

Automating Inclusion of Production Considerations in the Conceptual Design of Aircraft Structures

Darpan Bansal

Technische Universiteit Delft

Automating Inclusion of Production Considerations in the Conceptual Design of Aircraft Structures

by

Darpan Bansal

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Friday October 14, 2022 at 09:30 AM.

Student number:	5024900	
Project duration:	October 1, 2021 – October 14, 2022	
Thesis committee:	Dr. ir. G. La Rocca,	TU Delft, committee chair & supervisor
	Dr. ir. O. K. Bergsma,	TU Delft, examiner
	Dr. F. Oliviero,	TU Delft, examiner
	Ir. A. M. R. M. Bruggeman,	TU Delft, supervisor
	Dr. ir. T. v.d. Berg,	GKN Fokker Aerostructures, supervisor

This thesis is confidential and cannot be made public until October 14, 2027.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Cover photo: <https://imgur.com/us0T46L> (accessed 04 October 2022)

Preface

With this report, the final piece of my two-year journey to become an aerospace engineer is complete. I have learned more than I thought was possible in these two short years at the Faculty of Aerospace Engineering, Delft University of Technology, and at GKN Fokker Aerostructures, where I completed my internship and this thesis.

I would first like to thank Ton van der Laan and Tobie van den Berg from the Centre of Competence - Design department at GKN Fokker Aerostructures, for giving me the opportunity to work on this interesting and challenging thesis topic, and for their extremely helpful inputs at every step of this thesis. I would then like to thank Anne-Liza Bruggeman, for her constant availability, and detailed feedback that helped in maintaining the scientific relevance of this work. Lastly, I would like to thank Gianfranco La Rocca for helping me keep track of the bigger picture, and for his helpful inputs at all critical stages of this thesis.

Finally, I would like to express my gratitude towards my family and friends, who constantly supported me, and also helped in taking my mind off of work from time to time.

*Darpan Bansal
Delft, October 2022*

Summary

Considerations that decide producibility of a design are an important part of the design process, and must be included in the early design stages to ensure that these designs can be realised. Not including these considerations carries the risk of incurring additional costs and delays at later stages of product development because of design changes, or can lead to limiting oneself to conservative design choices to reduce the associated risk.

The current process in the industry accounts for these production considerations in design through a manual process that is iterative and time-consuming, and hence forms a bottleneck in being able to trade-off multiple design concepts. Attempts at accounting for these production considerations in an automated way are associated with the limitations of either only considering the manufacturing cost, being specific solutions that work only in certain scenarios, or being dependent on some commercial software tools, which are not fully suitable for use in context of automation and/or at the conceptual design stage. Additionally, the aspects of manufacturing and assembly are usually not considered at the same time in these studies.

Therefore, this thesis aims at developing a methodology that enables the automated inclusion of production considerations in the conceptual design process of aircraft structures, while overcoming shortcomings of the state-of-the-art.

This has been accomplished by leveraging model-based systems engineering to develop the Manufacturing Information Model, which provides a model-based approach for managing all production information of a product system. The model consists of three subpackages, namely, the manufacturing model, database, and the assembly model. The manufacturing model's main function is to capture production related information for every component in a product. The database subpackage supports the manufacturing model by providing it with detailed information on manufacturing methods, materials, equipment and manufacturing sites. Finally, the assembly model makes use of the information captured by the manufacturing model to create feasible assembly sequences that include all part manufacturing and joining operations in the product.

The manufacturing information model was implemented in ParaPy, a knowledge-based engineering platform, and its implementation was verified and validated by applying it to simplified use-case, whose product was composed of a limited number of parts and joints. The implemented system was linked to various analysis tools, such as cost, mass and compatibility, and also integrated with a model-based requirement verification framework to manage requirements related to both design and manufacturing. With these additional tools, the manufacturing information model was able to account for four categories of production considerations: joint information, compatibility of manufacturing and design choices, assembly sequence planning, and integration of functional parts.

The verification and validation case demonstrated the main functionalities of each subpackage of the manufacturing information model. To test its applicability to real world cases, the model was applied to an industrial use-case for the conceptual design of a wingbox at GKN Fokker Aerostructures. A design of experiments study was conducted with design variables from the manufacturing model and assembly model. The results from this study allowed for the identification of trends in the design space of different manufacturing concepts, and to rank these concepts based on the imposed requirements, which helped in the decision-making process. Some unapparent manufacturing concepts could also be identified from these results.

Additionally, the effect of including production considerations on gauging a product's viability was analysed. The results indicate that joints and part integration have a large impact on the total mass and total cost of the product, whereas, assembly sequence planning and compatibility analysis offer a reduction of manual iterative steps in the design process.

Finally, the main recommendations for improving the manufacturing information model with future work include, accounting for the effect of assembly sequence planning on other parameters in the product model (such as the non-recurring cost & number of workers required for production), and to fully automate the identification of feasible assembly sequences. The current implementation uses a

limited set of rules for the latter, which when expanded, would help eliminate any manual intervention that is required for checking assembly sequence feasibility.

Contents

List of Figures	ix
List of Tables	xiii
List of Abbreviations	xv
1 Introduction	1
1.1 Thesis Scope	3
1.2 Thesis Outline	4
2 State of the Art and Background Information	7
2.1 Overview of production considerations for product design	7
2.2 Manual approach.	9
2.3 Automated approaches	12
2.3.1 Accounting for production considerations only through the associated cost	12
2.3.2 Non-generic solutions	15
2.3.3 Use of some specialised commercial tools	17
2.4 Assembly sequence representation	20
2.4.1 AND/OR graph	20
2.4.2 Directed graph	21
2.4.3 Liaison diagram.	21
2.4.4 Precedence diagram	22
3 The Manufacturing Information Model	23
3.1 Manufacturing Model	23
3.2 Database.	29
3.2.1 Data directory.	29
3.2.2 Libraries subpackage	31
3.3 Assembly Model.	33
3.3.1 The operation graph	34
3.3.2 The station graph.	35
3.3.3 Integration of assembly model with a product.	36
4 Implementation of the Manufacturing Information Model	41
4.1 Software for implementing MIM	41
4.2 Integration of MIM with a product model.	42
4.3 Supporting models and analysis tools used in this thesis	43
4.3.1 MDM: Multidisciplinary Modellers.	43
4.3.2 CATMAC: Cost Analysis Tool for Manufacturing of Aircraft Components	44
4.3.3 MPM: Mass Properties Module	44
4.3.4 Compatibility	44
4.3.5 RVF: Requirement Verification Framework	45
4.3.6 Relation to production considerations and parameters of interest	45
5 Verification and Validation	47
5.1 The product model	47
5.2 MIM setup	47
5.3 RVF setup	49
5.4 Verification and validation of results	49
5.4.1 Manufacturing model	49
5.4.2 Database.	52

5.4.3	Assembly model	53
5.4.4	RVF	56
6	Industrial Use-Case: Conceptual Design of a Wingbox	57
6.1	Design workflow for the automated inclusion of production considerations in design using MIM.	57
6.2	Workflow setup for the use-case	61
6.2.1	Product topology (step 1)	61
6.2.2	Manufacturing model DOE (step 3)	61
6.2.3	Structural sizing (step 5)	61
6.2.4	Assembly model DOE (step 8)	63
6.2.5	RVF setup	63
6.3	Results and discussion	65
6.3.1	DOE I: manufacturing model.	65
6.3.2	DOE II: assembly model.	72
6.3.3	Identification of unapparent designs - research question 2.a.i	75
6.3.4	Identification of better designs - research question 2.a.ii	75
6.3.5	Effect of production considerations on gauging product viability - research question 2.b	76
6.3.6	Addressing limitations of the state-of-the-art	79
7	Conclusions and Recommendations	81
A	Glossary	85
B	ArchiMate Modelling Language Reference	87
C	Examples of User Inputs and Data Specifications in MIM	89
C.1	User input to define manufacturing model for a manufactured primitive.	89
C.2	MIM database file content examples	90
C.2.1	Material data file	90
C.2.2	Manufacturing method data entry.	91
C.2.3	Equipment set data file	91
C.2.4	Manufacturing site data entry	92
C.3	User input for operation set definition	92
C.4	User input for assembly station definition	93
D	Baseline Concept Data for the Industrial Use-Case	95
	Bibliography	97

List of Figures

1.1	Production considerations and parameters of interest within the scope of this thesis. . .	4
2.1	Manufacturability factors (in rectangles) organised under their respective core manufacturability concepts (CMCs) (in ovals) [21]. Note: the figure has been reconstructed using the original figure in [21] for clarity.	8
2.2	Process diagram showing the state of the art workflow for conceptual design at GKN Fokker Aerostructures. It includes prototype capabilities (such as RVF) at GKN Fokker that have been tested, but are not used in mainstream aircraft programmes. Legend for the ArchiMate notation used in this figure is shown in Figure B.1.	10
2.3	Product breakdown structure showing the included components in the study presented in [5].	14
2.4	Excerpt of a template of the text based input user to define geometrical product model in [7].	16
2.5	Graphical representation of the four machinability parameters considered in [9]. Note that the original captions from [9] have been included in the figures.	17
2.6	Suggested workflow using the aPriori Manufacturing Insights Platform.	18
2.7	List of production and economics related parameters available in the study of [10]. . .	18
2.8	AND/OR graph representation for the assembly of a product with four parts, represented by P, Q, R, S. The graph represents all possible combinations of assembly sequences. Source: [30]	20
2.9	Directed graph representation for the assembly of a product with four parts, represented by P, Q, R, S. The graph represents all possible combinations of assembly sequences. Source: [30]	21
2.10	Liaison diagram and matrix for the assembly of a product with four parts, represented by P, Q, R, S. Source: [30]	21
2.11	Precedence diagram for an assembly with four tasks. Source: [30]	22
3.1	UML diagram of the developed “manufacturing information model (MIM)” to manage production information of products.	23
3.2	Relation between the manufacturing model and manufactured primitives of a product, with an overview of the manufacturing model information categories.	24
3.3	UML class diagram of the manufacturing model integrated with an L-stringer manufactured primitive. The L-stringer primitive that represents its design is shown in yellow, whereas all other classes are related to the manufacturing model. Manufacturing information category classes are shown in purple. Remaining coloured attributes and classes are discussed in text. Note that the dependency links between the manufacturing information categories (in orange) are dynamic, and can change based on user input. Refer accompanying text for more details.	26
3.4	UML diagram showing the relation between a manufacturing method, equipment set, and equipment.	27
3.5	Structure of the database package showing the data directory and the libraries subpackage.	30
3.6	Screenshot showing the implemented result of the libraries subpackage.	31
3.7	Establishment of valid relations between the manufacturing information categories. The figure shows that valid relations do not have to be user defined (in blue and yellow) between all items of all information categories. From the user definition of some relations, the rest can be automatically derived (in green) through programmed rules in the libraries subpackage.	32
3.8	An example of an information query method in the libraries subpackage for the manufacturing methods category.	33

3.9	A simplified version of the operation graph to highlight some of its key aspects. Assembly stations are represented by the dashed orange boxes. Part and joint operation nodes are in dark/light blue, and result nodes are shown in light maroon. The overall product start and end nodes can also be seen. Some of the possible operation graphs are marked as 1, 2 and 3 with notes, but others can also be identified.	35
3.10	Simplified station graph with the critical manufacturing path highlighted by red edges. The critical path is calculated based on the <code>critical_time</code> attribute of each station node.	36
3.11	Process diagram showing the steps involved in setting up the assembly model.	36
3.12	Example of an operation set to install 20 ribs in a product.	37
3.13	Union of two operation sets. Duplicate input or output nodes that were defined to connect the operation sets are removed, and converted to single connectivity node. This is done while retaining edge relations. Note that the rectangle shapes between the input and output nodes are placeholders for the operation and result nodes of the operation sets. They are not shown as they are not relevant, and to keep the figure size manageable for clarity.	38
3.14	UML class diagram of the assembly model integrated in a product. Classes within the assembly model subpackage are highlighted in yellow.	39
4.1	Overview of the steps required for integrating MIM with a ParaPy based KBE application.	42
5.1	Manufactured primitives in the product model for verification and validation of MIM's implementation.	47
5.2	Integration of the manufacturing model in different types of manufactured primitives in the product model for the verification and validation case. Screenshots show the tree structure in ParaPy GUI.	50
5.3	Design specifications information category for the skin panel and joint 2 primitive in the verification and validation case.	51
5.4	Important slots in the manufacturing method information category for the skin panel primitive.	51
5.5	Mappings of selections in method and site information categories to other analysis tools in the system model.	52
5.6	Example from the database libraries, showing the available attributes for an equipment.	52
5.7	Results of information queries to get valid manufacturing methods using the database libraries subpackage. Screenshot shows the Python console window.	53
5.8	Tree structures showing details of the assembly model in a product for verification and validation.	53
5.9	Operation graph for the overall assembly sequence in the verification and validation case.	54
5.10	Station graph for the verification and validation case.	54
5.11	Operation graph for the production station <i>station_side1_stringer_installation</i> in the verification and validation case.	55
5.12	Integration of the verification and validation case product model in the RVF.	56
5.13	Auto-generated requirement compliance report from RVF for the verification and validation case.	56
6.1	Workflow for the automated inclusion of production considerations in conceptual designs using MIM. Note that step 5 (structural sizing) was excluded from the scope of this thesis due to time limitations. Material thickness values were set using some available reference values for the baseline concept (Table D.4), in consultation with experts at GKN Fokker Aerostructures.	58
6.2	Wingbox part primitives in the industrial use-case. Note: the top skin panel has been hidden for clarity.	61
6.3	Wingbox joint primitives representation in the industrial use-case.	61
6.4	Screenshots showing a few design points in the resulting spreadsheet from the manufacturing model DOE. The spreadsheet has been split into three sub-figures to improve readability. The design point index can be referred to in column A in each sub-figure.	66

6.5	Scatter plot of all valid design points from the manufacturing model DOE. The requirements imposed on the total cost and total mass are represented by horizontal and vertical red lines in the plot respectively. All design points that satisfy these requirements are referred to as feasible, while the others are infeasible. Eight groups of design points, each with very similar total mass can be identified. Total cost is calculated by using CATMAC.	67
6.6	Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected part material. The default material for all parts is CFRP, unless stated otherwise for some parts by the legend key. The baseline design and design selected for the assembly model DOE are marked with an arrow. Two types of markers are used for the design points (circle/triangle), based on the selection made for the stringer design.	67
6.7	Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected manufacturing method for joints. The default joining method is induction welding, unless stated otherwise for some joints by the legend key. It can be observed that the joining method is closely related to the identified eight mass groups.	68
6.8	Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected manufacturing method for parts. The default part manufacturing method is hand layup, unless stated otherwise for some parts by the legend key. No correlation between the part manufacturing method and mass groups can be observed. A relation between the part manufacturing method and total cost can be observed within each mass group. Note that not all possible combinations of selections are shown in the legend due limitation of space. These missing combinations are shown together with the legend key "Other". T.P. = Thermoplastic; M/c = Machining.	68
6.9	Selected design point from the manufacturing model DOE (index 271 - in dark green), for analysis in the assembly model DOE.	71
6.10	Screenshots of assemble model DOE results. The two sub-figures show a single design point, with the design point index shown in column A.	72
6.11	Selected design point from the assembly model DOE (index 274 - in white text).	72
6.12	Station graph of design point 751 in the assembly model DOE.	73
6.13	Station graph of design point 841 in the assembly model DOE.	73
6.14	Operation graph of design point 706 in the assembly model DOE. The reader is suggested to zoom-in in the digital version of this report.	74
6.15	Operation graph of design point 274 in the assembly model DOE. The reader is suggested to zoom-in in the digital version of this report.	74
6.16	Operation graph of design point 706 in the assembly model DOE. Zoom level 1.	74
6.17	Operation graph of design point 274 in the assembly model DOE. Zoom level 1.	74
6.18	Operation graph of design point 706 in the assembly model DOE. Zoom level 2.	74
6.19	Operation graph of design point 274 in the assembly model DOE. Zoom level 2.	74
6.20	Effect of accounting for part integration in a product on the total cost and mass.	76
B.1	Legend describing the ArchiMate modelling language notation used in this report.	87

List of Tables

4.1	Compatibility categories and criteria included with the scope of this research.	44
4.2	Which model or analysis tool accounts for which production consideration.	45
4.3	Which model or analysis tool accounts for which quantity of interest.	45
5.1	Manufacturing model information category selections for the manufactured primitives in the case for verification and validation.	48
5.2	Assembly model configuration for the verification and validation case.	48
5.3	Requirements imposed on the verification and validation case.	49
6.1	Design matrix of the manufacturing model DOE for the industrial use-case.	62
6.2	Details of the design point selected from the manufacturing model DOE.	63
6.3	Design matrix of the assembly model DOE for the industrial use-case.	64
6.4	Requirements imposed on product in the industrial use-case.	64
6.5	Breakup of total cost and total mass to show contribution of joints, for some design points with different joining methods. All design points have part material as CFRP, have L-stringers, and mechanical rib-spar joints are always included. The default joining method is induction welding, unless stated otherwise for some joints in the first column.	78
D.1	Parameters of interest for the baseline concept finalised at GKN Fokker Aerostructures.	95
D.2	Requirements imposed on the baseline concept finalised at GKN Fokker Aerostructures.	95
D.3	Manufacturing details of product primitives in the baseline concept finalised at GKN Fokker Aerostructures.	96
D.4	Reference material thickness values for part primitives for the baseline concept at GKN Fokker Aerostructures. Note that the average thickness is shown due to the presence of material zones of varying thicknesses on each part.	96

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
CAD	Computer Aided Design
CATMAC	Cost Analysis Tool for Manufacturing of Aircraft Components
CER	Cost Estimation Relation
CFRP	Carbon-Fibre-Reinforced Polymer
CMC	Core Manufacturability Concept
DAG	Directed Acyclic Graph
DFM	Design for Manufacturing
DOE	Design of Experiments
FEM	Finite Element Method
GUI	Graphical User Interface
JSON	JavaScript Object Notation
KBE	Knowledge-Based Engineering
MBSE	Model-Based Systems Engineering
MDAO	Multidisciplinary Design Analysis and Optimisation
MDM	Multidisciplinary Modellers
MIM	Manufacturing Information Model
MMG	Multi Model Generator
MPM	Mass Properties Module
OML	Outer Mould Line
PLM	Product Lifecycle Management
RVF	Requirement Verification Framework
TAPAS	Thermoplastic Affordable Primary Aircraft Structure
UID	Unique Identifier
UML	Unified Modeling Language

1

Introduction

Considerations that dictate producibility of a design are among the most important criteria that drive selection of a concept over others. This is because a design cannot be successful if it is not possible to manufacture it within the constraints and capabilities of the stakeholders. These production considerations can be defined as, “*factors from the perspective of production that have an influence on the system design*”. For instance, they can include the joining method for product assembly, or ensuring

Production consideration

A factor from the perspective of production that has an influence on the system design.

compatibility between choices made for the part material, manufacturing method and equipment that will produce it. These considerations can have a direct impact on the design. For example, different joining methods (such as fasteners or welding) have different requirements for minimum distance to edge and corner radius, which in turn would result in different flange widths for parts such as ribs. Similarly, if the selected

part material and manufacturing method are incompatible, the part may have to be redesigned to ensure that the design also remains compatible with the new material or manufacturing method.

The importance of including production considerations in design can be highlighted by analysing some problems that may arise without them. From discussion with experts at GKN Fokker Aerostructures¹, a tier 1 aerostructures manufacturer, the first problem is identified of such designs being associated with a high risk. This risk can be associated with either design changes at later stages of development to include the missing production considerations (such as accounting for correct fastener arrangement and assembly clearances), or with the reduced fidelity of the system and analyses that depend on it (such as cost and mass). Design changes are directly associated with higher costs and delays, which reduce a product’s viability. Additionally, analyses on a system with lower fidelity makes the calculated performance indicators of interest less reliable or unavailable for trade-off decisions. For example, additional mass of joints is often estimated as a percentage of the total mass of an aircraft structure [1]. This makes it difficult and inaccurate to trade-off two different joining methods in terms of their mass. Lastly, these risks can be especially high for novel designs for which no prior data or engineering experience exists.

The second problem is in part related to the first problem of high risk, as in order to mitigate this risk, engineers tend to limit themselves to safe conservative choices [2]. This ensures producibility of the designs, but prevents use of the best possible options which might result in better designs, for example, by selecting a new manufacturing method or material.

Currently, two main approaches are followed to include production considerations in the conceptual design of aircraft structures. The first is through a manual process in which a design engineer includes aspects of production in product design by using three main sources of information. These include 1) *documents*, such as design for manufacturing (DFM) guidelines, and documented company best practices; 2) *discussions* with manufacturing and stress engineers; and 3) their own *experience*. This results in a design which will be mostly feasible to produce, though it might require some minor changes in future. The resulting design concept is then optimised for its performance (such as structural sizing)

¹<https://www.gknaerospace.com/en/about-gkn-aerospace/fokker-technologies/> (accessed 25 September 2022)

through multidisciplinary design analysis and optimisation (MDAO) workflows. This manual approach represents a case from the industry as it has been identified from discussions with experts at GKN Fokker Aerostructures. It is currently followed by them for new product development.

Although this approach works, and gives engineers complete control over each step in the process, it has some disadvantages. Firstly, the process can be cumbersome and error-prone because of the extensive use of documents. Secondly, as the process depends on engineer's experience, it is possible to overlook some design choices that may be unconventional but better. Additionally, availability of experts cannot be relied upon, as they can leave the organisation, which can lead to project delays. Third, discussions with manufacturing and stress engineers can be helpful, but are usually very time-intensive. It can be deduced that these problems can lead to a bottleneck if multiple design concepts have to be traded-off, as each would require all these manual steps to ensure its producibility before it can be optimised for performance using MDAO. As a result, only a limited number of concepts can be considered in the trade-off because of the limited time at the conceptual design stage, potentially leaving out designs that may fit the requirements better.

One last disadvantage of the manual process is related to management of production related requirements. Efficient management of production related requirements can be very important to avoid added cost and delays, as these requirements can often conflict with other requirements on the system. For example, induction welded joints may be better for the total mass of the aerostructure, but fasteners may be required to fulfil certification requirements (from interviews with GKN Fokker Aerostructures experts). Because these production considerations are not digitally modelled in the manual process, model-based requirement management tools such as the requirement verification framework (RVF) [3], or commercial tools such as *IBM DOORS*² cannot be used for their efficient management.

The second approach to include production considerations in conceptual designs is through automated methods. Various approaches that attempt to automate aspects of the manual approach can be found in literature. However, none present a methodology that is free from some major limitation, which has prevented their mainstream adaptation. These studies can be broadly classified into four types based on their limitations, which are listed below:

- i. Using only the associated cost as an all-encompassing factor for all production considerations [4–6].
- ii. Non-generic solutions that only work with very specific types of product designs, manufacturing processes, materials, etc. [7–10].
- iii. Methodology depends heavily on the use of some commercial software such as product lifecycle management (PLM) tools or manufacturability analysis software that are not fully suitable for automation and/or use at the conceptual design stage [10–12].
- iv. Aspects of part manufacturing and product assembly are not considered at the same time. Some studies that only consider manufacturing are [4, 5, 9], and those that only consider assembly include [13–17].

In view of limitations of the current approaches, this thesis will focus on developing a methodology that enables automated inclusion of production considerations in the conceptual design process, while overcoming shortcomings of the state-of-the-art. To this end, the following research objective is defined for this thesis:

To automate the inclusion of production considerations in the conceptual design of aircraft structural components by developing an MBSE based approach to manage production information in the system design process.

Limitations of the manual approach have been overcome through the second part of the objective; through a model-based systems engineering (MBSE) approach to manage all production information. MBSE is defined by the International Council on Systems Engineering (INCOSE) as, "...the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." [18]. Formalisation of knowledge from documents and experience of engineers

²<https://www.ibm.com/products/requirements-management> (accessed 05 August 2022)

would allow its systematic re-use, and its digitisation would support automated design workflows, such as design of experiments (DOE) and MDAO studies. Furthermore, once data related to production considerations is made available in digital models, model-based requirement management tools can also be used to effectively manage production related requirements.

On the other hand, limitations associated with the state-of-the-art automated methods has

been addressed by (i) including production considerations other than manufacturing cost, (ii) developing the models in a way that enables them to be applicable to any component in a product, (iii) developing the models independent of any specific software, and (iv) including aspects of both, manufacturing and assembly for a product.

To help achieve the research objective, and to gauge the contribution of this research, the following research questions have been formulated:

1. How can information related to the production considerations, along with its link to the design, be structured in a model so that it gives an overview of the steps to manufacture the product, and it can be used for the calculation of cost, structural mass, production rate and requirements compliance?
2. What effect does the automated inclusion of production considerations in the conceptual design of a wingbox have on:
 - a) being able to identify feasible-to-produce design concepts:
 - i) that may be unapparent?
 - ii) that are better, when compared in terms of their cost, structural mass, production rate, and meeting production related requirements?
 - b) gauging the product's viability in terms of its cost, structural mass, production rate, and meeting production related requirements?

Model-Based Systems Engineering (MBSE)

...the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [18].

1.1. Thesis Scope

A product's design can include numerous production considerations. Therefore, to limit the scope of this thesis, only four main categories of production considerations will be considered. These include joints, compatibility, assembly sequence, and part integration, and are shown graphically in Figure 1.1 (sources: icons³, material selection chart⁴, co-bonding process [19], production consideration 3 and 4: GKN Fokker Aerostructures).

