systèmes, “SIMULIA: Disciplines,” [Online]. Available: https://www.3ds.com/products-services/simulia/disciplines/. [Accessed 13 04 2022].

[13] R. Curran, W. J. Verhagen, M. J. van Tooren and T. H. van der Laan, “A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD,” *Expert Systems with Applications,* pp. 7336-7350, 2010.

[14] M. Sandberg, “Towards a Knowledge-Based Engineering Methodology for Construction,” in *2015 International Conference on Construction and Real Estate Management*, Luleå, 2015.

[15] M. Stokes, Managing Engineering Knowledge - Moka Methodology For Knowledge Based Engineering Applications, London : Professional Engineering Publishing, 2001.

[16] H. Berger, P. Beynon-Davies and P. Cleary, “The Utility of a Rapid Application Development (RAD) Approach For a Large Complex Information Systems Development,” in *13th European Conference on Information Systems*, Turku, 2004.

[17] “Creatio.com,” Creatio, 2022. [Online]. Available: https://www.creatio.com/page/rapid-application-development. [Accessed 8 Febuary 2022].

[18] W. Adams, “Conducting Semi-Structured Interviews,” in *Handbook of Practical Program Evaluation Edition: 4*, Washington, Jossey-Bass, 2015, pp. 492-505.

[19] K. T. Ulrich and S. D. Eppinger, Product design and development, New York: McGraw-Hill, 2011.

Appendix

Explorative Tests of Type A

Four different tests were conducted were the frameAmount parameter was held constant. The value for test number, *i,* can be seen in Table 12. For these tests the total amount of frames equalled to manual frame planes added to the frame amount number. The model was regenerated 50 times with variable offset values, see Table 12.

Table 12: Parameter values of explorative tests of type A where frameAmount is constant and manual frame planes are present.

PARAMETER NAME	TYPE	RANGE	UNIT	
frameAmount	Constant	0 to i	[-]	where i = 11, 22, 33, or 44
offset_1	Variable	94 to 13000	[mm]	
offset_2	Variable	94 to 13000	[mm]	
…				
offset_i	Variable	94 to 13000	[mm]	

Explorative Tests of Type B and C

Two tests were also made with letting the frameAmount parameter to vary. Both tests used the same DOE where the frameAmount parameter took values from 0 to 22, see Table 13. The first test had manual frame planes present. The second test was made without any manual frame planes present, which means the total amount of frame planes could range from 0 to 22. To handle the varying number of offset values needed, the DOE was made with 22 offset values for each run. This entails that when frameAmount was for example of the value 4, the script would take the first four offset values (of 22) to generate frame planes.

Table 13: Parameter values of explorative tests of type B and C where frameAmount varies and no manual frame planes are present.

PARAMETER NAME	TYPE	RANGE	UNIT
frameAmount	Variable	0 to 22	[-]
offset_1	Variable	94 to 13000	[mm]
offset_2	Variable	94 to 13000	[mm]
…			
offset_22	Variable	94 to 13000	[mm]