The first and second production considerations of joints and compatibility have already been briefly discussed in the previous paragraphs. The third consideration deals with generating feasible assembly sequence to account for product assembly. And the fourth production consideration is of part integration, as in some cases individual functional parts in the product design, such as stringers and skin panels, can be integrated into a single part, such as a stiffened skin panel [20].

Lastly, The effect of all these production considerations would be quantified in terms of four key parameters of interest. These include the product's total cost, total mass, production rate, and the requirements' compliance.

³<https://www.flaticon.com/> (accessed 10 August 2022)

⁴http://www-g.eng.cam.ac.uk/125/now/mfs/tutorial/non_IE/charts.html (accessed 10 August 2022)

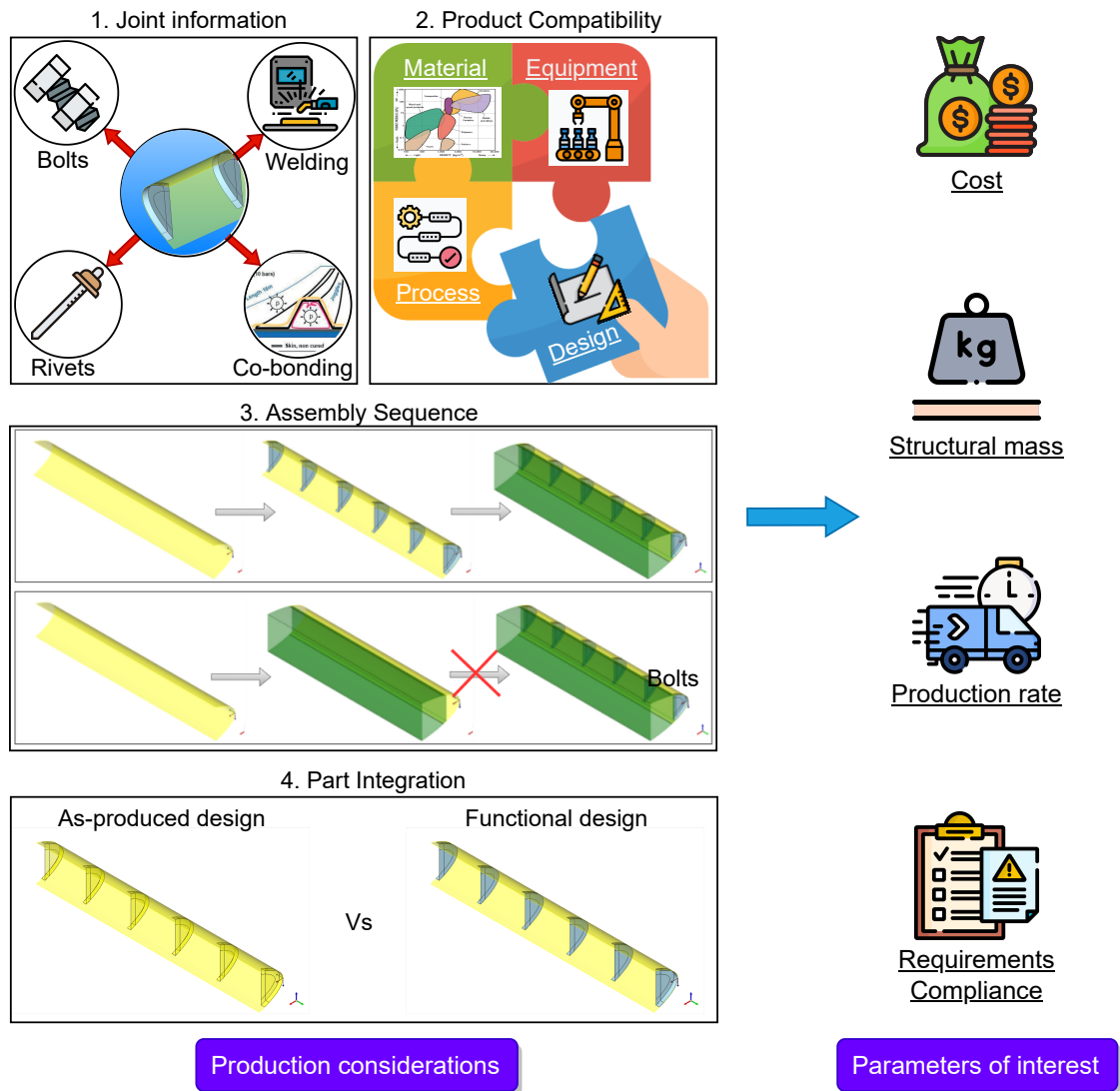


Figure 1.1: Production considerations and parameters of interest within the scope of this thesis.

1.2. Thesis Outline

- **Chapter 1: Introduction**

The current chapter introduces the main area of focus for this thesis. The need and motivation for conducting this research is highlighted, and the research objective, questions, and scope are defined in this chapter.

- **Chapter 2: State of the Art and Background Information**

This chapter discusses some state-of-the-art approaches for including production considerations in design, and provides some background information.

- **Chapter 3: The Manufacturing Information Model**

A theoretical explanation of the main contribution of this thesis is given in this chapter.

- **Chapter 4: Implementation of the Manufacturing Information Model**

Details of how the manufacturing information model is implemented, and what supporting software tools are used in this thesis are given in this chapter.

- **Chapter 5: Verification and Validation**

Implementation of the manufacturing information model is verified and validated in this chapter by applying it to a simple use-case.

- **Chapter 6: Industrial Use-Case: Conceptual Design of a Wingbox**

The manufacturing information model is applied in an updated design workflow for the conceptual design of a wingbox to analyse its real world applicability.

- **Chapter 7: Conclusions and Recommendations**

In this chapter, conclusions are drawn from the results of the industrial use-case, and the verification and validation case. Recommendations for future work are also provided.

- **Appendix A: Glossary**

A glossary of all terms defined in this report (in blue boxes) is provided in this appendix.

- **Appendix B: ArchiMate Modelling Language Reference**

A legend for the ArchiMate notation used in some of the diagrams in this report is provided in this appendix for reference.

- **Appendix C: Examples of User Inputs and Data Specifications in MIM**

Some templates for specifying user inputs, and some data in the manufacturing information model are provided in this appendix for reference.

- **Appendix D: Baseline Concept Data for the Industrial Use-Case**

Data for a baseline wingbox concept that was finalised at GKN Fokker Aerostructures, and is used as a reference for the industrial use-case in this thesis, is provided in this appendix.

2

State of the Art and Background Information

This chapter will discuss some state-of-the-art approaches for including production considerations in conceptual designs, and also provide background information on some relevant topics. The first section will give an overview of the various production considerations in design. A manual approach for including production considerations in design that is practically followed in the industry will be discussed in the second section. The third section will then present some cases from literature that aim at automatically accounting for production considerations in design. Finally, the fourth section will give an overview of some graphical methods for representing the assembly sequence.

2.1. Overview of production considerations for product design

An extensive range of factors that influence product manufacturability and design have been identified in [21]. The authors of this paper define manufacturability as “... *the ability to manufacture a product to obtain the desired quality and rate of production while optimizing cost* [21]”. By studying some common issues that arise when developing new products, they categorised factors that influence manufacturability under five main types: compatibility, complexity, quality, efficiency and coupling. These categories were referred to as “core manufacturability concepts (CMCs) [21]”, and are shown graphically in Figure 2.1 along with their respective manufacturability factors. The reader is can refer to their work for a description of each manufacturability factor with examples.

The goal of the authors while identifying these factors was to enable better designs, rather than to improve the manufacturing processes. To enable feedback for design improvement, quantification of these manufacturability factors was suggested as this would allow systematic comparisons to be performed between designs. For this, the authors present two approaches: using manufacturability measures and using manufacturability indices [21]. The former is a metric that uses an analysis of the manufacturing process, whereas the latter involves an analysis of the product design variables. The manufacturability measures were further classified into three types [21]:

- Type I: rely on estimating manufacturing cost
- Type II: estimate the manufacturing time (which can be easily converted to cost using labour and machine rates)
- Type III: involve an analysis of the sequence of processes for manufacturing/assembly to identify less efficient tasks.

Although metrics from these analyses can be very helpful, all of them suffer from a common limitation; they are difficult to calculate reliably at the conceptual design stage because of a lack of information [21]. The use of manufacturability indices was suggested by the authors to overcome this limitation. These “indices” are directly related to product design variables, such as surface finish, tolerance or tool accessibility. Instead of translating these factors into cost or time, the authors suggest directly using these factors for comparing designs. For this purpose, the factors would be quantified into normalised scores with some relevant parameters of the designs. They present an example for

quantifying tool accessibility for machined parts, which is based on identifying primary and secondary angles of accessing part features with reference to a coordinate system [21]. A normalised accessibility index was then calculated from the values of these angles, and used to compare two designs for tool accessibility. The main advantage of using manufacturability indices over manufacturability measures is that they can be much easier to calculate, and overall require less information.

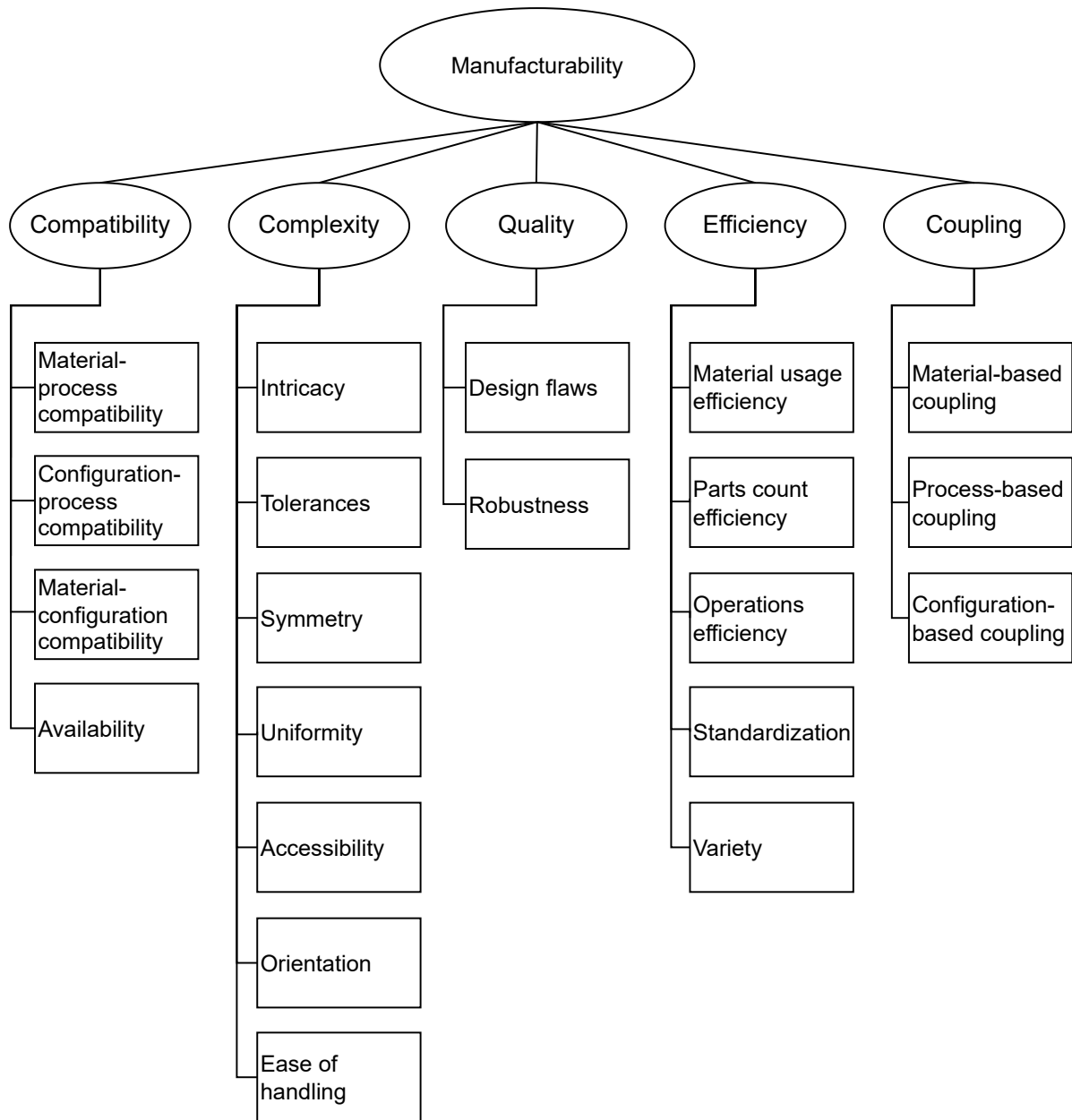


Figure 2.1: Manufacturability factors (in rectangles) organised under their respective core manufacturability concepts (CMCs) (in ovals) [21]. Note: the figure has been reconstructed using the original figure in [21] for clarity.

A different study extended the manufacturability factors in the compatibility CMC to include “material-material compatibility” [22]. This consideration ensures that materials of the individual parts in a product are compatible with each other. For example, a part made from carbon-epoxy composite is incompatible with an aluminium alloy part as it would result in galvanic corrosion in the aluminium part [22].

Although the manufacturability factors presented in [21] can be applied to aspects of product as-

sembly as well, there are a few additional factors that are quite specific to product assembly. One of these is assembly sequence planning, which involves coming up with a strategy to assemble the individual parts or subassemblies in a way that is feasible, and meets the required performance in terms of production volume or rate [13]. Many options can exist for feasible assembly sequences, and therefore it is important to select the most optimum one as it can have a big impact on the overall efficiency of the assembly process [13]. Additionally, if an optimum assembly sequence cannot be found for a product definition, then changes might be required to the product design to meet the assembly requirements. For instance, this may involve splitting the high level product components or subassemblies differently.

Some other production considerations related to product assembly include factory floor layout for the manufacturing and assembly stations, equipment idle time, and supply chain considerations. These were identified from discussions with experts at GKN Fokker Aerostructures. Factory floor layout is related to assembly sequence planning as it has an effect on the time taken to transfer raw material, parts and subassemblies between workstations, thereby affecting the production rate, and the overall efficiency of the production process. Equipment idle time refers to the time when an equipment is not in use, but is actually available for use. As no value is being generated during this idle time, it is desired to minimise it as much as possible. This can again be accomplished with a suitable assembly sequence. Lastly, supply chain considerations can also directly effect production as companies depend on receiving raw materials, parts, and even subassemblies from various suppliers. Production problems at the suppliers, or delays in delivery can lead to delays, and added costs, for example because of worker idle time.

To conclude, this section gives an idea about some of the numerous production considerations that have to be taken into account for product design. It is to be noted that the ones presented in this section do not exhaustively cover all possible production considerations. It is possible to have many more of these factors, and ultimately, which ones are considered in product design will depend on the specific requirements from the product.

2.2. Manual approach for including production considerations in conceptual design

This section will discuss specific details of the manual approach for including production considerations in design that was introduced in chapter 1. The main steps involved in this approach have been summarised as a process diagram in Figure 2.2. The ArchiMate notation used in this figure (and other ArchiMate process diagrams in this report) is shown in Figure B.1. The steps in this approach have been identified through discussions with experts at GKN Fokker Aerostructures. It represents the state-of-the-art workflow that is followed there for the conceptual design of new products. It is to be noted that the workflow presented in Figure 2.2 also includes prototype capabilities (such as RVF) at GKN Fokker Aerostructures. Although these capabilities have been tested, they are not currently used in mainstream aircraft programmes.

The production considerations and parameters of interest in the scope of this thesis are specified in coloured text in Figure 2.2 to put them in context of the presented workflow. This shows exactly at what step they are accounted for. The production considerations are shown in blue, and the parameters of interest are shown in orange. They are listed below in their respective colours for reference:

Production considerations:

- (1) Compatibility
- (2) Assembly sequence
- (3) Part integration
- (4) Joints

Parameters of interest:

- (i) Cost
- (ii) Mass
- (iii) Requirements compliance
- (iv) Production rate

Next, the various steps in the presented workflow will be described in more detail:

- **Steps 1, 1a, 1b, 1c**

The first step involves deciding the product topology, which includes deciding what parts should exist in the product (e.g. ribs, spars, skins etc.), the type of each part (e.g. L-stringer or blade stringer), quantity, location, and orientation of each part, and so on. After deciding the topology, it is decided how each part will be manufactured in step 1a. This involves selecting the material, manufacturing method, manufacturing equipment, and the site where each part will

be manufactured. This is followed by step 1b, where compatibility between the choices made for manufacturing each part, and the design choices made while fixing the topology is checked. If some choices are incompatible, then changes have to be made in either the design, or manufacturing related aspects. The back-and-forth double arrows between the steps, along with the looped arrow indicate the iterative nature of these steps. On the other hand, step 1c also occurs in parallel after deciding the product topology. In this step, it is checked if the selected layout of parts will be feasible to assemble. If not, the design has to be modified to allow for it. The four steps discussed here are all performed manually by engineers, who use three main sources of information to make decisions. These include documents (such as design guidelines or noted best practices), meetings with other engineers, and their own experience.

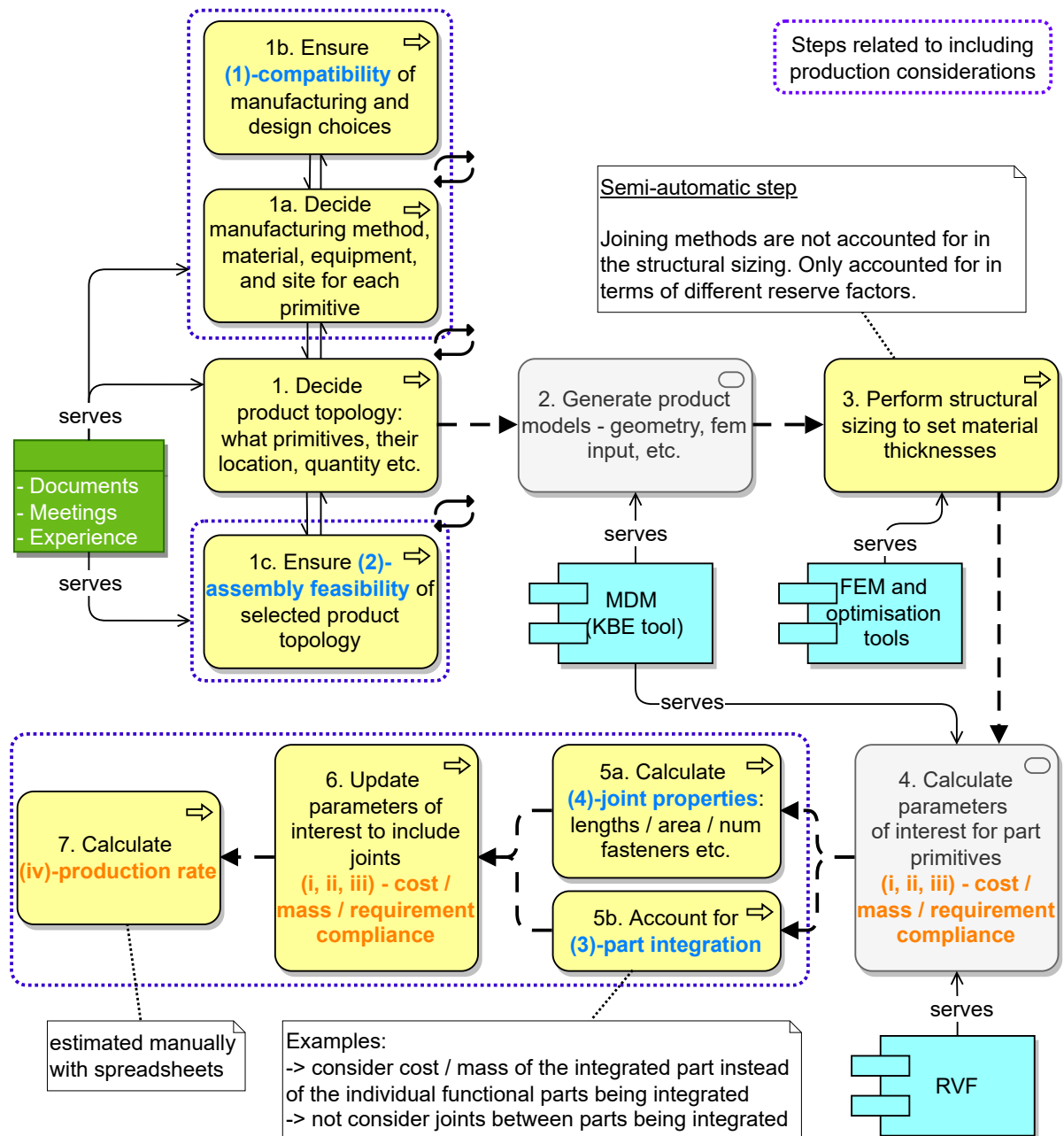


Figure 2.2: Process diagram showing the state of the art workflow for conceptual design at GKN Fokker Aerostructures. It includes prototype capabilities (such as RVF) at GKN Fokker that have been tested, but are not used in mainstream aircraft programmes. Legend for the ArchiMate notation used in this figure is shown in Figure B.1.

- **Step 2**

This is an automatic step that generates different product models using a knowledge-based engineering (KBE) tool. For instance, these models can include a computer-aided design (CAD) model to represent the geometry of the product, or a finite element method (FEM) model for structural analysis. The KBE tool has been developed in-house at GKN Fokker Aerostructures using ParaPy¹, and is called as multidisciplinary modellers (MDM). More details about this tool will be discussed in [section 4.3.1](#).

- **Step 3**

This is a semi-automatic step that involves setting up and running an optimisation workflow for structural sizing. This results in optimum values for material thicknesses for all parts in the product. Variation in thickness at different regions in a given part can also be accounted for as MDM supports the creation of “material zones” per part; a different value for thickness can be defined for each such material zone. It is to be noted that the current capabilities do not account for what joining method is used to connect parts in the structural sizing optimisation. As a simplification, nodes are considered to be equivalent at part junctions to represent a joint. The effect of different joining methods is only be considered in terms of different reserve factors.

- **Step 4**

This is an automatic step where certain parameters of interest of the product are calculated. Namely, these parameters include the cost, mass and requirements compliance. The cost and mass calculation is handled by MDM, whereas the requirements compliance is handled by RVF. More details about RVF will follow in [section 4.3.5](#).

- **Steps 5a, 5b**

These two manual steps are carried out at the same time. MDM does not support modelling of joints, and therefore, properties related to them have to be manually calculated based on the part geometry model. These properties can include the joint lengths and/or area, number of fasteners etc., and are required to account for joints in the parameters of interest. The other step (5b) involves accounting for integration of separate parts in the product to form a new single part (such as an integrally stiffened skin panel). If such integrated parts are to be included in the model, then their effect on the parameters of interest, and other parts or joints in the product model have to be considered. For example, this can include only considering the cost & mass of the new integrated part in the total cost & mass, and not the cost & mass of the individual parts that are integrated. Another example is to not consider any joints between the individual parts that are integrated. This is the reason why steps 5a and 5b must take place at the same time, as information about part integration has to be considered to know what joints exist in the model.

- **Step 6**

Once joint properties are made available from the last step, they are used to (manually) calculate the corresponding cost, mass and requirements compliance for the joints. The parameters of interest calculated for all parts in step 4 are then be updated with these values for the joints to give totals for the entire product. Requirement compliance is checked manually.

- **Step 7**

The last step involves a rough estimation of the overall production rate for the product, which is performed manually using some spreadsheets.

In conclusion, limitations of the manual approach that were discussed in [chapter 1](#) are only further justified after considering the detailed steps of the process. The first step to come up with a valid feasible-to-produce design concept is manual and iterative, and thus forms a major bottleneck in the process. This will be especially evident if the process has to be repeated multiple times in order to trade-off different concepts. As a result, only a limited number of concepts can be considered that account for production at the conceptual design stage.

¹<https://www.parapy.nl/> (accessed 16 August 2022)

2.3. Automated approaches for including production considerations in conceptual design

This section will discuss some studies that attempt to automate the inclusion of production considerations at early stages of design. The discussion will be organised in terms of the major limitations that could be identified from these studies.

2.3.1. Accounting for production considerations only through the associated cost

One of the most common approaches that was observed across various studies for including manufacturing and/or assembly factors into the design process was through their associated cost. Cost is an important consideration, not only because it decides a product's profitability or affordability, but also because most other factors can be directly or indirectly linked to the cost. For example, a design that is more complex to manufacture would usually lead to a higher cost. This would suggest that, in theory, cost can account for other production considerations as well, and explains why it has been the focus of many studies. Therefore, this section will discuss some studies from literature to get a better understanding of how cost can be used to include manufacturing and assembly considerations in design. Before moving on to the studies, a brief background on the different types of cost estimation methods and models will be given.

Types of cost estimation methods

Three types of cost estimation methods can be identified from literature [23]:

1. Analogous

These types of cost estimation methods are based on using historical data of similar products to estimate the cost. The differences between the old and new design can be compared, and accounted for to some extent to get a better cost estimate.

2. Parametric

In this method, cost estimation relations (CERs) are used to directly estimate the product cost. CERs are formulae derived from historical data of similar products, that relate cost to certain product features, such as weight or dimensions. As an example, such a model is presented in [6].

3. Process based / bottom-up

In these types of cost estimation methods, all steps, processes, and information about resources used during production of a product are accounted for. This requires a lot more information upfront compared to parametric and analogous cost estimation method. However, the main advantage of this method is that it is more flexible to work with new products and processes, for which historical data is not available. Examples include of some of these models [4, 24].

Lastly, it is also possible to use combinations of the above methods. For example, a combination of parametric and process-based methods is used in [25]. Using such an approach can be useful, as it combines advantages of both the options. Results can be obtained even when some information required by the process-based approach is not available. Therefore, it is described as an ability to perform calculations at different levels of fidelity in [25].

Proprietary vs. self-developed cost models

The next observation from literature is about the choice between either developing cost models ([5, 6, 24]), or using commercial cost estimation software ([8, 10]) such as SEER-MFG² and aPriori³. Either choice is associated with certain advantages and disadvantages.

Self-developed cost models offer the most flexibility in terms of integrating company specific information, while also ensuring that the model will work well with existing workflows. However, these models are commonly limited to only a few manufacturing and assembly processes, as limited resources can be dedicated to increasing the capabilities of these models. This is one area where commercial tools have an advantage, as they often offer extensive industry specific libraries of manufacturing models⁴.

²<https://galorath.com/products/seer-for-manufacturing/> (accessed 20 August 2022)

³<https://www.apriori.com/solutions/products/> (accessed 20 August 2022)

⁴<https://www.apriori.com/solutions/industry/> (accessed 20 August 2022)

Another factor is the cost associated with developing or acquiring these tools. Commercial tools will have a recurring subscription cost associated with them. On the other hand, tools developed in house will not have a subscription cost, but will still have some associated recurring cost to maintain the tool. This can be a difficult choice, as now the cost of developing and maintaining these tools in-house also has to be determined.

Another option, which might reduce the cost associated with maintaining self-developed tools, is to use open source software, such as the open source cost tool presented in [24]. But the availability of such specialised tools as open source software is rare, limiting users to just a few options. Additionally, their development and maintenance cycles can be unreliable when compared to the support offered by commercial tools, or even with the dependability of having a dedicated in-house team working on developing these cost models.

A big advantage of in-house developed or open source cost models is that the logic behind calculations is transparent. This can be essential for understanding and making sense of the results. Though commercial software might provide cost estimates that are very accurate in an absolute sense, they act as blackboxes, as the inherent logic or algorithms used by them is proprietary information. This is an important consideration while selecting the best option for use in MDAO workflows, where getting accurate cost estimates might not be a priority, as long as the trends are correct [5, 24]. For example, estimating cost of a design with different manufacturing processes must show the right trend for how expensive a process is, relative to the others. Therefore, If minimising for cost, this will ensure that the optimised design has the least cost, even if its absolute value is not accurate.

Cases from literature

Some cases from literature that account for production considerations through the associated cost will be discussed in this section. The degree to which these approaches use design specific information in their methods will also be analysed. This is important because if the estimated cost cannot be traced back to design choices at the part or product level, then the results from these tools cannot be used to improve the design.

The first study is by Gantois and Morris [5]. As a first step in their approach, the product (a wingbox in this case) is defined with the help of a multi-model generator (MMG). Their product definition was quite detailed, and contains information about substructures as well (such as joints, stringers, access doors etc.). The complete product breakdown structure is shown in Figure 2.3.

Then, as a second step, each component of the defined product is analysed to make selections for its material, manufacturing method and assembly method. For estimation of cost, they developed two separate cost models, one for a metal structures, and one for composite materials. However, their cost models only utilised very simple geometric parameters of the components, such as length, area and number of features (like holes). This information was used along with a cost factor, which was determined from different sources and internal communications from British Aerospace Airbus [5]. Additionally, the second step of defining the right material, manufacturing and assembly method is performed completely manually, and has to be repeated for every new type of design. This can be a slow process if multiple concepts have to quickly traded-off.

In a different study, a commercial cost estimation tool (SEER-MFG) was used instead of a self-developed cost model [8]. Both part manufacturing and assembly costs were included, however, their approach was not based on the use of a CAD model. Instead, they used simple parameters, such as dimensions of the wing and the fuselage, to define an outer mould line (OML). Then, a MATLAB⁵ based “manufacturing translator” was used to generate information about the internal structure, such as the size and layout of ribs and spars. Then some “rules of thumb” were used by them to estimate the number of fasteners in each part. Finally this manufacturing data was fed to the SEER-MFG cost tool.

By limiting themselves to simple geometric parameters such as overall OML dimensions, area or number of features, these studies miss out on geometric features, such as complexity of a curved surface or ply ramps in composite parts, that can play an important role in estimating cost. While these approaches can generate quick results, they do not use or allow for the definition of unique or complex product models. This can be essential in performing effective trade-offs between design concepts, when the differences in the designs are quite small.

These challenges have been addressed to some extent by the open-source cost estimation tool [24]. This tool uses process-based models that use data about the manufacturing process and methods,

⁵<https://mathworks.com/products/matlab.html> (accessed 20 August 2022)

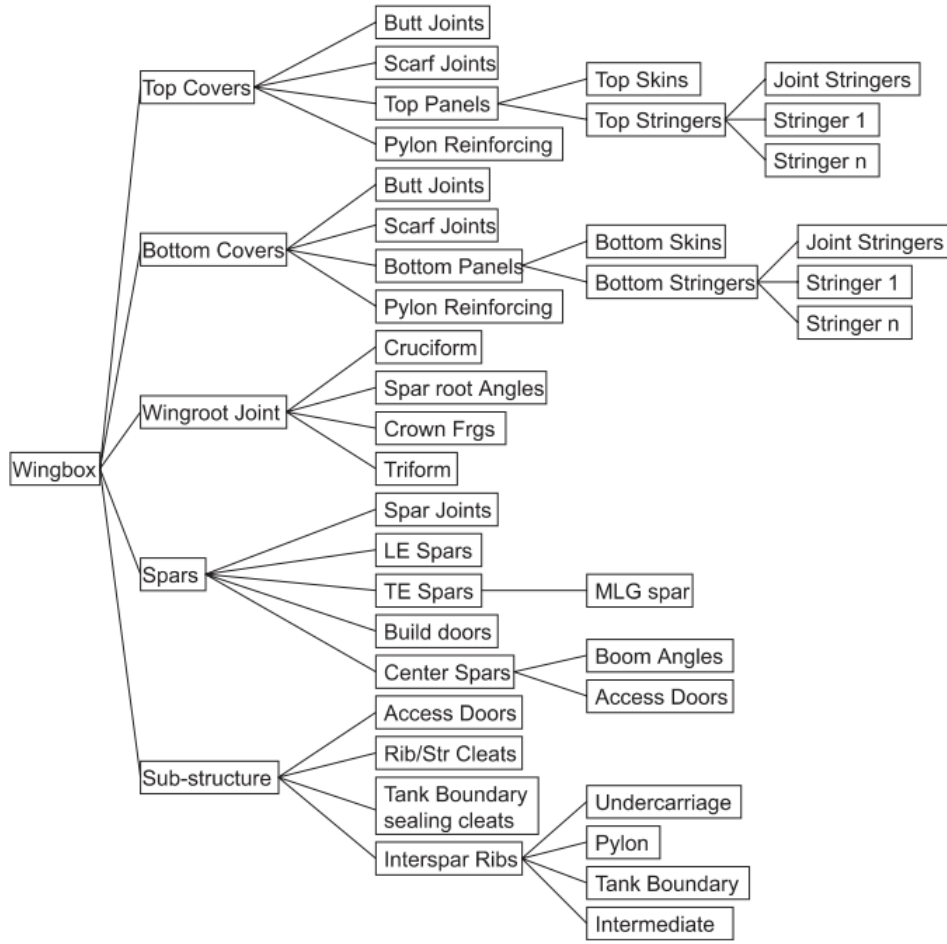


Figure 2.3: Product breakdown structure showing the included components in the study presented in [5].

manufacturing environment (labour and machine rates), materials, and connection method (fasteners, bonding etc.). An initial approach towards including shape complexity was also provided, however, it was reported that more work is required to fully support its inclusion in the analysis [24]. The cost tool was integrated in a KBE framework for generating wing moveables [26], which is capable of producing detailed and unique product models.

Despite its capabilities, the tool does have some limitations. It only supports a limited number of manufacturing processes, which were mostly for composite based components. Additionally, it uses information from an outdated database. Connections for assembly had to be manually defined by a user, instead of using information from the connected design model. Additionally, it was reported that for some manufacturing methods, the tool generated values that deviated by approximately 50%, when compared to another proprietary cost estimation tool [24]. Although, absolute cost estimates are not of primary importance for trading off different design concepts, such a big discrepancy might suggest possible problems with the cost model for these manufacturing methods.

Overall, there have been various attempts at quantifying manufacturing and assembly considerations through cost, although, for it to give meaningful results, the fidelity of the cost analysis tools will have to be increased to consider more design specific details. The first limitation of this approach of only using cost is that there are factors which might be more relevant to consider directly, instead of their associated cost. For example, production rate can be an important factor for a trade-off between design concepts if achieving a certain production rate is an important project requirement. And the latter is often true to ensure just-in-time production, a commonly followed lean manufacturing practice in the aerospace industry [27].

The second problem with this approach is that cost models cannot identify infeasible designs. A

design concept can be infeasible, for example, if the selected manufacturing processes are not valid for the corresponding type of parts or their materials. Usually, this validity is ensured manually by an engineer while finalising details of the design concept, which is then optimised for performance. Such a case has been presented in [5], and was also discussed in [section 2.2](#). The importance of automating this step is evident if multiple design concepts have to be traded-off in a limited time.

Therefore, it can be concluded that using cost as an overall metric for analysing a product from the perspective of manufacturing or assembly might not be sufficient. Certain specialised analyses may be required in addition to cost, for example, to ensure compatibility between manufacturing and design choices, and assembly feasibility. Additionally, in some cases it might be useful to quantify and directly use a manufacturing consideration as a parameter in design trade-offs, instead of translating it to cost.

2.3.2. Non-generic solutions

This section will discuss some approaches that are non-generic, and are designed to work with very specific types of product designs, manufacturing processes, materials, and so on.

The first case is presented in [22], who builds upon the work of [21] that was discussed in [section 2.1](#). This study presents approaches to quantify the manufacturability factors as both manufacturing measures and indices. These were then used in the preliminary design of aircraft structures using MDAO workflows with constraints on performance, cost and manufacturability variables [7]. The link between manufacturing and design was provided through the quantified measures and indices. A tool was developed that can run in either analysis or optimisation mode, and allows fuselage and wingbox products. In the analysis mode, the system can be checked for a design's manufacturability and cost, and then revised based on the results. Whereas, in the optimisation mode, weight of the product is minimised.

A major limitation of their approach is that their system is not linked to a CAD model to derive the geometrical properties of the product. All calculations and analyses are performed based on manually defined, text based user inputs that specify the system geometry and details of its sub-components (like ribs and spars). An excerpt from the template for such an input is shown in [Figure 2.4](#). Without a visual reference of the product, it can be very difficult to check the defined inputs for their validity, and to analyse the generated results. The author mentions that a tool is being developed to extract the required geometric inputs from a CAD geometry, but it is not clear whether the tool will also enable visualisation of the results. Another limitation of this work is that assembly considerations are reduced to a single "assembly complexity" factor for the whole system, and few details are provided regarding its calculation.

Turning to a different study, a software package is presented in [9] that can automatically analyse 3D CAD models for their manufacturability. Inputs to the software are given as STL files that have the geometry. The software is supposed to have different modules for different manufacturing processes, such as casting, welding and machining. Only the machining module is discussed in the present work, with the other modules being under development. The machinability of parts is analysed in terms of four parameters: visibility, reachability, accessibility and setup complexity [9]. These are visually represented in [Figure 2.5](#).

A part with high visibility is explained to be one whose surface area is visible from the perspective of a machine tool [9]. Reachability checks if the length of the tool can reach all surfaces of the model [9]. In general, it is undesirable to use longer tools, as it requires slower feed rates to maintain dimensional conformance [9], and also carries a higher chance for tool breakage. Therefore, presence of deeper features is penalised with a weight that is based on the surface area of the feature. Next parameter is accessibility, which checks if the tool can reach all areas without collisions with the design features. Lastly, setup complexity estimates the number of setups required for machining all features of a part. A complex part with features on multiple faces would require more setups, as its orientation would have to be changed to provide tool access [9].

The software provides feedback from these analyses in two ways. The first is as coloured graphical models, that highlight areas of concern for each of the above analyses. The second result is a normalised score for each analysis. These are also aggregated into a singular manufacturability score for the design [9]. The availability of these numerical scores make the approach suitable for use in context of optimisation and/or automation workflows. Different designs can also be traded-off with this approach. Furthermore, a common concern of engineers for trusting abstract scores to judge designs is addressed to an extent, as the coloured plots provide more information about how those scores are

- ② **System Definition** (includes a computational model such as an FE mesh)
- a. wing box
 - a.1 to a.4 single, double, triple, or quad cell
 - a.5 combination (specify the number of cells in the inboard and outboard sections)
 - b. fuselage
 - b.1 to b.3 fore-section, mid / wing-section, aft-section
 - c. horizontal tail box
 - c.1 to c.3 single, double, or triple cell
 - d. vertical tail box
 - d.1 to d.3 single, double, or triple cell
- ③ **Features Identification** (assuming wing box was chosen in ②)
- Wing:
- | | |
|---|--|
| a. straight, swept forward, or swept back | [select one] |
| b. leading-edge break(s) | [0, 1, or 2] |
| c. trailing-edge break(s) | [0, 1, or 2] |
| d. chord taper | [Y or N] |
| e. thickness taper | [Y or N] |
| f. dihedral | [None, Up, Down] |
| g. geometric twist | [None, Small, Moderate, Large] |
| h. aerodynamic twist | [None (1 airfoil), 2 airfoils, 3 airfoils] |
| i. control surfaces | [Y or N] |
| flap(s) | [0, 1, 2, or 3] |
| aileron | [Inboard, Outboard, Both] |
| slat(s) | [0, 1, 2, or 3] |
| spoiler(s) | [0, 1, 2, or 3] |
| j. mechanical components | [Y or N] |
| k. fuel tank(s) | [Y or N] |

Figure 2.4: Excerpt of a template of the text based input user to define geometrical product model in [7].

related to specific design features. This is further complemented, as the logic behind the working of the different analyses is clearly presented in their publication [9].

As for the limitations associated with this software, it is currently applicable only to machined parts, although, more manufacturing processes such as welding and casting are planned to be added [9]. This would still limit the tool for analysing individual parts, as assembly processes would be left out. Even manufacturing of individual parts commonly involves multiple processes, for example machining a part after it is cast. The presented software cannot handle this analysis, as it can only consider one process at a time [9]. This limitation has been recognised by the author, and is included as future work in his work.

Lastly, the study presented in [8] that was discussed in [section 2.3.1](#), can also be mentioned in this section. To restate, their methodology had the limitation of using dimensional parameters of an OML to mathematically estimate details of the internal structure instead of deriving these details from a CAD model. This was followed by the use of a “manufacturing translator”, which was a script to generate manufacturing related data to be fed to a cost tool. The use of mathematical relations to completely define the product geometry and manufacturing related information makes their approach very specific to certain types of products and manufacturing considerations.

To conclude, the ability to easily include new manufacturing technologies, or unconventional design concepts in the conceptual design process can prove to be beneficial for both academia, and the industry. It would allow the industry to stay competitive by being able to analyse the viability of, and hence offer state-of-the-art products to their customers. And it would enable academia to inspire confidence in highly unconventional design concepts among the industry, such as the Flying-V [28], by being able to say something about the manufacturing feasibility of such designs in addition to their performance benefits.

Additionally, extendability of the approach is an important consideration, especially for production related information, as often, new manufacturing processes are developed, or added to an organisation’s capabilities. These would need to be added to the modelling and analysis capabilities of the

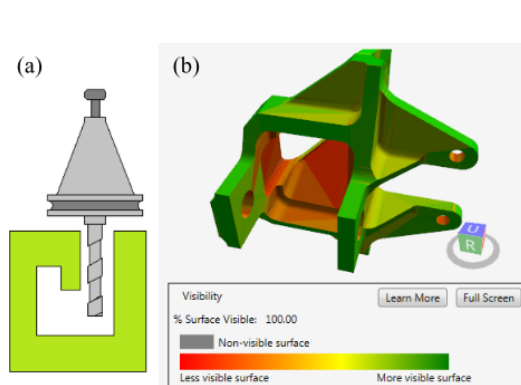


Figure 1: Visibility characteristic for machining; a) Example of problem geometry for tool line of sight. b) Colored visibility model feedback

(a) Visibility characteristic [9]

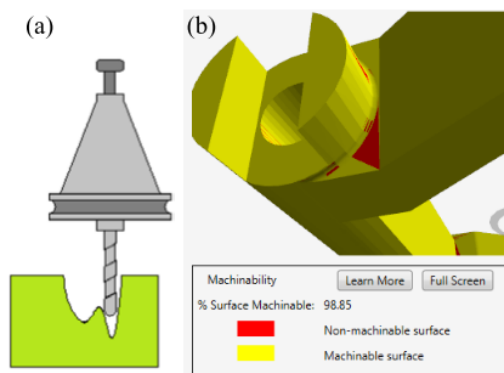


Figure 3: Accessibility characteristic for machining; a) Geometry with a tool accessibility issue, b) Color map of accessibility to facets

(c) Accessibility characteristic [9]

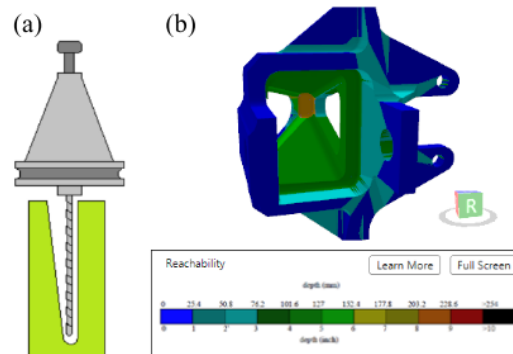


Figure 2: Reachability characteristic for machining; a) Part geometry with poor reachability, b) Color map of reachability depths

(b) Reachability characteristic [9]

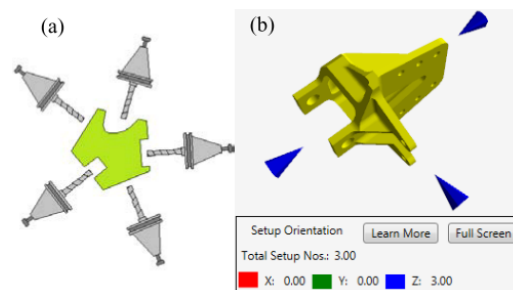


Figure 4: Setup complexity characteristic for machining; a) Geometry requiring multiple setup orientations, b) 3D representation of setup requirements

(d) Setup complexity characteristic [9]

Figure 2.5: Graphical representation of the four machinability parameters considered in [9]. Note that the original captions from [9] have been included in the figures.

organisation as well for an effective product development process.

2.3.3. Use of some specialised commercial tools

This section will consider the use of some specialised commercial tools to account for manufacturability of products.

One category of such software includes manufacturing simulation tools, such as the aPriori Manufacturing Insights Platform⁶. Such tools are capable of performing detailed manufacturing analysis of products, but often require user interaction to interpret and include results in the design. This can be seen from steps 1, 5 and 6 in Figure 2.6.

On the other hand, there are some studies that make use of product lifecycle management (PLM) tools to model the manufacturing systems with the product design [11, 12]. But PLM systems are not suitable for use at the conceptual design stage, as they depend on having a coherent product topology definition to model information effectively [29]. This is not possible at the conceptual design stage, where significant changes can be made to the product definition frequently.

The next discussion is of the work in [10], where the authors extended the work presented in [8]

⁶<https://www.apriori.com/manufacturing-insights-platform/> (accessed 20 August 2022)

⁷<https://www.apriori.com/blog/apriori-22-1-expands-manufacturing-insights-cost-savings-analysis/> (accessed 10 August 2022)

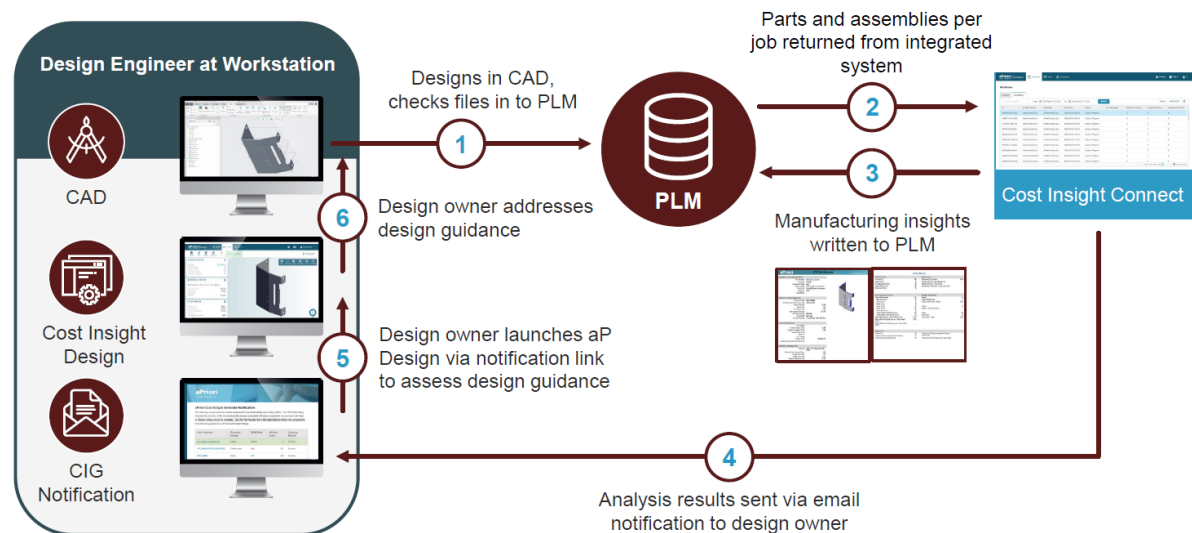


Figure 2.6: Suggested workflow using the aPriori Manufacturing Insights Platform⁷.

(discussed in section 2.3.1). The main addition by [10] is the inclusion of production rate and process planning analysis in addition to manufacturing cost. This was accomplished by integrating their methodology with a commercial production planning software, Simio⁸. In their workflow, SEER-MFG is first used to calculate the manufacturing process times. This information becomes an input for the production planning model to define the process time at each station, and is complemented by some additional user inputs, such as the number of workers, shifts, workstations etc.

An extensive use of commercial programmes enabled these authors to perform detailed analyses at early stages of design in an automation/optimisation context. As a result, their methodology offers various parameters that are relevant for production and economic considerations of designs, in addition to the conventional performance indicators. A complete list of these factors from [10] is shown in Figure 2.7. The availability of these parameters enabled them to perform optimisation studies with production based design variables and constraints. Additionally, the overall steps for assembly were also considered in their production planning process.

Production	Economics
Production rate	Capital cost
Workstation utilization rate	Tooling and machine cost
Component flow time	Material cost
Component waiting time	Labor cost
Component processing time	Overhead cost
Technician utilization	Return on investment
Yearly Backlog	Cumulative cash flow
Excess Inventory	Breakeven year

Figure 2.7: List of production and economics related parameters available in the study of [10].

Despite these benefits of these commercial software, such tools are still limited by the available options in their databases. To work with the latest manufacturing technologies, users are dependent on the software provider to include them in the database. Furthermore, this may even not be possible in some cases, for example, if a company has developed a proprietary manufacturing process, and do

⁸<https://www.simio.com/> (accessed 09 August 2022)

not wish to share that information. The open source cost tool [24] is one software overcomes the last limitation by allowing users to define their own manufacturing methods with the help of a database that holds various manufacturing steps, which can be used to define manufacturing methods.

2.4. Assembly sequence representation

Graphical methods are common for representing assembly sequence information as they offer an intuitive way to represent this data, and they allow the modelling of necessary assembly information, such as connections and sequence of operations [30]. This section will introduce some of the existing methods for the graphical representation of assembly sequence. An extensive review of such methods is presented in [30], and will be used here to introduce these topics.

As per [30], there are four main phases in assembly sequence planning, and each utilises a different graphical representation. These phases are summarised below:

1. All possible sets of assembly sequence are obtained in the first phase. For this purpose, AND/OR graph or directed graph representations can be used [30].
2. In the second phase, assembly sequences are chosen from the set of all possible sequences based on the connections that exist between components. Liaison graph are commonly used for this purpose [30].
3. The number of assembly sequences are reduced in the third phase by querying on the sequences for the right order.
4. In the fourth phase, feasible assembly sequences can be represented by a precedence graph.

Next, the different kinds of graphs introduced above will be explained.

2.4.1. AND/OR graph

This type of graph represents all possible assembly sequences into account. The nodes in this graph represent product components, which can be a part, subassembly or the final product (after completion of assembly). The edges represent connections between the components [30]. An example of this graph is shown in Figure 2.8.

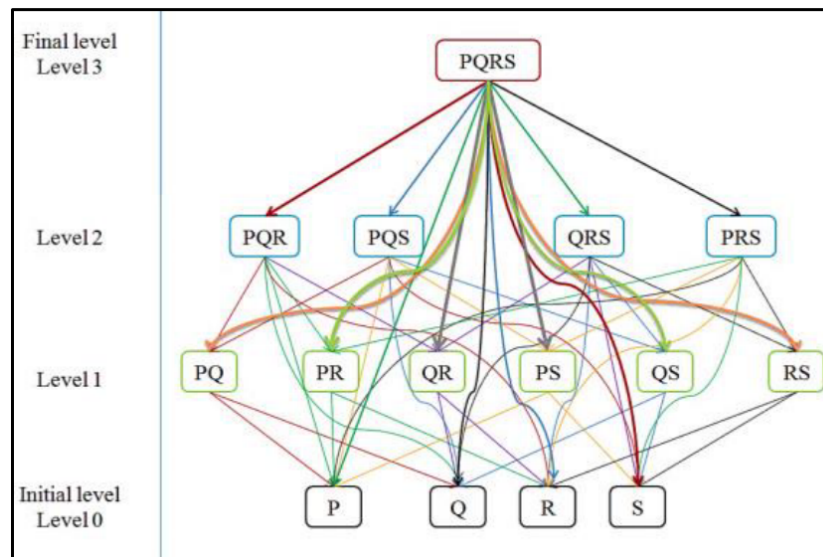


Figure 2.8: AND/OR graph representation for the assembly of a product with four parts, represented by P, Q, R, S. The graph represents all possible combinations of assembly sequences. Source: [30]

2.4.2. Directed graph

This is similar to the AND/OR graph, but only one intermediate level is used to reach the final assembly in its representation [30]. An example of this graph is shown in Figure 2.9. Groups of components are represented by putting them in the same bracket.

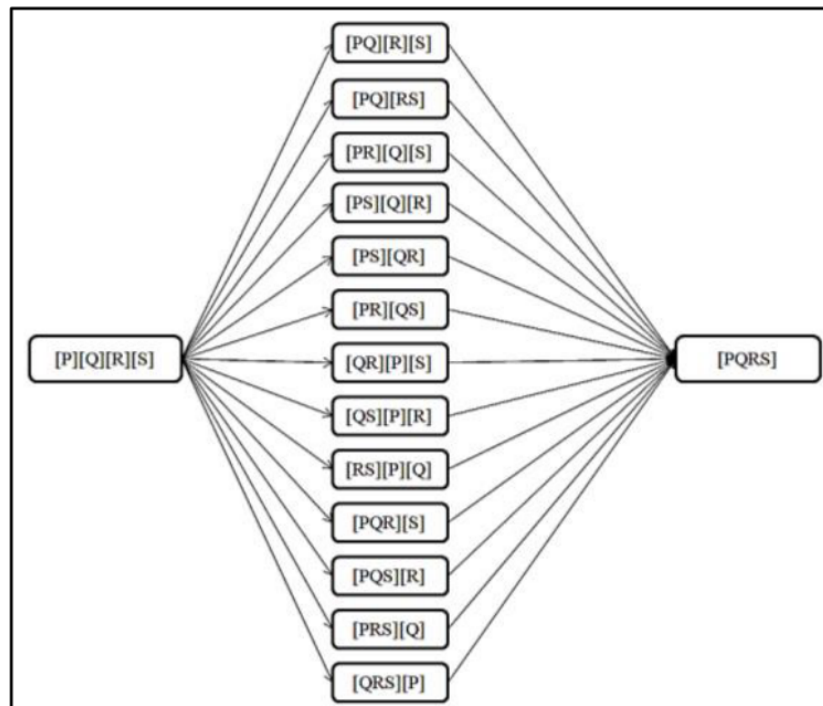
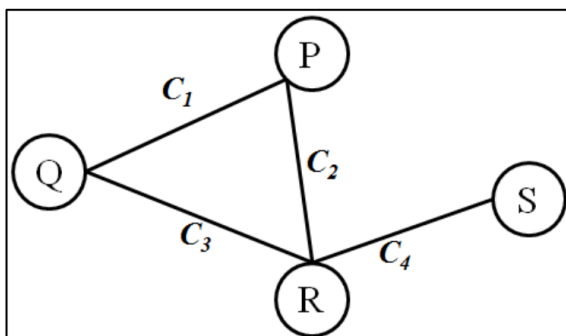


Figure 2.9: Directed graph representation for the assembly of a product with four parts, represented by P, Q, R, S. The graph represents all possible combinations of assembly sequences. Source: [30]

2.4.3. Liaison diagram

A liaison is defined as, "a connection established between components [30]". In a liaison diagram, liaisons are represented between pairs of components, to represent significant relationships in a product [30]. Therefore, this graph also represents components as nodes, and connections between components as edges. This graph can also be represented by a liaison matrix. An example of a liaison graph and matrix is shown in Figure 2.10.



(a) Liaison diagram for the assembly of a product with four parts, represented by P, Q, R, S. Edges represent connections between the parts. Source: [30]

	P	Q	R	S
P	0	1	1	0
Q	1	0	1	0
R	1	1	0	1
S	0	0	1	0

(b) Liaison matrix for the assembly of a product with four parts, represented by P, Q, R, S. Source: [30]

Figure 2.10: Liaison diagram and matrix for the assembly of a product with four parts, represented by P, Q, R, S. Source: [30]

2.4.4. Precedence diagram

This diagram is used to represent assembly tasks. These tasks are represented as nodes, and dependencies between them are represented by lines [30]. A separate precedence diagram is created for each subassembly in a product. An example of this diagram is shown in Figure 2.11.

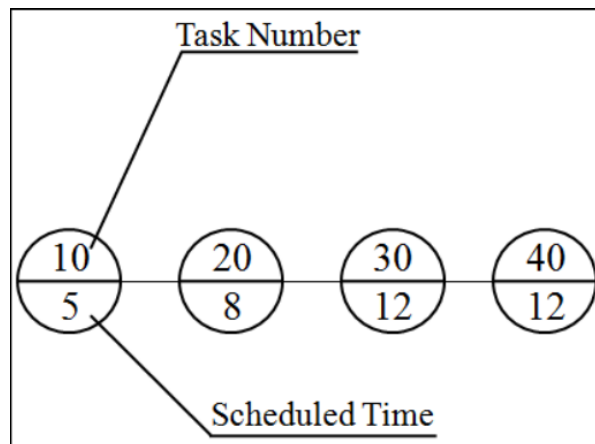


Figure 2.11: Precedence diagram for an assembly with four tasks. Source: [30]

In conclusion, this chapter presented some state-of-the-art approaches for including production considerations in conceptual designs. Limitations of both the manual and the automated approaches were identified. It can be noted that very few cases consider manufacturing and assembly at the same time, and those that do, also have their own limitations. An overview of the various production considerations in design was given, and different methods for the representation of assembly sequence were discussed.

3

The Manufacturing Information Model

The methodology proposed in this thesis for the automated inclusion of production considerations in design is focused around a model that has been developed to manage all production related information for a product. This model will help in formalising the information that is usually stored in documents, or exists only with engineers as experience. This in turn will allow the effective re-use of this information by other disciplinary models and analysis tools that are connected to the product model. This chapter will discuss details of this newly developed model.

The overall model is referred to as the *Manufacturing Information Model (MIM)*, and it is a package that consists of three subpackages. These include the manufacturing model, database, and the assembly model. Their Unified Modeling Language (UML) representation has been shown in [Figure 3.1](#). Some dependencies between the subpackages can be observed in this figure, and these will be explained in the next sections, along with a detailed description of each subpackage.

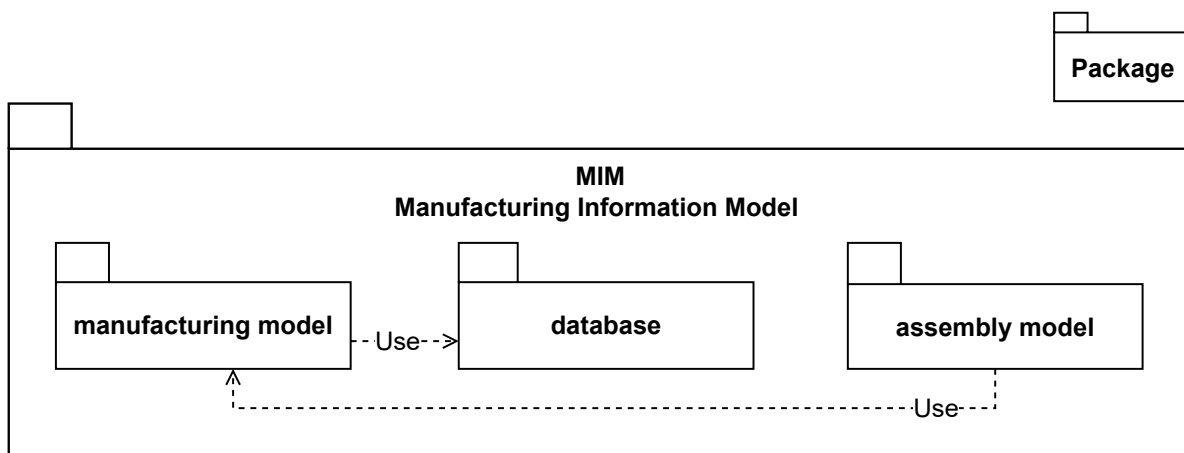


Figure 3.1: UML diagram of the developed “manufacturing information model (MIM)” to manage production information of products.

3.1. Manufacturing Model

The main role of the manufacturing model is to capture production related information for each component of a product in a structured manner, and to allow its re-use in the system design process. A better term for this product *component* is a product *primitive*, which is defined as, “a *parametric building block to define the product*” [26]. Therefore, a product primitive can not only be a physical

part, but it can also represent more abstract objects such as joints, that still play an important role in defining the product. For the purposes of this thesis, the given definition of a primitive is further expanded, and split into two types: *functional* and *manufactured* primitives. A functional primitive is defined as, “a constituent of a product, identified based on its specific functions in the product”. For example, this can be a skin panel, stringer or a joint between the two. Specific functions for each of these can be identified. Skin panels provide the necessary aerodynamic surface, and help transfer load to the underlying structure. Stringers improve stiffness of the skin panels, which in turn is dependent on the joint between the two.

On the other hand, a manufactured primitive is defined as, “a constituent of a product, identified based on what object is manufactured”. Based on this, a manufactured primitive can be same as an individual functional primitive, such as a spar or a rib. However, a manufactured primitive can also be a composition of multiple functional and/or manufactured primitives. For example, an integrally stiffened skin panel is manufactured as an individual part [31], but it is composed of two functional primitives; the skin panel and the stringer.

Additionally, a functional primitive can be composed of a set of manufactured primitives, as in the case of a built-up rib. The rib itself is the functional primitive, which in this case is composed of a separate web and flanges (the manufactured primitives), which are joined together to make the complete rib. These joints would classify as both functional and manufactured primitives.

Functional primitive

A constituent of a product, identified based on its specific functions in the product.

Manufactured primitive

A constituent of a product, identified based on what object is manufactured.

The main goal behind defining manufactured primitives is to identify exactly what object will be manufactured, as it is for these manufactured primitives that the manufacturing model is defined. As the manufactured primitives represent the design, the manufacturing model can be directly linked to the primitive's design as well. And, as each manufactured primitive can be manufactured in a unique manner, a separate manufacturing model definition is used for each primitive. This is shown in Figure 3.2.

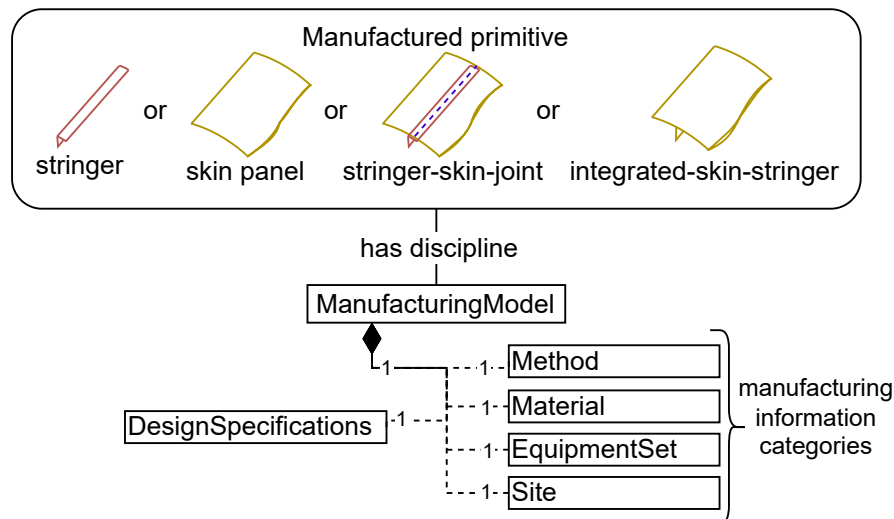


Figure 3.2: Relation between the manufacturing model and manufactured primitives of a product, with an overview of the manufacturing model information categories.

Based on the discussion so far, the manufacturing model can be described in a single statement as, “a disciplinary view of a manufactured primitive in a product that complements its design specification to capture production related aspects of the primitive”. An overview of the structure

of the manufacturing model can also be seen in [Figure 3.2](#). It shows that the manufacturing model is a composition of five class objects. These are called as the *information categories* of the manufacturing model, and are used to capture production information in an organised manner. Four out of the five categories capture manufacturing related information, and hence are labelled as the “manufacutring information categories” in [Figure 3.2](#). The fifth category is design specifications, and it holds information that is relevant for manufacturing and has to be derived from the manufactured primitive’s geometric design. For example, this can include the joint length for a joint primitive, overall bounding box dimensions for a machined part, average thickness for a part, etc.

Next, the structure of the manufacturing model will be discussed in more detail by considering an example of a manufactured primitive, an L-stringer. The UML class diagram of this example is shown in [Figure 3.3](#). Note that only relevant attributes are shown for clarity. The LStringer class can be seen on the top left, which is the manufactured primitive being considered, and represents the design of the L-stringer. Specifications of the L-stringer’s design are based on the product’s design definition model, which is defined below.

Manufacturing model

A disciplinary view of a manufactured primitive in a product that complements its design specification to capture production related aspects of the primitive.

Design definition model

A product model with primitives arranged per a defined topology. It defines what primitives are present in the product, and their specifications (e.g. dimensions, location, quantity, type etc.).

To add the manufacturing model to the L-stringer, a specialisation of the base class `ManufacturingModel` is created for it (class `LStringerManufacturingModel`). Similar specialisation classes can be created for each manufactured primitive in a product. Three things are defined differently in each manufacturing model specialisation based on the manufactured primitive.

These are shown in pink text in [Figure 3.3](#), and are explained below:

- **Attribute `valid_material`**
This is a list of materials that are suitable (hence “valid”) for the design of the manufactured primitive, which in this case is the L-stringer.
- **Attribute `valid_mfg_method`**
This is a list of valid manufacturing methods which can be used to manufacture the manufactured primitive.
- **Class `LStringerManufacturingDesignSpecs`**
The design specifications information category of the manufacturing model requires a specialisation to be created of the base design specifications class `ManufacturingDesignSpecs` for each manufactured primitive. This is necessary as the rules to derive the design specifications information from the manufactured primitive’s design model will differ for each primitive. For example, the reference length of a stringer maybe defined along its central guide curve, whereas the reference length of a skin panel maybe along a particular coordinate system axis. Hence, these primitive specific classes allow for the these differing rules. The dependency link between `LStringerManufacturingDesignSpecs` and `LStringer` in [Figure 3.3](#) represents derivation of the required information from the design model.

The next four information categories deal specifically with production related data, and are hence called as the *manufacturing information categories*. These are the manufacturing method, material, equipment set, and manufacturing site categories, and are shown in purple in [Figure 3.3](#). Unlike the design specifications information category, these four categories only contain production related information about the primitive. Therefore, specialisations of the category base classes do not have to be created for each manufactured primitive.

A brief description of each manufacturing information category is provided below:

- **Class ManufacturingMethod**

The manufacturing method category holds information about what manufacturing method will be used to produce the primitive. This is hand layup in the considered example, which can be seen in the blue comment on the right side. For a joint primitive, this can be an appropriate joining method, such as bonding, or mechanical fasteners. Details about the format and name of the corresponding attribute, `selected_dicts_list`, will be discussed in the paragraph that follows after this list.

- **Class ManufacturingMaterial**

The material category holds information about the material specifications of the primitive. For a part, this would include information about the constituent materials of the part, and for a joint primitive, this would include material of the joint, if applicable. For example, it can be the material(s) of a fastener system in a mechanical joint (which can include a bolt, nut, washers etc.), or can `null` for induction welded joints.

- **Class ManufacturingEquipmentSet**

Equipment set

A set of equipment required for the completion of all steps of a manufacturing method.

The definition of an equipment set is given in the above box. This concept has been introduced because a complete manufacturing method (which can have many steps that use different equipment) cannot be associated with just one equipment. The complete relation between a manufacturing method, equipment set and equipment is shown as a UML diagram in [Figure 3.4](#). Some important aspects shown in this figure are summarised below:

- A manufacturing method can be completed using one or more equipment sets. For example, the same manufacturing method may use slightly different equipment at two different manufacturing sites.
- An equipment set can only be used for one manufacturing method. This is because an equipment set is defined with a unique set of equipment that will fulfil a given manufacturing method. It is to be noted this would in general hold true, but can technically be violated in an unlikely scenario where two different manufacturing methods use the exact same set of equipment. The author did not come across such an example.
- An equipment set can be associated with one or more equipment. This is possible by definition of an equipment set.
- An equipment can be associated with one or more equipment sets. For example, an autoclave is used in the manufacture of composite parts, and for the metal co-bonding. These two manufacturing methods will have two different equipment sets, but autoclave will be one of the common equipment in these equipment sets.

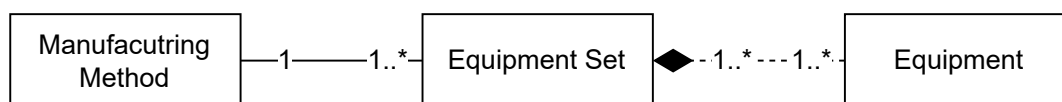


Figure 3.4: UML diagram showing the relation between a manufacturing method, equipment set, and equipment.

- **Class ManufacturingSite**

The last category is manufacturing site, which defines at what location will the manufacturing operation take place. In a lot of cases, the manufacturing capabilities of a company are spread across different manufacturing sites. This category accounts for that.

Each of the manufacturing information categories has three main slots. These can be seen in the manufacturing method category class in [Figure 3.3](#). Note that they are not shown for the other manufacturing information categories due to space constraints. Each of these is described below:

- **Attribute `selected_dicts_list`**

The `selected_dicts_list` attribute is a list of (Python¹) dictionaries, and holds information for the selection made in that category. A selection is made by first selecting a library, and then selecting certain item(s) from that library. For example, the LStringer will be manufactured with the “hand_layup” method (item) from the “mfg_method_lib” (library). These names for the library and the item are their unique identifiers (UIDs), that are defined in the MIM database. More information on the database and libraries will follow in [section 3.2](#). Furthermore, to make selections, one dictionary is defined in the list for each library that has to be accessed. Then one or more items from that library can be selected by adding their UIDs in the `uid_list` key.

Even though multiple selections can be made using this format, they are only allowed for the material information category. This is because multiple selections for the manufacturing method (and hence equipment set as well) would not make sense for a given primitive. They are allowed for material, as part and joint primitives can have multiple specifications of material per primitive. For example, a part can have different material zones with varying thicknesses and ply stacks, and a joint with a fastener system can have different materials for the bolt, nut and washers. Lastly, multiple selections for manufacturing site is currently not allowed as there are some additional consequences of such a selection, and these are not included with the scope of this thesis because of time limitations. For example, defining the production of a part over multiple manufacturing sites would mean that the time to transfer the part between these sites would also have to be considered if the production rate has to be calculated. Even though multiple selections can only be made for the material category, the selection format (list of dictionaries) has been kept the same for all manufacturing information categories for consistency.

Lastly, there are two ways to make these selections. The first involves a direct specification by the user, and second option derives this information from the design definition model, provided that the required information is available. For example, if the design model already contains information about the selected material for the primitive, then the material information category can derive this information from the design model by using some rules. This is also shown in [Figure 3.3](#) through the optional dependency relations from the manufacturing information categories to class LStringer. By using this option, duplicate or conflicting information between the design and manufacturing model can be avoided. An example for the user input through which these selections are made is provided in [appendix C.1](#).

- **Input `valid_dicts_list_from_drivers`**

The order in which selections are made for each information category matters, as it decides what *valid* options remain for the categories that follow the selection. For example, if the method is selected first as hand layup, then only composite materials are a valid choice. Similarly, if the material was given a higher priority, and a metal was selected, then hand layup would no longer be a valid manufacturing method. This also applies to the equipment set and manufacturing site information categories. Therefore, if the equipment set is to be selected after manufacturing method and material, then the valid options for it would depend on the selected manufacturing method and material. This has been shown with the orange dependency links in [Figure 3.3](#). It is to be noted that these dependency relations between the manufacturing information categories are dynamic, and the ones shown in this figure can change based on the order in which selections are made.

The `valid_dicts_list_from_drivers` input slot holds the valid options that are available in each manufacturing information category, based on the selections that have been made in the categories before it. The term *drivers* in its name refers to it being *driven* by selections in other information categories. The order of selection is controlled by user input, which is an ordered list of UIDs that identify the manufacturing information categories. An example of a user input with this list can be seen in [appendix C.1](#), where the list is assigned to the `info_category_order` key.

Lastly, the value of `valid_dicts_list_from_drivers` is derived automatically with the help

¹<https://www.python.org/> (accessed 29 September 2022)

of the database subpackage. For this, values of the `selected_dicts_list` attributes of the categories on which a given category depends are given as an input. This is also the main dependency of the manufacturing model on the database that was previously shown in Figure 3.1. The rules for deriving the value for this input slot are located at a higher level than the manufacturing information categories in Figure 3.3. Hence, these rules are located in class `ManufacturingModel` (or its specialisations). This allows easy access to the `selected_dicts_list` attributes of each information category, and avoids any potential dependency loops between the information categories.

Finally, once the value of `valid_dicts_list_from_drivers` is available, it can be checked against the `selected_dicts_list` attribute to ensure that the selections are valid. This ensures a valid definition of the manufacturing model. The validity of the selections can be checked in each information category with another attribute with a Boolean value (not shown in Figure 3.3). In case a selection is invalid, warnings are printed in a window that indicate what incorrect selection was made, and what the valid options are. An invalid selection can occur even if the selection was derived from the design model, as this would just indicate that the value from the input specification (also a user input) to the design model is not a valid selection based on the defined dependency links between the manufacturing information categories.

- **Attribute specifications**

This attribute is a pointer to an object in the MIM database for the selected item(s) in each manufacturing information category. The object to which it points contains detailed information about the selected items. For example, for the material category, the specifications would be the material properties. For the method category, this would include any specific details related to the selected manufacturing method, e.g. optimum cutter radius sizes for machining. For the manufacturing site category, some of the details include the site's location and available equipment at the site. Lastly, for the equipment set, the specifications would include details of each equipment that makes up the equipment set.

The last remaining aspect shown in Figure 3.3 is mappings of selections made in the manufacturing information categories to analysis tools and other disciplinary models connected to the product model. This allows the manufacturing model, and these other tools to use their own UIDs for the various options for manufacturing methods, materials, equipment and site; while establishing a mapping that will help in accessing the right information from the manufacturing model by these other tools. An example is shown for this in Figure 3.3 for the manufacturing site category. A separate "Mapping" class exists for each manufacturing information category, which is a component of the category. This is the class `MFGSiteMapping` in Figure 3.3. Note that this has only been shown for the manufacturing site category in the figure to limit the figure size for clarity. The `open_source_cost_environment` attribute in this mapping class holds the name of the "manufacturing environment" that is associated with the selected manufacturing site for the open source cost tool [24]. A manufacturing environment in the open source tool is the equivalent of a manufacturing site in the manufacturing model.

Similarly, additional attributes can be added to the mapping class of each manufacturing information category for other analysis tools. The values for these attributes are provided by the MIM database, where these mappings are defined by the user.

3.2. Database

The database is a supporting subpackage in the manufacturing information model, and provides the manufacturing model with the necessary data for its information categories, along with some methods to access this information effectively. The structure of the database package can be seen in Figure 3.5. It consists of a *data* directory that holds all relevant data files, and a *libraries* subpackage, that holds methods and modules to help access information from the data files.

3.2.1. Data directory

The data directory holds all reference data that would be needed by the manufacturing information categories of the manufacturing model. All data is stored in the JavaScript Object Notation (JSON)² file format. This format was selected because of the following advantages:

²<https://www.json.org/json-en.html> (accessed 14 August 2022)

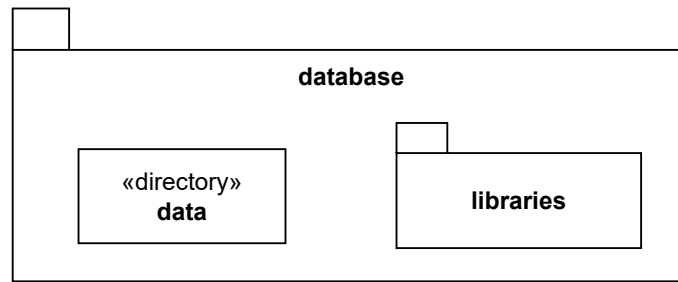


Figure 3.5: Structure of the database package showing the data directory and the libraries subpackage.

- It is a human-readable format
- It is easy to modify or add information to existing files either manually or through programmed scripts.
- The manufacturing information model is implemented using the Python programming language, which would make working with JSON files straightforward, as they can be directly parsed as Python dictionaries. More details on selecting Python as the programming language for implementation of MIM will follow in [section 4.1](#).

Data for each manufacturing information category is grouped together in separate JSON files. Therefore, at least four data files are required for the complete definition of the manufacturing model. It is to be noted that any information that is specific to a product primitive is not defined in the database. Hence, the same database can be used for different products.

A brief overview of the information stored in the data files for each manufacturing information category is given below:

- **Material data files**

Three things are defined in the material data files. These are as follows:

- The first is a mapping between the material UUIDs used in the database and the material names used by other disciplines or analysis tools in the product model (such as cost analysis). Details of this mapping has been previously discussed in [section 3.1](#) with an example for the manufacturing site.
- The second is a mapping to define material families. A material family is defined as, “*a set of materials grouped together for convenient reference*”. An example of this is the aluminium 2000 series alloys. It can be useful to refer to material families over specific materials, when the properties being considered do not change from one material to another in the same category. For example, a carbon fibre part cannot be in contact with an aluminium alloy part, as it will lead to galvanic corrosion in the aluminium part. Therefore, it is easier to refer to the material family of all aluminium materials and carbon fibre materials to specify this condition as a rule in the system. An example of where this rule is defined will be given in the next point.

Material family

A set of materials grouped together for convenient reference.

- The last specification in the material data file is of criteria for the material-material compatibility analysis. For example, this is where a criteria can be defined for the galvanic corrosion compatibility rule discussed in the previous point.

An excerpt from a material data file with examples for the three types of material information discussed above is available in [appendix C.2.1](#) for reference.

The last point to note is material properties (such as related to strength) are not defined in the material data file, and can be managed with another database of user’s choice. The material UUIDs specified in the material data files would have to reflect the corresponding names in the

selected database. Within the scope of this research, a material database developed in-house at GKN Fokker Aerostructures has been used for this purpose.

- **Manufacturing method data files**

The data file for manufacturing methods consists of a listing of the available manufacturing methods, along with their specifications. The method is assigned a type, which can be either “part” or “joint”, for part and joint manufactured primitives respectively. Valid materials for the manufacturing method, and mappings to identify the method in other disciplinary models or analysis tools is also specified. An example for such an entry is provided in [appendix C.2.2](#).

- **Equipment data files**

The data file for equipment defines two kinds of information. The first is a list of equipment, and the second is a list of equipment sets. Each equipment entry includes technical information about the equipment, valid materials that the equipment can be used with, and also some other data such as operating cost. On the other hand, each equipment set entry specifies what equipment is included in that set and to what manufacturing method is the equipment set associated to. An excerpt from an equipment data file is provided in [appendix C.2.3](#).

- **Manufacturing site data files**

These files have a list of specifications for the available manufacturing sites. Each entry includes the location of the site, what equipment is available at the site, and a mapping to identify the site name in other disciplinary models or analysis tools. An example of a manufacturing site entry can be seen in [appendix C.2.4](#).

3.2.2. Libraries subpackage

The main role of the libraries is to process and help access information stored in the data directory. After this data is parsed, it results in a tree structure with the “library category” at the first level, one or more libraries in each library category at the second level, which is followed by one or more items in each library at the third level. This can be seen in [Figure 3.6](#), which shows the tree structure in a graphical user interface (GUI). More details about the software used for the GUI will be discussed in [section 4.1](#). Based on this structure, it is possible to have multiple libraries under each library category. Therefore, data for each manufacturing information category can be split into multiple files for better organisation. Each such data file would then become a library under the appropriate library category.

As a part of its other functions, the libraries subpackage establishes valid relations between items of the information categories, and provides methods to perform information queries to extract such data. From the previous discussions on the manufacturing model and data directory, it has been established that the user specifies valid items from the four manufacturing information categories as attributes and lists in the manufacturing model and data directory. It must be noted that not every valid relation between each information category is defined by the user. From the definition of some relations, the rest can be automatically derived through programmed rules in the libraries subpackage.

[Figure 3.7](#) shows which relations have to be user defined, and which are derived. The four manufacturing information categories can be seen at the top in purple in this figure. Design is also shown at this level, but its relations are not defined in the database, and are instead defined in the primitive

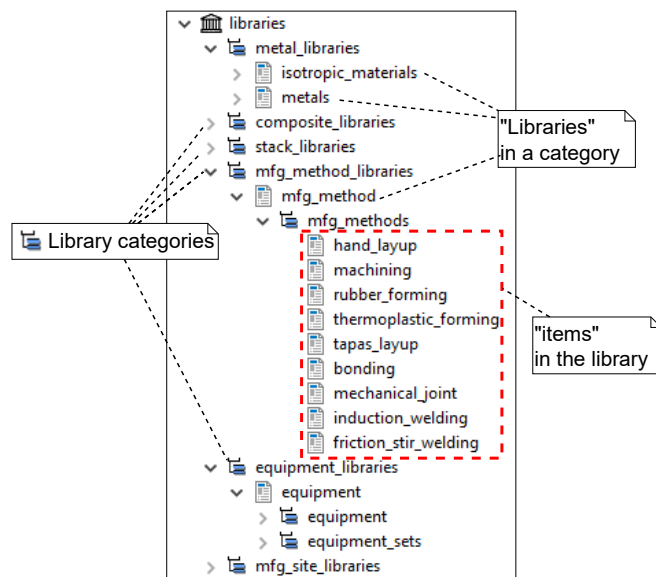


Figure 3.6: Screenshot showing the implemented result of the libraries subpackage.

specific manufacturing model classes. The derived relations (green) can depend on both, user defined relations (blue or yellow), or other derived relations. But ultimately, all derived relations can be traced back to a user defined relation.

The derived relations can be understood better with an example. Consider the “Method” category. If it is desired to find the `valid_equipment_sets` for a given method, say machining, then the libraries subpackage would look at all available equipment set definitions in the data directory, and identify the equipment sets that are associated with the machining manufacturing method. This is shown with a dependency link from `valid_equipment_sets` under the Method category, to `valid_method` under the Equipment set category. And as discussed in the previous section, “`valid_method`” is defined by the user for each equipment set in the data files.

Lastly, on the basis of these established valid relations, the libraries subpackage offers methods to access valid data for each of the manufacturing information categories, based on some given data for the other categories. An example of such an information query method can be seen in Figure 3.8 for the manufacturing method category. The default input for the other information categories is `None`. Therefore, without inputs, these methods would return all possible items in the category for which the query was made (so all possible manufacturing methods in this case).

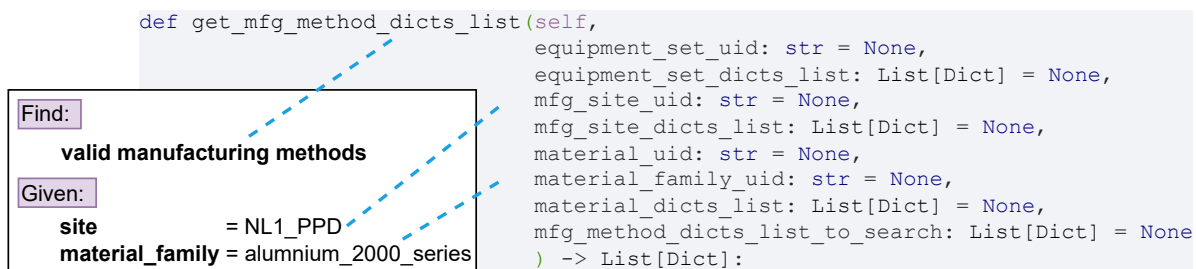


Figure 3.8: An example of an information query method in the libraries subpackage for the manufacturing methods category.

3.3. Assembly Model

The assembly model is responsible for capturing information about the sequence of steps for manufacturing the complete product. The starting point for modelling this information is to identify manufacturing steps for the product without consideration for the sequence in which they might occur. These steps can be identified through the manufactured primitives of a product, as each such primitive would be manufactured, and hence, can be associated to a manufacturing step. And since a manufacturing model is defined for each manufactured primitive, the assembly model identifies those primitives in the product model, for which a manufacturing model exists. This is the reason for the dependency link between the assembly and manufacturing model that was previously shown in Figure 3.1.

Although the process of identifying manufactured primitives with a manufacturing model in a product may seem straightforward, it is not the case in reality. This is because a manufactured primitive can be composed of multiple functional/manufactured primitives. Consider the example of an integrally stiffened skin panel [31], which replaces a skin panel, some stringers, and the joints between the two in a product model. There are a number of consequences of this integration:

- First, the individual part primitives that make up the integrated part, and the joints between them would no longer be produced in reality, and hence they should not be included in the identified manufacturing steps.
- The second consequence is for the joints that may have existed between any of the individual parts that were integrated, and other parts in the model. These joints should be redefined so that they exist between the new integrated part, and those other parts in the model. For example, joints between the skin panel and ribs may exist in the product model, but in consideration of the integrated part, these joints should now be between the integrally stiffened skin panel and ribs. Hence, these new joints must be identified in the manufacturing steps.

In the presented assembly model, these manufacturing steps are called as “operations”. Another relevant term in the assembly process is the concept of an assembly station. These two terms are defined below:

Operation

Execution of manufacturing process(es) that result in the materialisation of a manufactured primitive.

Assembly station

A physical location where a set of manufacturing operations take place, whose result (part/sub-assemblies) is either passed on to another assembly station, or is the final product itself.

With the definition of these two terms, the approach used to model information in the assembly model will be discussed. In the presented research, assembly sequence information of a product would be modelled using directed acyclic graphs (DAGs). In contrast to the approaches presented in [chapter 2](#), the nodes of these graphs would represent the operations, and the graph edges would represent the order of execution of these operations. It is to be noted that an operation can be defined for both part and joint primitives, and therefore connections between product components are also modelled as nodes with this approach. The benefit of this approach is that the assembly sequence information is modelled in a single type of graph, and there is no need to use the different kinds of graphs presented in [chapter 2](#). The feasibility of the assembly sequence can be ensured using some knowledge rules and checks that are applied during the creation of the graph. A detailed discussion of these will follow later.

The presented approach uses two DAGs to completely represent the assembly process information. These are named as the *operation graph* and *station graph*, and will be discussed in detail in the next sections.

3.3.1. The operation graph

The operation graph is defined on the right, and a simplified example of an operation graph is shown in [Figure 3.9](#). In addition to the operations and their results, a *product start* and *product end* node can be seen in this figure. These represent the start and end of all steps for producing the product. It can be observed that the nodes are connected to each other with directional edges, which represent the sequence in which these operations will be executed. The overall process starts at the product start node, then “stringer1” and “skin_panel_1” parts are manufactured at the same time, in parallel, at stations 1 and 2. Both of these part operations are then connected to their respective result node. A result node represents the result of a completed operation, which in this case would be the produced parts. The two part results are then connected to “operation: joint 1”, which leads to “result: joint 1”. This result represents the two parts, and the joint between them.

Operation graph

A directed acyclic graph with nodes as manufacturing operations and their results, and edges representing the sequence order from one operation to the next.

Therefore, as one progresses through the graph, result nodes accumulate results of previous operations. Hence, the result nodes represent subassemblies in different states of progress throughout the operation graph. Finally, the process continues on with two additional joints, and ultimately connects to the product end node, which represents completion of the product. DAGs have been specifically chosen to represent these graphs, as each operation node is unique, and would never point to itself, either directly or indirectly, once the operation is complete. Therefore, a valid operation graph would have no cycles or self loops.

The complete assembly process is represented by operation graph 3 (refer comment at top right), which includes all operations and results. Since, by definition, an operation graph can refer to any set of operations and their results, various other operation graphs can also be identified in operation graph 3. The simplest of these is operation graph 1, which includes just one operation and its result. Operation graph 2 is another valid operation graph, that has two sequential operations. Similarly, other operation graphs can also be identified in the figure, even those that include multiple assembly stations.

Furthermore, each node and edge in a graph can store information as attributes. For an operation node, some of these attributes include name of the assembly station where they will be executed

and execution time of the operation. Based on this information, nodes have been grouped by dashed orange boxes in Figure 3.9, representing at which station they are located. For a result node, important attributes include a list of parts and joints that are accumulated in the result, and also a list of parts and joints that are *required* in the result. The former is determined from the edge connections in the

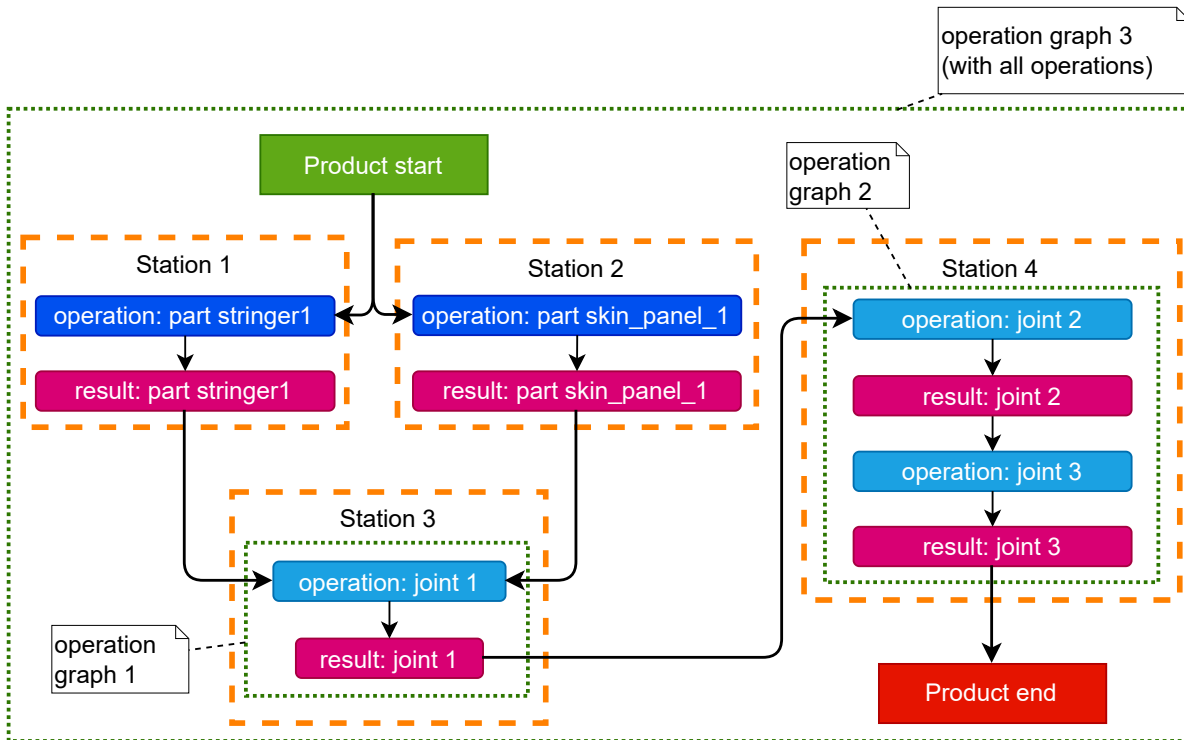


Figure 3.9: A simplified version of the operation graph to highlight some of its key aspects. Assembly stations are represented by the dashed orange boxes. Part and joint operation nodes are in dark/light blue, and result nodes are shown in light maroon. The overall product start and end nodes can also be seen. Some of the possible operation graphs are marked as 1, 2 and 3 with notes, but others can also be identified.

operation graph, but the latter is determined from the operation node that corresponds to the result node. For example, in node "result: part stringer1", the `required_parts` attribute would be `stringer1`, as the result node represents the operation node "operation: part stringer1". Similarly, for the node "result: joint 1", `required_parts` attribute would include the parts involved in the joint operation, as defined by the node "operation: joint 1", and the `required_joints` attribute would include the joint in the operation node. The availability of these required parts and required joints attributes in the result nodes allows for a check to ensure that connections have been modelled correctly in the operation graph. This is accomplished by verifying that all required parts/joints at a result node are also in the list of accumulated parts/joints.

3.3.2. The station graph

The station graph is defined on the right. It is automatically derived from the operation graph that consists of all operations for the complete product. To be precise, it is a quotient graph [32], where the station nodes are derived by merging nodes in the operation graph that belong to the same assembly station. Figure 3.10 represents the station graph that has been derived from operation graph 3 in Figure 3.9. Each station node includes an attribute for the critical time, which is the minimum time required for the completion of all operations at the station. Using this information, the critical manufacturing path through the station graph can be determined. This path represents the

Station graph

A directed acyclic graph with nodes as assembly stations, and edges representing the flow of subassemblies from one station to the next.

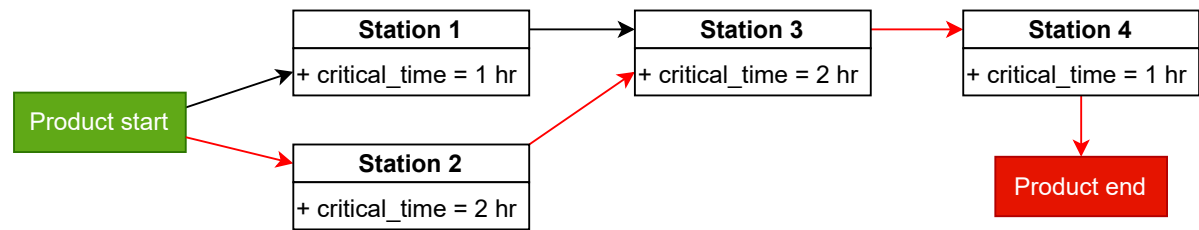


Figure 3.10: Simplified station graph with the critical manufacturing path highlighted by red edges. The critical path is calculated based on the `critical_time` attribute of each station node.

minimum time required for the complete production of the product, and therefore it is the path with the highest total time from the product start node to the product end. Once the minimum time to produce one product is known, the production rate can be easily calculated.

3.3.3. Integration of assembly model with a product

This section will discuss how the assembly model can be integrated with a product model. Two main aspects are relevant to understand this: steps involved in setting up the assembly model to get the operation and station graphs, and the final structure of the assembly model once integrated into the product.

Steps for setting up the assembly model

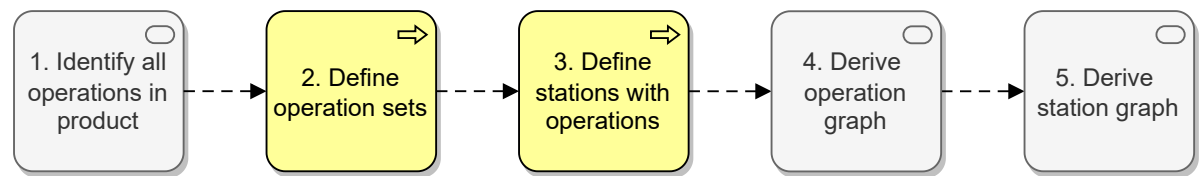


Figure 3.11: Process diagram showing the steps involved in setting up the assembly model.

The main steps in setting up the assembly model have been summarised as a process diagram in [Figure 3.11](#). Each of these steps will be discussed in detail below:

- **Step 1**

The first step to identify all operations is an automatic step, and is carried out by identifying all manufactured primitives with a manufacturing model, while taking integrated parts into account. This has been previously discussed in detail in the beginning of [section 3.3](#).

Next, two main pieces of information are required for the creation of the operation and station graphs. These are 1) relations between the operations, and 2) location of operations at assembly stations, and have been represented by the two manual steps in [Figure 3.11](#) in the respective order.

- **Step 2**

Once all operations are identified, valid relations must be defined between them, that represent their order of execution. Because complex products such as aerostructures can include a very large number of parts and joints (which directly translate to operations), manually defining relations between them would be a very tedious and error-prone task. Their automatic definition would first involve populating all possible relations between operations and then filtering out the feasible sequences. This can be very challenging, as the number of relations (given by $N!$, where N is the number of operations) would increase dramatically with the number of operations.

The use of operation sets is proposed to overcome this challenge. The definition of an operation set is given below. Operation sets are used as an aid to define relations between the operations in the assembly operation graph in a semi-automatic way.

Operation set

A set of operations with defined interrelations (edges), with an overall input and output node.

An example of an operation set for the installation of 20 ribs is shown in Figure 3.12. Step 2 of Figure 3.11 involves definition of such operation sets in a semi-automatic way. Edge relations shown Figure 3.12 do not have to be defined manually by the user, instead, they are derived from an abstract input given by the user. This input consists of three pieces of information, and is given in the format of a Python dictionary. These are as follows:

- i. Information about what operations should be included in the operation set.
- ii. Specification of a *method* to create relations between the operations. For example, this method can specify that operations be carried out, "all in parallel", "all in sequence", or a certain number of parallel sequences can be specified (as in Figure 3.12). Some programmed rules are then used to parse these specifications and to create the edge relations.
- iii. UUIDs of the input and output nodes. These are used to connect two operation sets together by specifying the same UUIDs. For example, the output of an operation set can be connected to the input of another operation set by specifying the same UUID for both these nodes.

The user input for creating the operation set shown in Figure 3.12 is included for reference in appendix C.3.

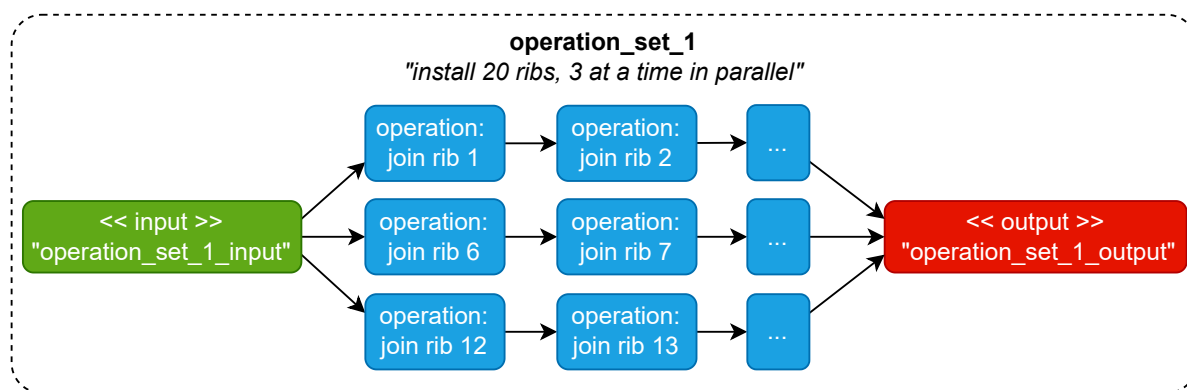


Figure 3.12: Example of an operation set to install 20 ribs in a product.

• Step 3

The next manual step in Figure 3.11 is to define stations with operations. In this step, the user defines what assembly stations exist, what operations are carried out there, and the quantity of each station (indicating duplicate stations that operate in parallel). The last quantity is useful for assembly line balancing, as by adding duplicate stations, the critical time per shipset for the station can be reduced [33]. This is a completely manually step, and involves another user input as a Python dictionary. An example of such an input can be seen in appendix C.4. Note that, in addition to assigning operations to a station by specifying operation sets, individual operations that are not part of an operation set can also be directly assigned.

• Step 4

Once the above information is available, the operation graph can be automatically derived, which is the next step in Figure 3.11. This is accomplished by creating a union of all operation sets assigned at all stations. In this process, duplicate input and output nodes that were specified in step 2 are removed, while maintaining all edge relations. This process is shown graphically with a simple example in Figure 3.13.

The union operation results in a single input or output that is connected to both the operation sets. Such nodes represent connections between the operation sets, and are therefore relabelled

as *connectivity* nodes (shown in yellow in Figure 3.13). The union is followed by some post-processing steps, in which these intermediate connectivity nodes are removed, and operation nodes are directly connected to each other. This step improves readability of the resulting operation graph, as it shows relations as they would be executed in reality, from one operation to the next. It is to be noted that the connectivity nodes are not “created” during the union process. They are already present as input or output nodes before the union, and are just relabelled after the union to correctly identify them for the post-processing step.

During these post-processing steps, some checks that ensure feasibility of the operation graph are also included. The checks included within the scope of this thesis ensure the following for an assembly sequence to be valid:

- Predecessor result nodes of all joint operation nodes have the parts that are required by the joint operation in their accumulated results. Absence of a part would indicate that relations have been modelled incorrectly between the operations.
- Each operation in the product is present in the operation graph exactly once. So, it must not be missing, and must not be repeated.
- Union of the operation graphs is a DAG. For example, this condition can be violated if the user defines input and output nodes of operation sets in a way that results in a cyclic graph after the union.
- Both the product start and product end nodes have been defined.

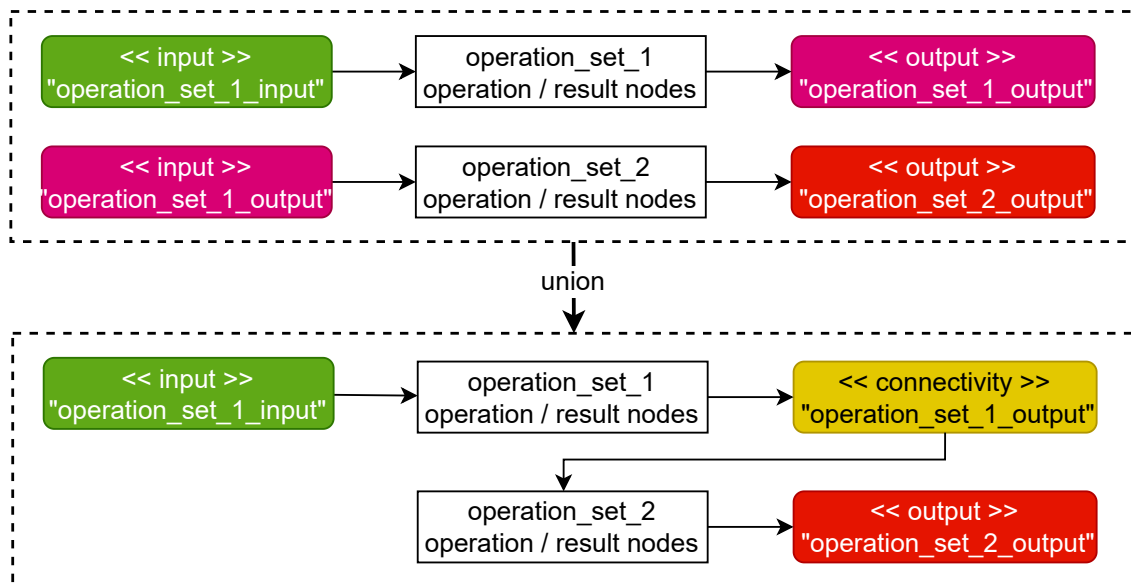


Figure 3.13: Union of two operation sets. Duplicate input or output nodes that were defined to connect the operation sets are removed, and converted to single connectivity node. This is done while retaining edge relations. Note that the rectangle shapes between the input and output nodes are placeholders for the operation and result nodes of the operation sets. They are not shown as they are not relevant, and to keep the figure size manageable for clarity.

• Step 5

The final step in Figure 3.11 is the automatic derivation of the station graph. This is accomplished by creating a quotient graph from the operation graph, where nodes belonging to the same station are merged together. This has been explained previously in section 3.3.2.

Structure of the assembly model in a product system

This section will discuss structure of the assembly model relative to a product with manufactured primitives. A UML class diagram for the assembly model in a product is shown in Figure 3.14. Note that only the relevant classes, and attributes within them are shown for clarity. The product is at the top,

which is composed of one assembly model (class `AssemblyModelBase`), and one or more manufactured primitives (shown by class `ManufacturedPrimitive`). The most important attributes in the assembly model include the `operation_graph`, `station_graph`, and the `critical_time_per_shipset`. Their calculation method has been discussed in detail in the previous sections. The assembly model itself is composed of the class `ProductOperationCollector`, one or more operation sets (class `OperationSet`), and one or more production stations (class `ProductionStation`). Class `ProductOperationCollector` is responsible for the first automatic step in Figure 3.11, to identify all operations in the product, and therefore, it is composed of one or more objects of class `OperationBase`. There is a one-to-one

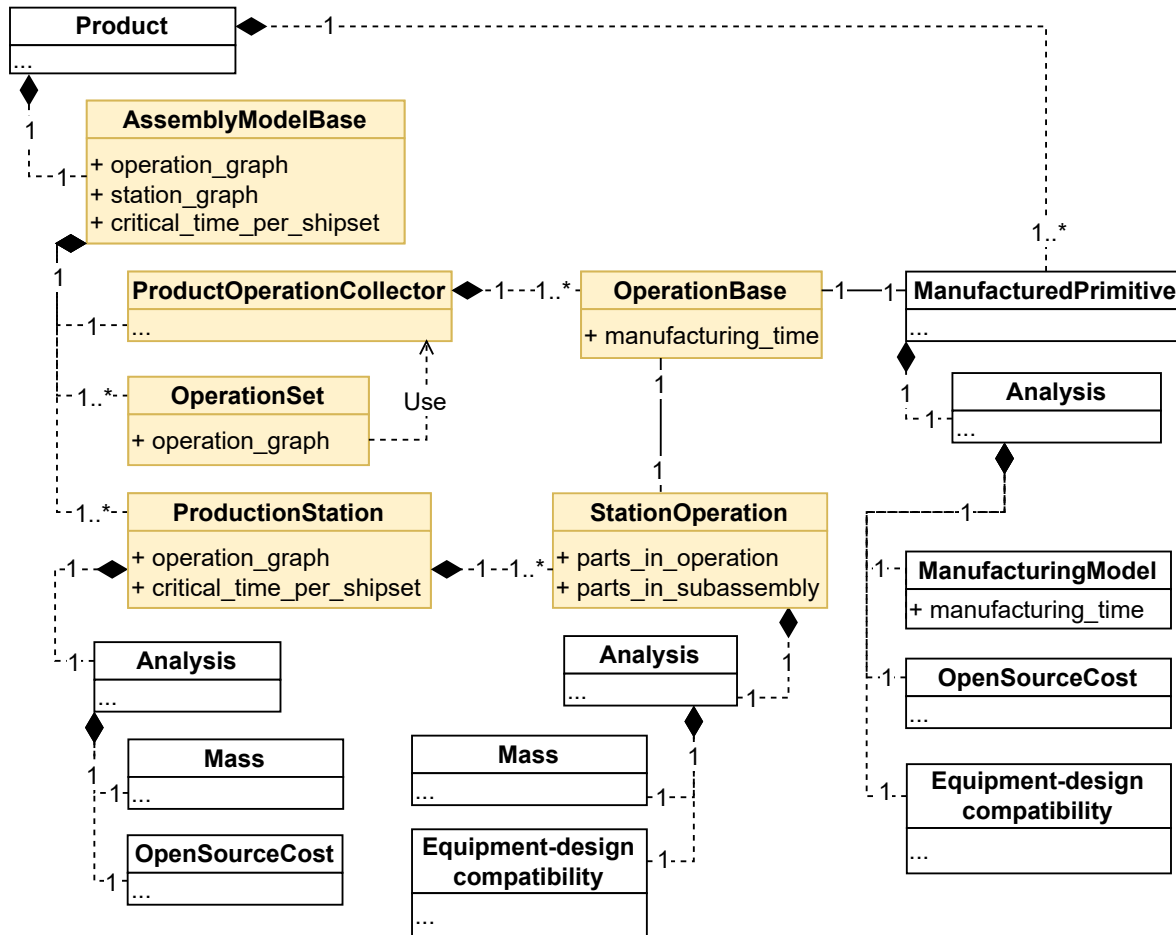


Figure 3.14: UML class diagram of the assembly model integrated in a product. Classes within the assembly model subpackage are highlighted in yellow.

correlation between each `OperationBase` object and a manufactured primitive in the product. The manufactured primitive can be linked to one or more analysis tools or disciplinary models, which has been shown on the right. The `OperationBase` objects have a `manufacturing_time` attribute, whose value is obtained from the `ManufacturingModel` of the manufactured primitive.

Next, the assembly model also has one or more `OperationSet` objects, which are created based on the user inputs, and depend on the `ProductOperationCollector` to access all operation objects. Each `OperationSet` also has an `operation_graph` attribute to represent the operation sets.

Additionally, the assembly model also has one or more `ProductionStation` objects. By definition, any set of operations can be assigned to a station, and hence these represent both, part manufacturing and assembly (joints) stations. An operation graph is also available at the stations, representing the assigned operations, and relations between them. The `critical_time_per_shipset` attribute is calculated on the basis of this operation graph.

Each `ProductionStation` is also composed of one or more `StationOperation` objects. These are created based on the operations that have been assigned to the station, and therefore have a

one-to-one association with operations under the `ProductOperationCollector`. The main role of the `StationOperation` object is to add information that only becomes available once the overall assembly sequence is known. This is the `parts_in_subassembly` attribute, which is derived from the `parts_in_result` attributes of the result nodes in the assembly model operation graph that represents all operations. To restate, this attribute accumulates parts from the predecessor results, and this gives an idea of the state of subassemblies at different levels of completion.

The last point to discuss is that, based on the presented structure of the assembly model, analysis tools can also be added to assembly model objects. This has been shown for the `ProductionStation` and `StationOperation` objects in [Figure 3.14](#). In addition to providing important parameters of interest, there is an additional benefit of including analyses at the assembly level, as in some cases the analysis results might change with additional information from the assembly model.

For instance, consider the equipment-design compatibility analysis in [Figure 3.14](#). One of the checks performed by this analysis ensures that the overall dimensions of the `ManufacturedPrimitive` is within the maximum allowable dimensions of the equipment being used. Now consider a joint based manufactured primitive. At the `ManufacturedPrimitive` object, this analysis would consider just the parts involved in that joint operation. But at the `StationOperation` object, this analysis would consider all parts in the subassembly, as additional parts may have already been connected to the parts involved in this joint. Therefore, it is possible that the equipment-design compatibility is no longer satisfied, as the effective overall dimensions of the primitives have changed.

This concludes the discussion about the theoretical aspects of the manufacturing information model. The next chapter will discuss some details about its implementation.

4

Implementation of the Manufacturing Information Model

4.1. Software for implementing MIM

The manufacturing information model has been implemented using ParaPy¹, which is a KBE platform developed in Python. The use of a KBE platform offers some specific advantages over directly using a programming language. These have been discussed in detail in [34], but have also been summarised below with some specific examples of their benefit for the manufacturing information model.

- **Runtime caching:** With this feature, the KBE system is able to store calculated results in memory during runtime, and makes them available for reuse when required instead of re-calculating them again. This helps save computational resources. An example where this will be useful is for the manufacturing time attribute in the manufacturing model. The value for this attribute is obtained through a connected analysis tool, based on the details defined in the manufacturing model. The manufacturing time is then accessed multiple times during the creation of the assembly model, such as during creation of the Operation objects, and for the calculation of critical station time.
- **Dependency tracking:** This feature is related to runtime caching, and in this, the KBE system ensures that the cached values are always valid. In case a change is made during runtime that renders the cached value invalid, it is removed from the cached values, and re-calculated if its value is needed. For example, if the thickness of a part is changed in the product system, then its manufacturing model, and hence also manufacturing time would be invalidated.
- **Demand driven evaluation:** With this feature, only required values in the system are calculated when a value related to them is requested by the user. For example, if the mass of a manufactured primitive is requested, then only the relevant properties, such as material density, and total surface area, and thickness would be calculated and made available in the manufacturing model. Other information, such as that related to the equipment or site will not be processed at this moment. This again helps in saving computational resources.

Furthermore, ParaPy also offers a built-in well featured graphical user interface (GUI) for visualisation, and verification of the results.

In addition to ParaPy, two other Python packages are used in MIM:

1. **material_db**

The first package is a material database, developed in-house at GKN Fokker Aerostructures. This will be used for material properties to complement information in the database subpackage of MIM. Note that, any other similar material properties database can also be used, as long as the required information about material properties is provided by such a package.

¹<https://www.parapy.nl/> (accessed 16 August 2022)

2. NetworkX²

This package is used for the creation and manipulation of the various graphs in the assembly model. This package has been selected as it offers a pure Python implementation, making it easy to integrate with ParaPy, extensive documentation, and availability of relevant built-in features and methods. For example, it supports calculation of the longest path through a DAG, creation of quotient graphs, and union of graphs.

4.2. Integration of MIM with a product model

This section will give an overview of the steps that are required to integrate MIM with a product model. These steps are shown as a process diagram in Figure 4.1. Note that since MIM has been implemented using ParaPy, it is necessary that the product model is also based in ParaPy.

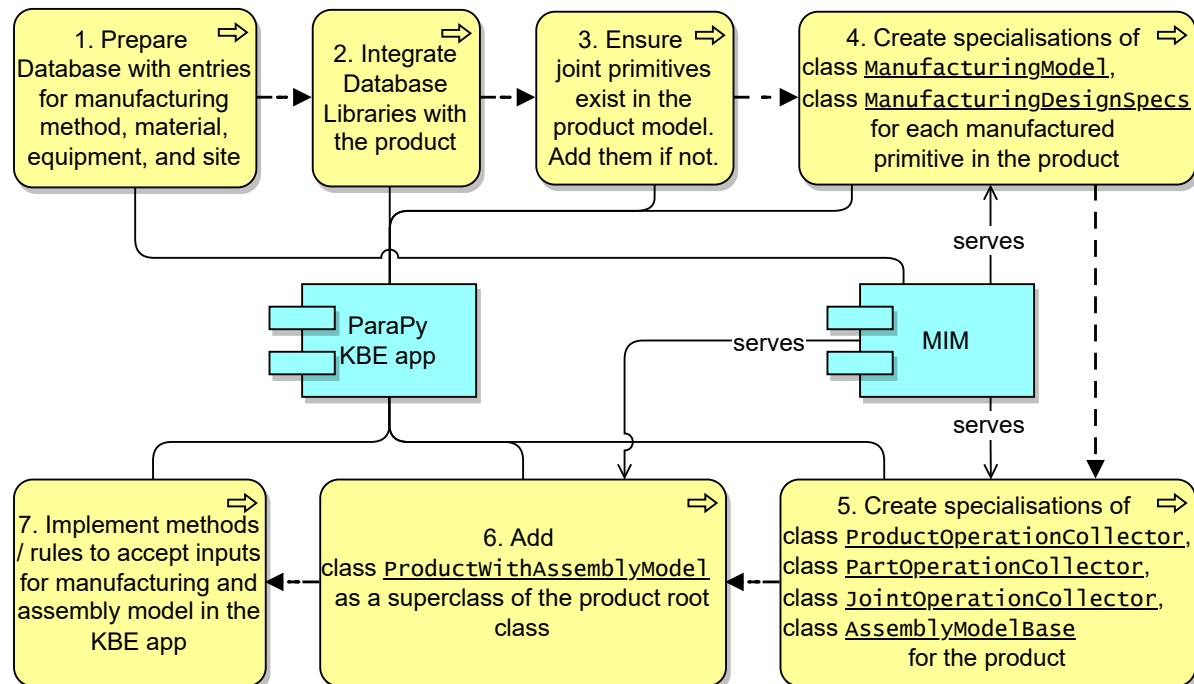


Figure 4.1: Overview of the steps required for integrating MIM with a ParaPy based KBE application.

A brief description of each of the steps in Figure 4.1 is given below. Note that all steps in the process diagram are performed manually.

- **Step 1**
In this step, the MIM database is prepared with information about each manufacturing information category. This step is associated to MIM.
- **Step 2**
This step involves integrating the prepared database with the product model, which is shown as "ParaPy KBE app" in Figure 4.1. The libraries subpackage provides the necessary base classes for this.
- **Step 3**
The product model must have joint primitives, as they are essential to model the assembly process. The KBE app must be updated to include joints if they are not implemented.
- **Step 4**
Here, specialisations of base classes in the manufacturing model are created for each identified manufactured primitive in the product. This step is associated to the KBE app as these specialisations are created there. And this step is dependent on MIM, as the base classes are provided

²<https://networkx.org/> (accessed 16 August 2022)

by it.

- **Step 5**

In this step, specialisations of base classes in the assembly model are created for the specific product in consideration. The first three classes are related to identifying the operations from the product model. A specialisation of class `AssemblyModelBase` is created to specify which specialisation of `ProductOperationCollector` should be used for the product model. This step is also associated to the product KBE app, and is dependent on MIM, just as the last step.

- **Step 6**

This step adds the assembly model as a component of the product model. Again, this step is performed in the KBE app, so it is associated with that, and it gets the class to be used as the product superclass from MIM, making the step dependent on it.

- **Step 7**

As discussed in [chapter 3](#), the manufacturing and assembly model require some user inputs. Therefore, provisions have to be made in the product KBE app to accept these inputs, and to provide them to the product specific specialisations of the manufacturing model and assembly model classes.

In addition to the details mentioned in the above steps, the product model must satisfy some other requirements to work properly with MIM. These are listed below:

- An analysis tool must be linked to the product model that is able to calculate the process time for each manufactured primitive. These values are required by the assembly model for the calculation of the critical path, and the production rate. In this thesis, an open source cost tool [24] that is able to calculate the process time and cost is used for this.
- The manufacturing methods specified in the MIM database must also be supported by the analysis tool mentioned in the previous point for the process time calculation.
- A material properties database must be linked to the product model that can be used by the MIM database, **or**, capabilities of the MIM database must be extended to include these material properties. In this thesis, the `material_db` package mentioned in [section 4.1](#) is used for this.

4.3. Supporting models and analysis tools used in this thesis

Certain models and analysis tools will be used in this thesis for the verification & validation of MIM's implementation ([chapter 5](#)), and to also apply MIM to an industrial use-case ([chapter 6](#)). This section will introduce these tools and models.

4.3.1. MDM: Multidisciplinary Modellers

MDM is a KBE based multidisciplinary design system for developing aircraft moveable structures that has been developed at GKN Fokker Aerostructures [26]. It is based in ParaPy, and has a generic set up that supports the design of a variety of aerostructures such as, flaps, horizontal & vertical stabilisers, and fixed wingboxes [26]. This design system will be used as the product model with which MIM will be integrated in this thesis. The main reasons for its selection include:

- Ease of integration, as both MDM and MIM are based in ParaPy. This will also allow full use of the KBE specific features that were discussed in [section 4.1](#).
- MDM has a fairly generic set-up, making it suitable for the design of different kinds aerostructures. This will help show MIM's ability to work with such a generic system.
- MDM has an extensive integration with various analysis tools, some of which will also be used for the use-case in this thesis. This will help in reducing implementation effort that is not directly associated to show functionality of MIM.
- This thesis was conducted in collaboration with GKN Fokker Aerostructures, and therefore, demonstrating integration with their systems will prove to be of practical benefit for them.

4.3.2. CATMAC: Cost Analysis Tool for Manufacturing of Aircraft Components

CATMAC is an open source, process-based cost estimation tool [24], developed with a focus on aerostructures. This analysis tool will be used for the calculation of recurring cost and process time of each manufactured primitive. The main reasons for its selection include:

- It is a process-based cost tool, and therefore is better suited to work with new types of products and manufacturing processes when compared to parametric cost tools. The latter use cost estimation relations (CERs), which are derived from historical data [6].
- The tool is open-source, and is set up in a way that makes it easy to add new manufacturing methods, materials, and manufacturing environments. Hence, it can be setup to work with even proprietary in-house manufacturing methods.
- The tool is already integrated with MDM, and hence would not require additional implementation effort.

4.3.3. MPM: Mass Properties Module

MPM is a ParaPy based package developed in-house at GKN Fokker Aerostructures for product mass calculation. It will be used for the same, as mass is one of the parameters of interest within scope for quantifying the included production considerations. The package is integrated with MDM.

4.3.4. Compatibility

Compatibility is a ParaPy based package developed by the author to support this research, and will be used to perform a check for the product's compatibility in some of the categories identified in [21, 22]. Each compatibility category identifies broad groups of related information that must be compared with each other to ensure compatibility. Some examples of compatibility categories include: material-material compatibility and equipment-design compatibility.

The actual compatibility check is then performed with one or more criteria under each category. Compatibility criteria define the exact comparison that will take place using information from the objects in a compatibility category. For a compatibility category to be compatible, all criteria in that category must pass the compatibility check.

The compatibility categories and their respective criteria included in this thesis are summarised in Table 4.1.

Table 4.1: Compatibility categories and criteria included with the scope of this research.

Compatibility category	Criteria	Description
Material – Method	database	The selected material and manufacturing method must be compatible based on their respective specified information in the MIM database.
EquipmentSet – Method	database	The selected equipment set and manufacturing method must be compatible based on their respective specified information in the MIM database.
Equipment – Design	allowable_dimensions	The bounding box dimensions of the object(s) under consideration must be within the maximum allowable dimensions of all equipment in the selected equipment set.
Material – Material	galvanic_corrosion	Aluminium fasteners must not be used in carbon fibre parts, or vice-versa, to prevent galvanic corrosion in the aluminium component.
	induction_welding	Parts being joined using the induction welding method must have the same matrix material.

4.3.5. RVF: Requirement Verification Framework

RVF is a model-based requirements verification framework that can also be used for setting up optimisation workflows from the defined requirements [3]. It will be used for managing design and manufacturing requirements for the use-case in this thesis as it offers an implementation to which adding requirements from MIM would be an easy and straightforward process, without any special modifications to either RVF or MIM.

Additionally, RVF is also implemented in ParaPy, making it easy to integrate with the other tools and models in this thesis. Lastly, the ability to set up optimisation and DOE workflows from requirements is a unique feature offered by RVF, and may prove to be beneficial in setting up optimisation workflows that include production considerations with MIM.

4.3.6. Relation to production considerations and parameters of interest

This section gives an overview of which model and/or analysis tool discussed above accounts for what production considerations and parameters of interests within scope. This information is summarised in Table 4.2 and Table 4.3.

Table 4.2: Which model or analysis tool accounts for which production consideration.

Production consideration	Accounted by
Joint information	MDM, MIM
Product compatibility	Compatibility, MIM
Assembly sequence	MIM
Part integration	MDM, MIM

Table 4.3: Which model or analysis tool accounts for which quantity of interest.

Parameter of interest	Accounted by
Cost	CATMAC
Structural mass	MPM
Production rate	CATMAC, MIM
Requirements compliance	RVF

5

Verification and Validation

In this chapter, implementation of MIM will be verified and validated by applying it to a product model. Specification of the product model will be presented in the first section, which will be followed by details of how MIM has been set up. Some requirements will also be imposed on the product to test the functionality of RVF. Finally, the last section will discuss verification and validation of the implementation based on the generated results.

5.1. The product model

This case will consider skin panel subassemblies of a wingbox product, with limited number of manufactured primitives. These primitives are shown in [Figure 5.1](#), and consist of two skin panel subassemblies. One is a conventional skin panel with an L-stringer connected to it with a joint. The other is an integrally stiffened skin panel, with one integrated stringer (shown in the same colour as the panel). The integrated skin stringer is also connected to another conventional L-stringer with a joint. Only a limited number of simple manufactured primitives are being considered in this use-case as the main goal here is to verify implementation of the manufacturing information model.

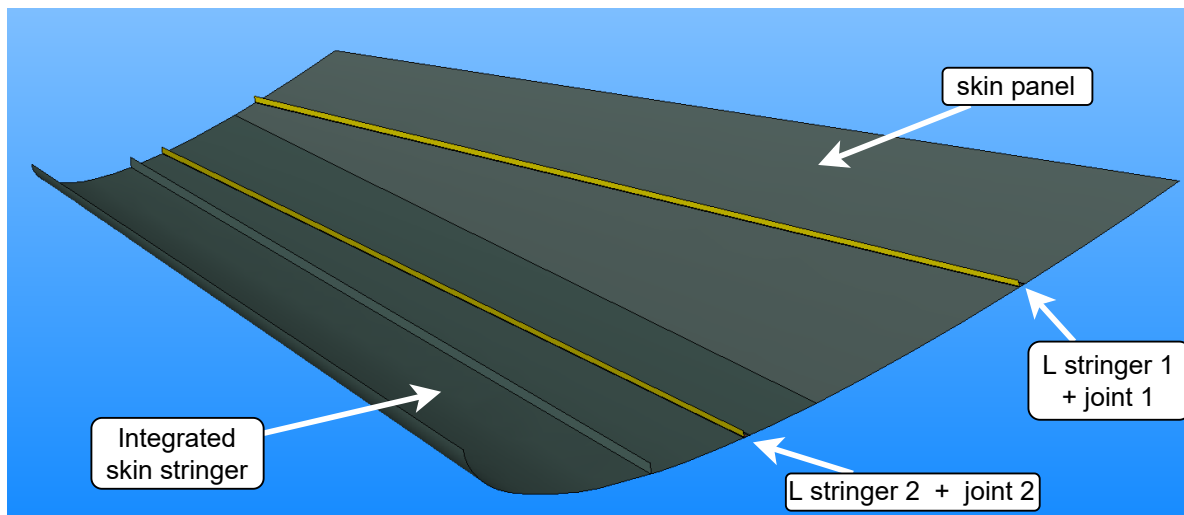


Figure 5.1: Manufactured primitives in the product model for verification and validation of MIM's implementation.

5.2. MIM setup

As the first step, the manufacturing model for each manufactured primitive will be set up. For this, a user input will be given in the format shown in [appendix C.1](#) for each manufactured primitive.

Table 5.1: Manufacturing model information category selections for the manufactured primitives in the case for verification and validation.

Manufactured primitive	Method	Material	Equipment set	Site
Skin panel	hand_layout	CFRP ^a	hand_layout equip_set_1	NL1_PPD
L stringer 1	hand_layout	CFRP	hand_layout equip_set_1	NL1_PPD
Joint 1	induction_welding	-	induction_welding equip_set_1	NL1_PPD
Integrated skin stringer	tapas_layout [20]	CFRP	tapas_layout equip_set_1	NL1_PPD
L stringer 2	hand_layout	CFRP	hand_layout equip_set_1	NL1_PPD
Joint 2	mechanical_joint	CRES ^b / Ti ^c	hi_lite_joint_manual equip_set_2	NL1_PPD

^a CFRP: carbon-fibre-reinforced polymer^b CRES: corrosion-resistant steel^c Ti: Titanium**Table 5.2:** Assembly model configuration for the verification and validation case.

Station UID	Assigned operation set UID	Operation set relations description
stringer_manufacturing	oper_set_part_mfg_stringers	Sequential, sorted in the direction from the leading edge to the trailing edge
skin_panels_side1_manufacturing	oper_set_part_mfg_skin_panels_side1	All in parallel
integrated_skin_stringers_manufacturing	oper_set_part_mfg_integrated_skin_stringers	All in parallel
side1_stringer_installation	oper_set_side1_stringers_installation	Sequential, sorted in the direction from the trailing edge to the leading edge

This input will be used to make selections for each manufacturing information category in the manufacturing model of each manufactured primitive. These selections are summarised in [Table 5.1](#). A variety of manufactured primitives, manufacturing methods, materials, and equipment sets have been chosen which will help verify the implementation across different cases. Selections for all manufacturing information categories are made using direct user definition of the option shown in [Table 5.1](#), except for the material category. Selection for this category is made using the option that derives this information from the design definition model. Note that, with the options in [Table 5.1](#), the production considerations of part integration and having different joint information are also accounted for.

Next, inputs for the assembly model can be prepared. Note that the user inputs for the manufacturing and assembly models are given to the product model as Python dictionaries. The product model (MDM) was updated to handle these inputs, as described in step 7 of [Figure 4.1](#).

Configuration of the assembly model involves specifying what stations exist, and what operations are assigned to each station. The specified stations, operation sets assigned to them, and a description of operation relations in the operation set is given in [Table 5.2](#). Note that only one operation set has been assigned to each station for simplicity. The operation set relations describe how operations are arranged in the operation set. For instance, in the first operation set, all stringers would be manufactured in sequentially, one after the other, starting from the stringer closest to the leading edge, and then selecting the next stringer in the direction towards the trailing edge.

Lastly, the input and output of the operation sets have been specified in a way so that all part manufacturing stations (the first three stations in [Table 5.2](#)) will first work in parallel, and then be connected to the stringer installation station for the joints.

5.3. RVF setup

Integration with RVF will be verified by imposing some requirements on the product. These have been summarised in [Table 5.3](#).

Table 5.3: Requirements imposed on the verification and validation case.

Type	ID	Stakeholder	Requirement text	Test case
Performance	R-1001	Client	The skin panel subassemblies shall have a total cost of less than \$10000	CATMAC (cost tool)
Performance	R-1002	Client	The skin panel subassemblies shall have a total mass of less than 30 kg	MPM (mass tool)
Performance	R-1003	Client	The skin panel subassemblies shall have a number of shipsets per month of at least 25	Assembly model
Suitability	R-1004	Design team	The skin panel subassemblies shall have passed all compatibility checks	Compatibility analysis
Suitability	R-1005	Design team	The skin panel subassemblies shall have passed all manufacturing model validity checks	Manufacturing model

5.4. Verification and validation of results

5.4.1. Manufacturing model

The main role of the manufacturing model was to capture all information about how a primitive would be manufactured. Therefore, the model must be able to handle different kinds of primitives in a product definition, which can include parts, joints, and integrated parts. The simplified case includes all types of these primitives, and the manufacturing model has been integrated in each one of them. The resulting tree structure for each primitive is shown in [Figure 5.2](#).

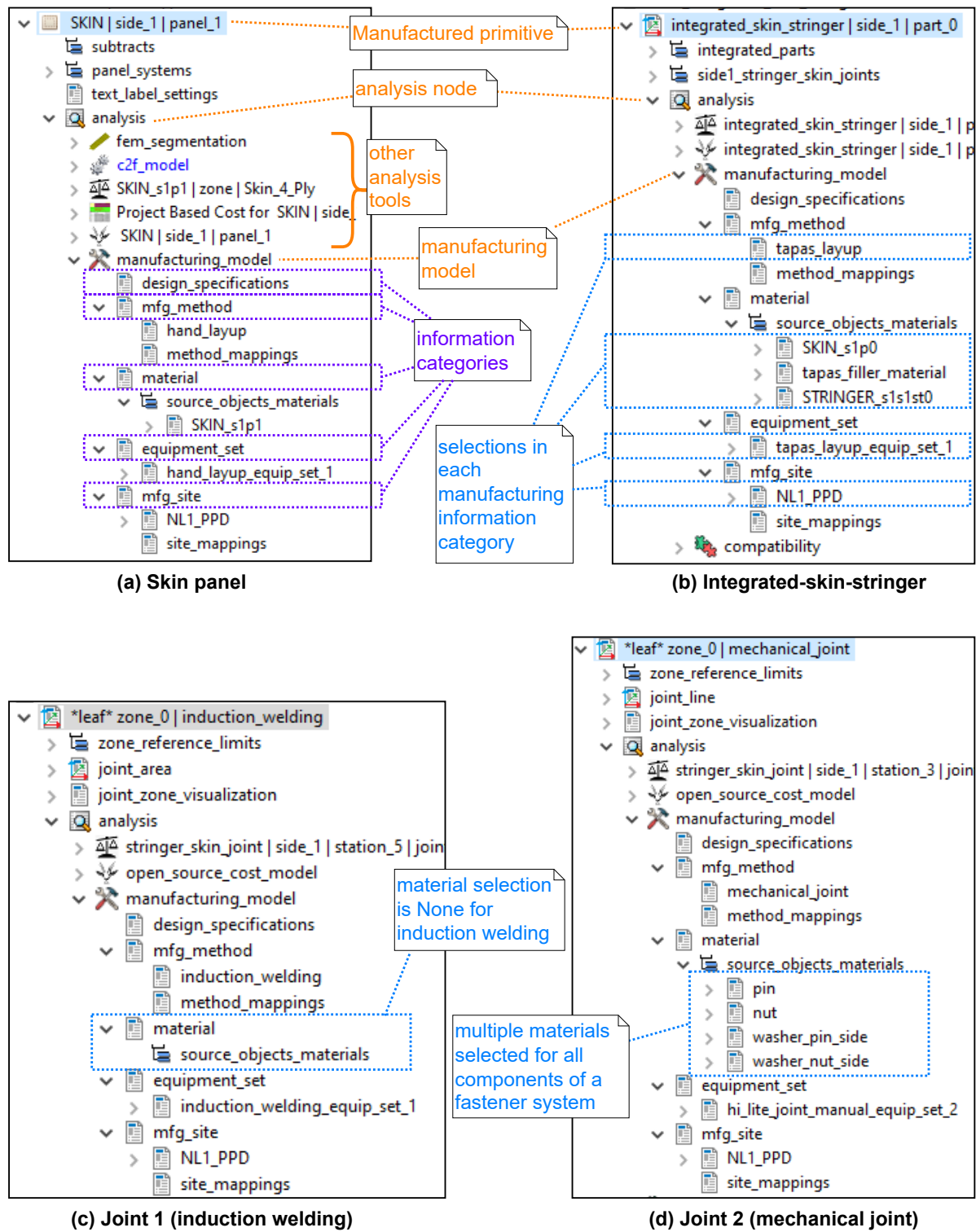


Figure 5.2: Integration of the manufacturing model in different types of manufactured primitives in the product model for the verification and validation case. Screenshots show the tree structure in ParaPy GUI.

A few aspects will be highlighted next. The manufacturing model is located under the *analysis* node of the tree structure, which includes all analysis tools and/or disciplinary models. The five information categories under the manufacturing model node can also be seen. Selections in each manufacturing information category can be seen as the first sub-node in each category. The material category has

information for material selections for each object under the manufactured primitive. Therefore, it can host information not just for parts, but also for integrated parts, which have more than one functional parts (Figure 5.2 (b)), and for joints, which can also be composed of more than one objects in case of a fastener system (Figure 5.2 (d)), or be nothing in case of an induction welded joint (Figure 5.2 (c)).

Attributes available in the design specifications information category of two of the primitives, skin panel and joint 2, are shown in Figure 5.3. The name of the specific class used for each design specifications can also be seen in the top right corner of each image. For the skin panel (Figure 5.3 (a)), these specifications include dimensional properties, such as total area, average thickness, reference length, width and height etc. On the other hand, for Joint 2 (Figure 5.3 (b)), these include the total number of fasteners, fastener pitch, indexing hole diameter etc.

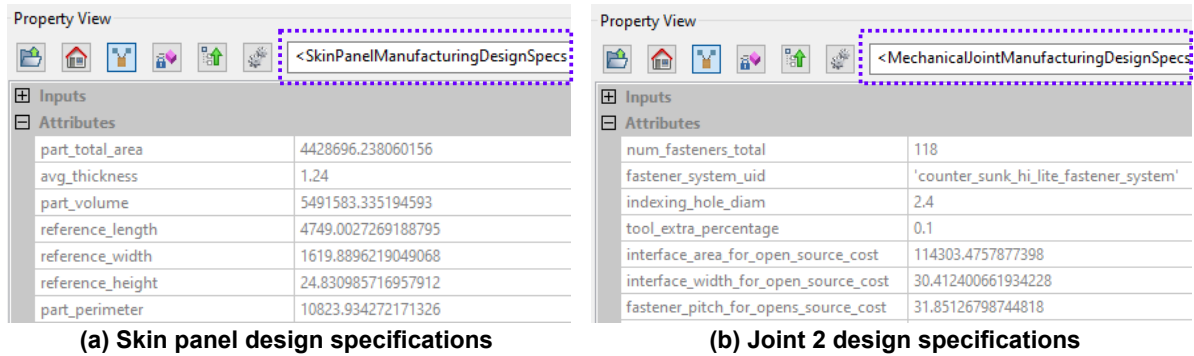


Figure 5.3: Design specifications information category for the skin panel and joint 2 primitive in the verification and validation case.

Next, the important input slots and attributes for the method information category can be seen in Figure 5.4. The selected_dicts_list attribute and valid_dicts_list_from_drivers input slot can be seen here. Additionally, the attribute is_selection_valid is available that indicates validity of the selection.

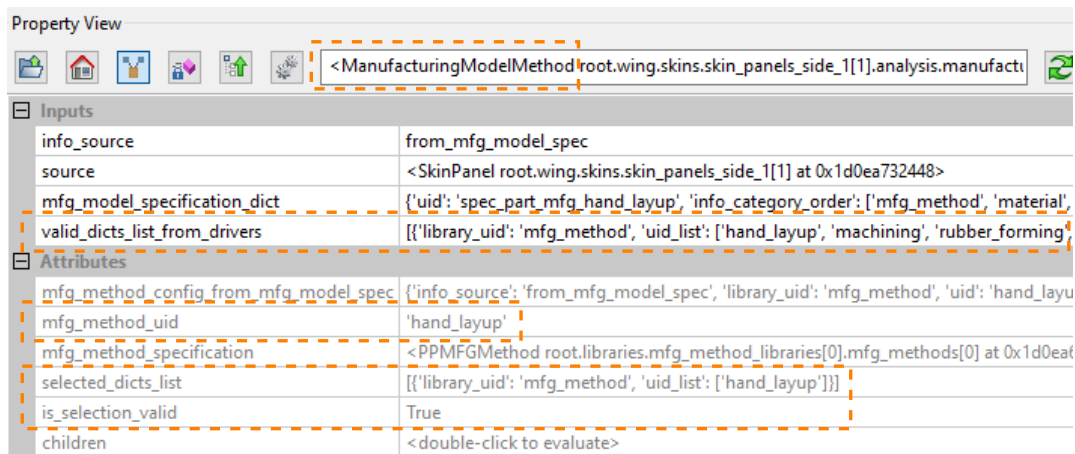


Figure 5.4: Important slots in the manufacturing method information category for the skin panel primitive.

Lastly, attributes in the method_mappings and site_mappings nodes of the manufacturing model can be seen in Figure 5.5a and Figure 5.5b respectively. They contain information for referencing an information category selection to other analysis tools. For example, the site name in the manufacturing model is NL1_PPD, but the correct name for the open source cost tool is Netherlands_HLU_Batch1.

It can be concluded from the above discussion that the desired structure and functionality of the manufacturing model has been achieved with this implementation. The manufacturing model is able to capture production related information for different types of manufactured primitives. Selections in

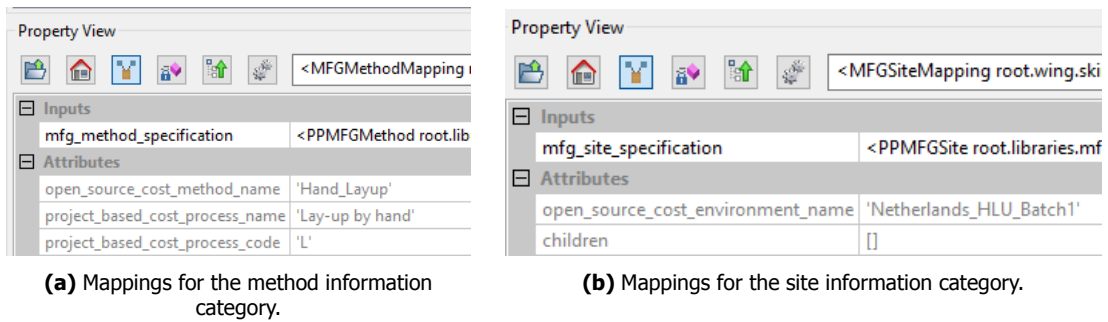


Figure 5.5: Mappings of selections in method and site information categories to other analysis tools in the system model.

each manufacturing information category can be checked for validity using the `is_selection_valid` attribute. Lastly, it is able to provide mappings for the selected manufacturing method and site to other analysis tools.

5.4.2. Database

The main function of database is to provide the manufacturing model with necessary data for its information categories, and methods to access it. This is accomplished through the libraries subpackage of database, whose implemented tree structure has already been shown in Figure 3.6.

To get an idea about the information stored in the data files and managed by libraries, consider the example shown in Figure 5.6. The list of all available equipment is shown in Figure 5.6a, and Figure 5.6b shows data associated with one such equipment ("milling centre 1").

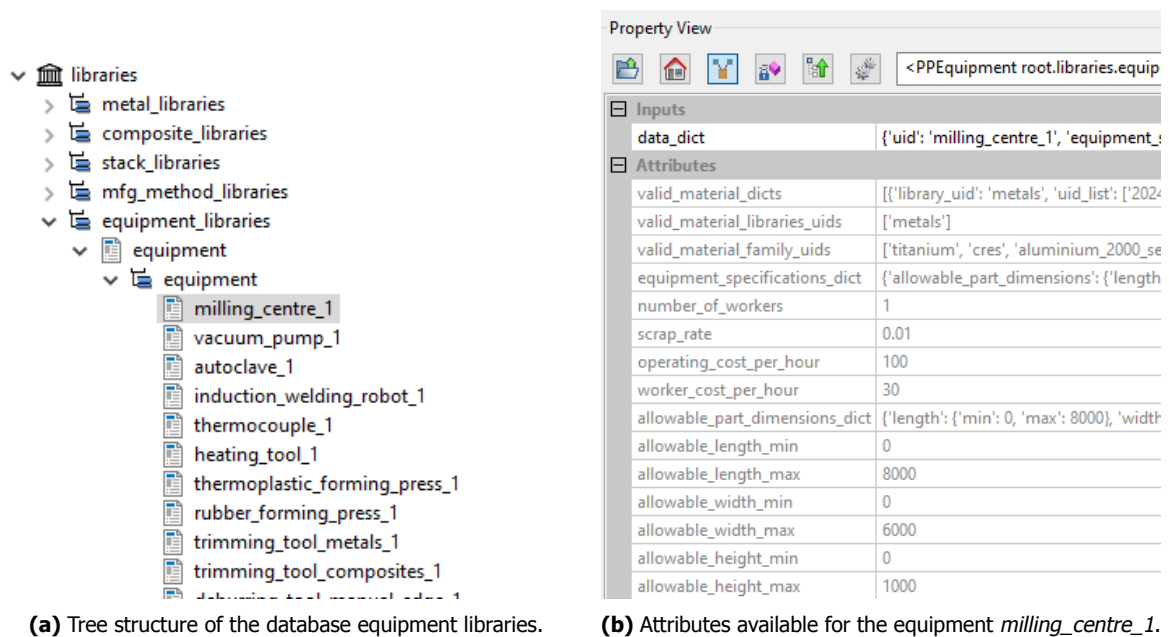


Figure 5.6: Example from the database libraries, showing the available attributes for an equipment.

Special methods to access valid selections for each of the four manufacturing information categories of manufacturing model were also implemented in the libraries subpackage. An example of such a method was previously shown in Figure 3.8. The result of two such queries is shown in Figure 5.7. Query 1 shows all available manufacturing methods at the site *NL1_PPD*, and query 2 shows all valid manufacturing method at the site *NL1_PPD*, that are valid if the primitive material belongs to the *aluminium_2000_series*. A list of all such methods implemented in the libraries subpackage is as follows:

- `get_mfg_method_dicts_list()`
- `get_equipment_set_dicts_list()`
- `get_mfg_site_dicts_list()`
- `get_mfg_material_dicts_list()`

To reiterate, these queries are made automatically in the manufacturing model to find the value of the `valid_dicts_list_from_drivers` slot for each manufacturing information category, based on selections made in the other categories. These query methods can also be used manually by a user to explore possible options.

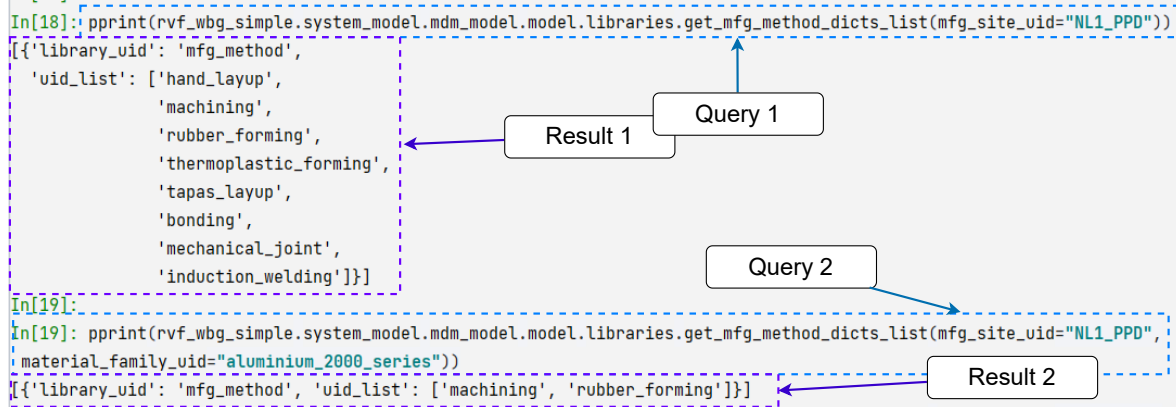


Figure 5.7: Results of information queries to get valid manufacturing methods using the database libraries subpackage. Screenshot shows the Python console window.

5.4.3. Assembly model

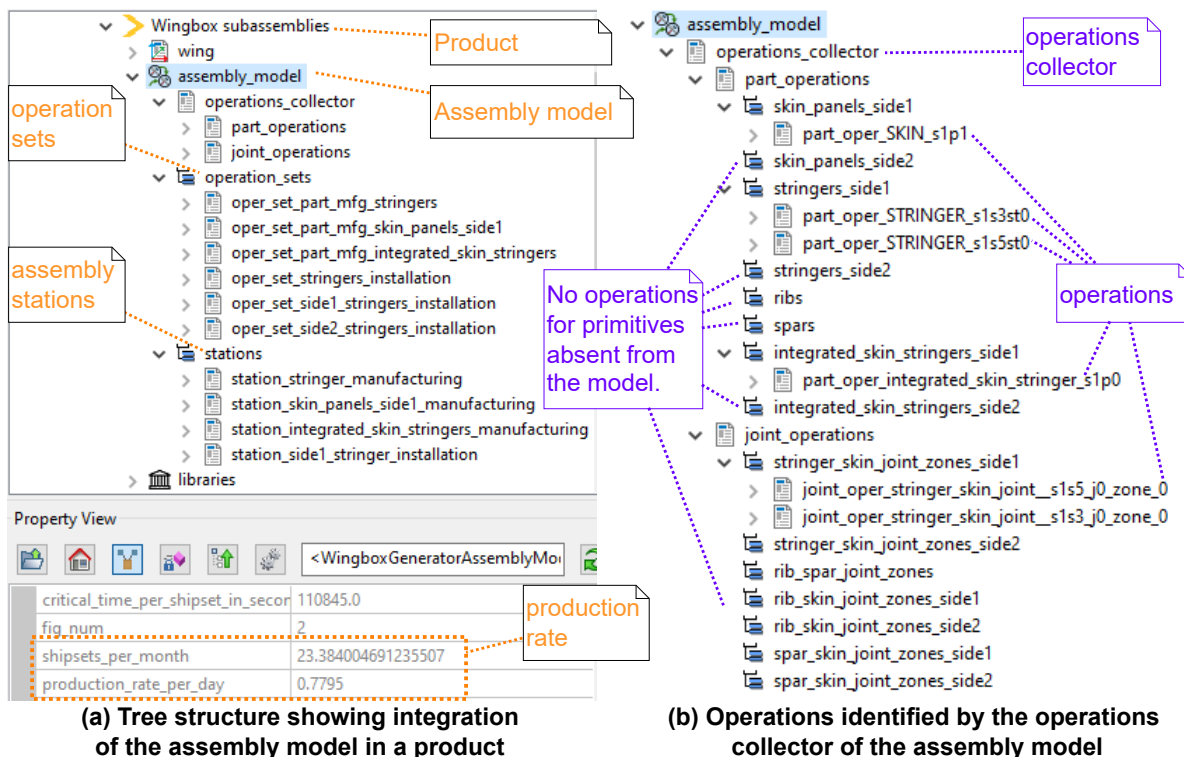


Figure 5.8: Tree structures showing details of the assembly model in a product for verification and validation.

Tree structure of the assembly model integrated in the product can be seen in Figure 5.8. The tree represents the UML diagram of the assembly model shown in Figure 3.14, with the operations

collector, operation sets, and stations nodes. Additionally, the attributes `shipsets_per_month` and `production_rate_per_day` are available in the property view below the tree. An expanded view of the operations collector can be seen in Figure 5.8 (b). It can be observed that only the manufactured primitives considered in this case (Figure 5.1) are identified as operations under the operations collector. Nodes of all other primitives are empty (indicated by a lack of dropdown arrow symbol). This is because the manufacturing model was only defined for the primitives considered in this case.

Operation graph for the overall assembly sequence can be seen in Figure 5.9. A legend for the different types of nodes is available in the bottom left corner. Each edge from an operation node to a result node also has a green label indicating the name of the manufacturing method in the operation to get that result. A number in blue can also be seen on the same edge, representing execution time for the operation in seconds.

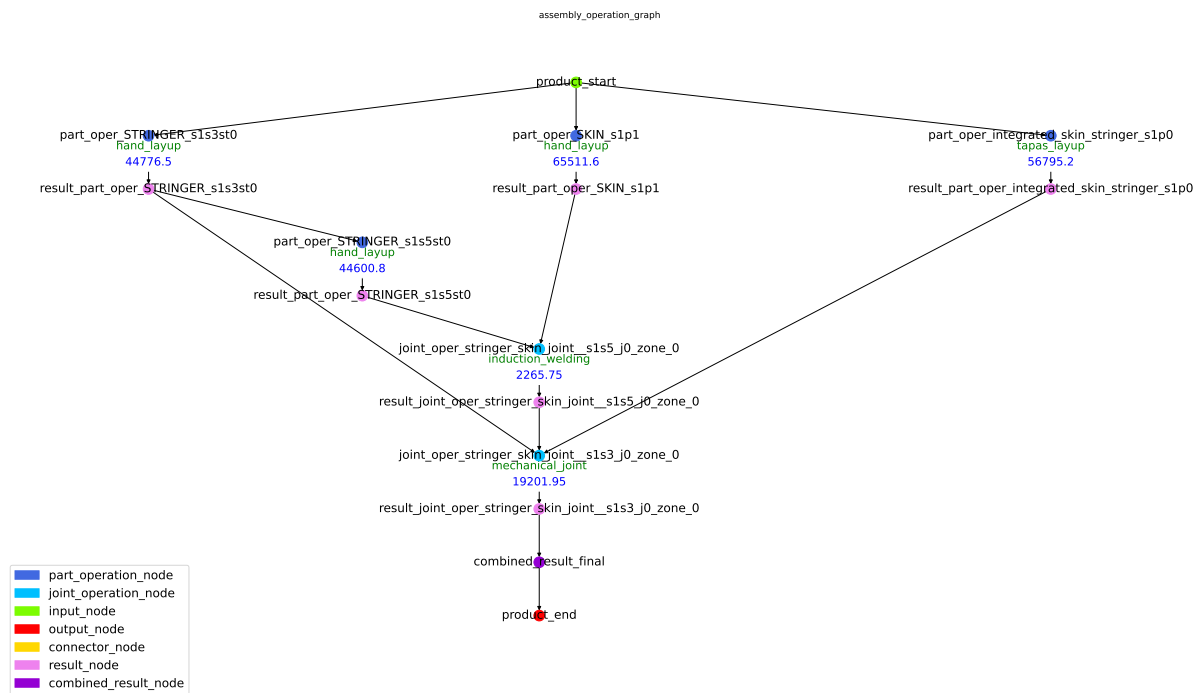


Figure 5.9: Operation graph for the overall assembly sequence in the verification and validation case.

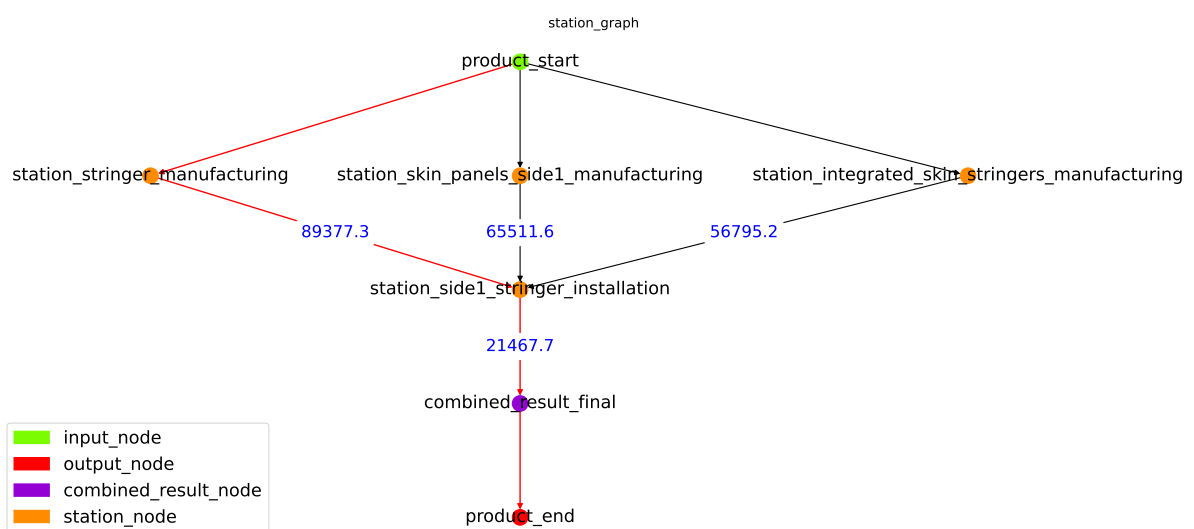


Figure 5.10: Station graph for the verification and validation case.

The station graph derived from the operation graph is shown in Figure 5.10. Critical time for the completion of operations at each station is represented on the outgoing edge from the station in blue (value is in seconds, and is shown on the edge for clarity). From this, the critical manufacturing path is derived and is shown with red edges. From the operation and station graphs, it can be observed that the part manufacturing operations start in parallel, with the two stringers being manufactured sequentially. This also contributes to the critical path being through the stringer manufacturing station, as it takes the most time. The joint operations are then carried out in sequence on the stringer installation station.

Furthermore, operation graph of the stringer installation station is shown in Figure 5.11, which represents just the operations at that particular station. The starting nodes in this graph are the result nodes of part operations that were executed at different stations.

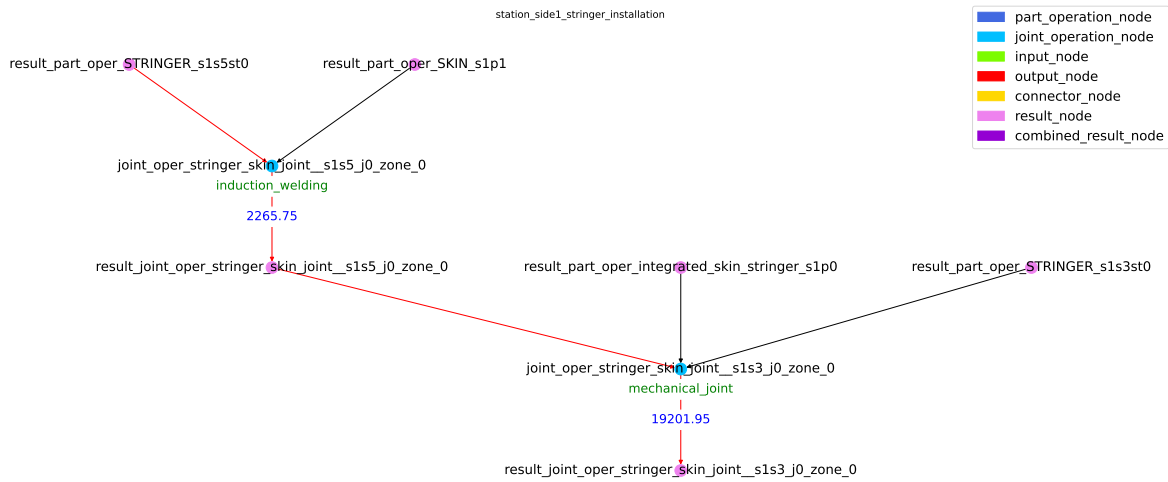


Figure 5.11: Operation graph for the production station *station_side1_stringer_installation* in the verification and validation case.

In conclusion, the assembly model has been properly integrated with the considered product, with all the discussed functionality. The right operations are identified and collected by the operations collector, operation graphs can be created at different levels (e.g. for the entire process or for a station). Finally, the assembly operation graph can be used to derive the station graph. The critical path was correctly identified, and production rate could be calculated from the station graph.

5.4.4. RVF

The result of integrating the considered product system in the RVF can be seen in Figure 5.12. It shows the requirement model next to the system model with all the requirements that were defined in Table 5.3. The means of compliance and identified test case can also be seen under each requirement. Finally, the compliance report was successfully auto-generated (Figure 5.13), indicating the correct working of RVF.

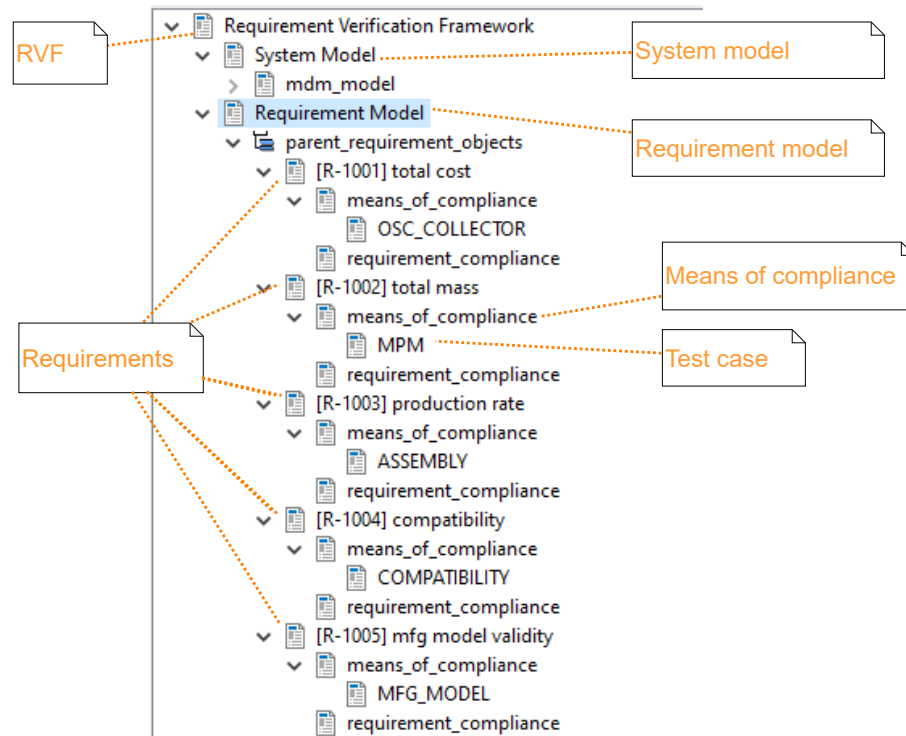


Figure 5.12: Integration of the verification and validation case product model in the RVF.

	Requirements	Textual Requirements	Compliance	Value	Difference
R-1001	total cost	The skin panel subassemblies shall have a total cost of less than \$10000	True	7914.74	20.85%
R-1002	total mass	The skin panel subassemblies shall have a total mass of less than 30 kg	True	21.66	27.78%
R-1003	shipsets per month	The skin panel subassemblies shall have a number of shipsets per month of at least 25	False	23.38	-6.46%
R-1004	compatibility bool	The skin panel subassemblies shall have passed all compatibility checks	True	True	
R-1005	mfg model validity bool	The skin panel subassemblies shall have passed all manufacturing model validity checks	True	True	

Figure 5.13: Auto-generated requirement compliance report from RVF for the verification and validation case.

6

Industrial Use-Case: Conceptual Design of a Wingbox

This chapter will discuss application of the manufacturing information model to the conceptual design of a wingbox to help assess its applicability real cases in the industry. The case will consider a product definition based on a generic design study conducted at GKN Fokker Aerostructures.

The analysis of this use-case will be guided by the second research question. Part 2.a of the question asks if the methodology allows identification of feasible-to-produce design concepts that may be unapparent (part 2.a.i), and better (part 2.a.ii). These sub-questions will be answered by implementing the use-case in a DOE workflow, with choices in the manufacturing and assembly model as the design variables. Hence, this DOE study will result in different manufacturing concepts for the product. An overview of the general steps involved in this workflow will be discussed in the first section of this chapter. Then, sections two to six will discuss specific details of the various parameters used in the workflow for this specific use-case.

For the identification of better designs, valid designs from the DOE study will be compared to a baseline concept that was finalised in the study at GKN Fokker Aerostructures. The available data for the baseline design can be referred to in [appendix D](#).

Lastly, the industrial use-case will also help answer part 2.b of the research questions. This question is related to analysing the effect of production considerations on gauging product viability. For this, the parameters of interest for some of the design points from the DOE study will be compared with and without the production considerations.

6.1. Design workflow for the automated inclusion of production considerations in design using MIM

This section will present an updated version of the manual approach for including productions considerations in design ([Figure 2.2](#)). This new workflow will show how MIM can be used to automatically include production considerations in the conceptual design process.

The process diagram for the new workflow is shown in [Figure 6.1](#). Similar to the process diagram of the manual workflow, the production considerations and parameters of interest in the scope of this thesis are specified in coloured text in the diagram to put them in context of the workflow. These are also listed below in their respective colours for reference:

Production considerations:

- (1) Compatibility
- (2) Assembly sequence
- (3) Part integration
- (4) Joints

Parameters of interest:

- (i) Cost
- (ii) Mass
- (iii) Requirements compliance
- (iv) Production rate

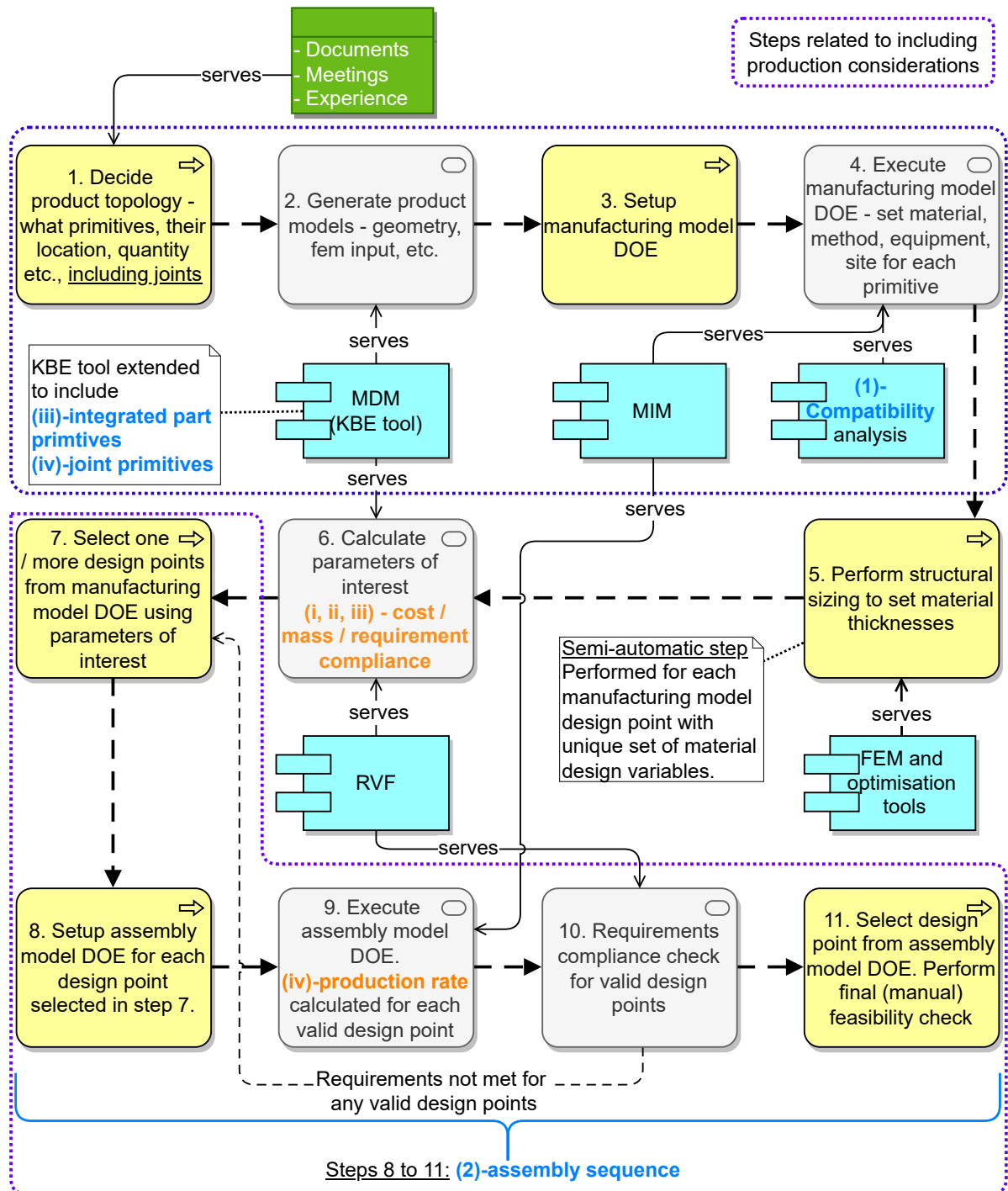


Figure 6.1: Workflow for the automated inclusion of production considerations in conceptual designs using MIM. Note that step 5 (structural sizing) was excluded from the scope of this thesis due to time limitations. Material thickness values were set using some available reference values for the baseline concept (Table D.4), in consultation with experts at GKN Fokker Aerostructures.

The steps in Figure 6.1 are explained below:

- **Step 1**

As in the old workflow, the process starts with a manual step to define the product topology. Engineers use documents, meetings with other engineers, and their own experience to make decisions. The new addition here is that joints are also included.

- **Step 2**
This is an automatic step that generates different product models using MDM. This is again similar to the old workflow, but now joint and integrated part primitives are also included in these models. The KBE tool (MDM) was extended by the author to add these capabilities. This is in accordance with step 3 of the process to integrate MIM in a product that was shown in [Figure 4.1](#).
- **Step 3**
In this step, a DOE is set up with design variables from the manufacturing model of each manufactured primitive in the product. Therefore, these design variables control the material, manufacturing method, site and equipment for each primitive. The setup of the DOE involves deciding what options should exist for each design variable, and then updating a Python script that runs the DOE with these values. This is a manual step.
- **Step 4**
Once the DOE script is set up, it can be executed automatically. Since it uses the manufacturing model, this step is served by MIM. The validity of each design point in this DOE is decided by the compatibility analysis, which performs this check based on some included criteria. The compatibility criteria included in this thesis was previously shown in [Table 4.1](#).
- **Step 5**
This step is the same as in the old workflow, and it involves performing a sizing optimisation to set material thicknesses all parts in the product. To reiterate, this is a semi-automatic step, as the optimisation workflow has to be set up manually. The difference here is that, multiple valid design points are available from the manufacturing model DOE. Out of the four types of design variables in the manufacturing model DOE, only material selection has an influence on structural sizing (considering the capabilities at GKN Fokker Aerostructures). Therefore, the sizing has to be performed for each valid design point from the manufacturing model DOE, that has a unique set of material design variables.
- **Step 6**
This is an automatic step where the parameters of interest are calculated for all valid design points (from the manufacturing model DOE). Namely, these include the total cost, total mass and requirements compliance. The cost and mass calculation is handled by MDM, whereas the requirements compliance is handled by RVF. This step is performed by the same Python script that was used for the DOE, and it generates a spreadsheet as the output. Each row in this spreadsheet is a design point, and different values are stored in each column. For example, these include the selections made for the design variables, and the calculated results of the parameters of interest.
- **Step 7**
In this manual step, one or more of the valid design points from the manufacturing model DOE are selected to be further analysed for assembly. This selection can be made using the calculated parameters of interest from the last step. For example, valid design points with the least total cost and/or total mass can be selected by filtering the spreadsheet.
- **Step 8**
This is also a manual step, where a DOE that considers design variables from the assembly model is set up for each of the design points selected in step 7. Design variables from the assembly model can include, relations between operations (sequential / parallel) at in each operation set, inputs and outputs node values for each operation set etc. This DOE is also set up using a Python script, where the options for each design variables are set manually.
- **Step 9**
The assembly model DOE is executed automatically using the script, and the production rate is calculated for each valid design point. Here, the validity of design points is based on the assembly sequence feasibility. The included checks were previously discussed in step 4 of [Figure 3.11](#). Lastly, this step is dependent on MIM as it uses the assembly model. A similar spreadsheet is also generated as an output in this step.
- **Step 10**
Once the production rate is available for the valid design points, requirements compliance can be

automatically checked using RVF. In case the imposed requirements are not met for any of the valid design points, the process goes back to step 7 to select a different manufacturing model DOE design point, and the process is repeated.

- **Step 11**

In the last step, a final design point is selected based on the results from the assembly model DOE. A final check for the assembly sequence feasibility has to be performed manually by the user for the selected design. This is because, currently, the assembly model only checks for feasibility using a limited set of criteria. However, this step can also be fully automated in future with some additional work.

Lastly, some important aspects of the presented workflow will be highlighted:

- **It is not necessary to strictly follow the presented workflow**

The design workflow shown above represents the exact process that will be followed for the industrial use-case, but it does not represent the **only** way of using MIM. For example, it is not necessary to conduct a DOE study, as it can also be used for a single point design, where engineers have an appropriate idea of what choices will be made in the manufacturing/assembly model. These choices can then be automatically checked by MIM, and allow engineers to proceed further with their design.

- **The presented workflow is sequential in nature**

It can be observed that after moving ahead of a step that analyses a particular concept (design / manufacturing / assembly), it is not possible to account for the effect of a different choice at the previous steps. Therefore, it is possible that a potentially good design concept is missed, depending on what choices were made at the previous steps. For example, results from the assembly model DOE depend on what design point is selected from the manufacturing model DOE in step 7. This limitation of the presented workflow is recognised, but will not be dealt with within the scope of this thesis. Recommendations for overcoming this will be provided in the last chapter.

- **Step 5 for structural sizing has been skipped with the scope of this thesis**

Due to time limitations, step 5 of the workflow will not be conducted. Material thickness values for the included part primitives in the product will be set using some reference values from GKN Fokker Aerostructures. These will remain constant for a given part and material, and are shown in [Table D.4](#).

6.2. Workflow setup for the use-case

6.2.1. Product topology (step 1)

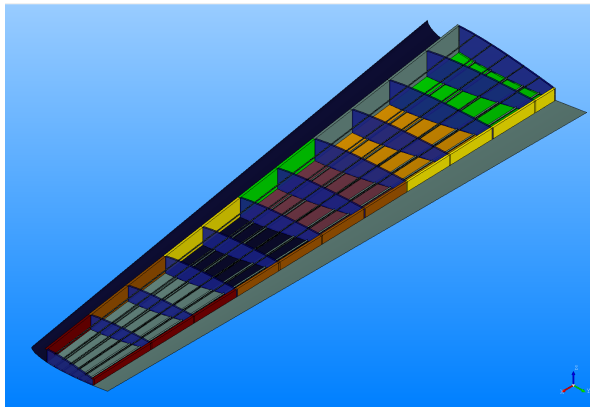


Figure 6.2: Wingbox part primitives in the industrial use-case. Note: the top skin panel has been hidden for clarity.

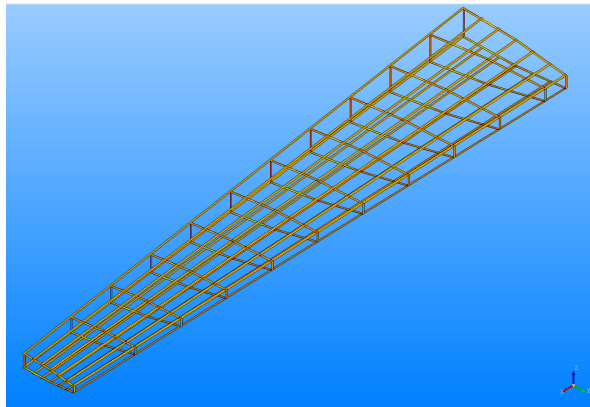


Figure 6.3: Wingbox joint primitives representation in the industrial use-case.

The part and joint primitives included in the product model of this use-case are shown in [Figure 6.2](#) & [Figure 6.3](#) respectively, and are listed below:

Part primitives:

- (1) Skin panels (quantity: 2)
- (2) Stringers (quantity: 10)
- (3) Ribs (quantity: 12)
- (4) Spars (quantity: 2)

Joint primitives:

- (i) Stringer-skin joints
- (ii) Rib-skin joints
- (iii) Spar-skin joints
- (iv) Rib-spar joints

It is to be noted that the product topology will remain fixed in the design workflow for this use-case. Therefore, the DOEs will only be used to explore the design space for different manufacturing and assembly concepts. The quantity, position etc. of the part and joints primitives will remain constant.

6.2.2. Manufacturing model DOE (step 3)

The design matrix for this DOE is shown in [Table 6.1](#). Design variables from only the material, design and manufacturing method information categories are considered to limit the total number of design points. Therefore, selections for the equipment set and manufacturing site information categories are kept as constants. These constant values have been manually set by the author in the DOE script in a way that they are always compatible with the selected manufacturing method and material. This is to ensure that they do not have an effect on the DOE results.

Additionally, part integration is also accounted for through the integrated-skin-stringer primitive. This primitive uses the blade stringers, and is manufactured using the Tapas layup process [20]. On the other hand, L-stringers are used with the skin panels (with the stringer skin joints). To summarise, these are the consequences for MIM and other analysis tools (such as cost/mass) when the “stringer” design variable is set to a blade stringer:

- Integrated-skin-stringers primitives are considered.
- (Individual) stringers are not considered.
- (Individual) skin panels are not considered.
- Stringer skin joints are not considered.

6.2.3. Structural sizing (step 5)

This step was excluded from the scope of this thesis due to time limitations. Material thickness values were set using some available reference values for the baseline concept ([Table D.4](#)), in consultation with experts at GKN Fokker Aerostructures. These thickness values remain constant throughout the design process.

Table 6.1: Design matrix of the manufacturing model DOE for the industrial use-case.

Info category	Primitive	Design variable values			Total
Material	Skin panels	CFRP ^a			1
	Ribs	CFRP	AL_7075 ^b		2
	Spars	CFRP	AL_7075		2
	Stringers	CFRP			1
	Integrated-skin-stringer	CFRP			1
Design	Stringer	L	Blade		2
Mfg. Method	Skin panels	Hand layup			1
	Ribs	Hand layup	TP form. ^c	Machining	4
	Spars	Hand layup	TP form.	Machining	3
	Stringers	Hand layup	TP form.		2
	Integrated-skin-stringer	Tapas layup [20]			1
	Stringer skin joints	Mechanical joint	Ind. weld. ^e		2
	Rib skin joints	Mechanical joint	Ind. weld.		2
	Spar skin joints	Mechanical joint	Ind. weld.		2
	Rib spar joints	Mechanical joint			1
Number of design points (product)					1536

^a CFRP: carbon-fibre-reinforced polymer^b AL_7075: Aluminium alloy 7075^c TP form.: Thermoplastic forming^d Rubber form.: Rubber forming^e Ind. weld.: Induction welding

6.2.4. Assembly model DOE (step 8)

A design point from the manufacturing model DOE has to be selected for the setup of this DOE. Details of the design point selected from the manufacturing model DOE are shown in [Table 6.2](#). Note that, the selection of this point will be discussed in detail in the results section, but these details have to be considered to discuss the setup of the assembly model DOE.

Table 6.2: Details of the design point selected from the manufacturing model DOE.

Design variables	Selection
Material of all part primitives	CFRP
Design of stringers	Blade
Manufacturing method - ribs and spars	Thermoplastic forming
Manufacturing method - integrated skin stringer	Tapas layup
Manufacturing method - all joints	Induction welding

Based on these details, the following stations are defined in the assembly model:

- station ribs manufacturing
- station spars manufacturing
- station integrated skin stringers manufacturing
- station rib skin joints
- station spar skin joints
- station rib spar joints

Note that, no stations are defined for the manufacturing of stringers and skin panels, or for stringer skin joints, as these primitives will not be considered because blade stringers are present in the selected design point.

The defined stations are kept constant in the assemble model DOE, however the arrangement of operations at each station, and how the stations are connected to each other is varied through the design variables. The design matrix showing this is shown in [Table 6.3](#).

The first two rows of this table include design variables to control relations between the operations at the stations. It can be noted that the stations have been grouped together in terms of part and joint operations. This has been done to limit the total number of design points in consideration of the available time.

The next section of rows in the table have design variables that define the input and output from each station for joints. The last section of rows is for the input and output from each part manufacturing station. It can be seen that only stations with joint operations have more than one option for these design variables. This is again to limit the number of design points.

Since the input and output for part manufacturing stations just contain one option, they remain constant in the DOE. From their defined value, it can be deduced that all part manufacturing stations are executed after the *product start* node in parallel, and have the output node set to *product_output_node*. The latter can then become an input to one or more of the joint stations.

6.2.5. RVF setup

The requirements imposed on the product in this use-case are shown in [Table 6.4](#).

Table 6.3: Design matrix of the assembly model DOE for the industrial use-case.

Design variable	Possible values		Total
operation relations at all part operations stations	sequential	parallel	2
operation relations at all joint operation stations	sequential	parallel	2
input: station rib skin joints	parts_output_node	output:spar skin joints	output:rib spar joints
output: station rib skin joints	product_end	output:rib skin joints	2
input: station spar skin joints	parts_output_node	output:rib skin joints	output:rib spar joints
output: station spar skin joints	product_end	output:spar skin joints	2
input: station rib spar joints	parts_output_node	output:rib skin joints	output:spar skin joints
output: station rib spar joints	product_end	output:rib spar joints	2
input: station ribs manufacturing	product_start		1
output: station ribs manufacturing	parts_output_node		1
input: station spars manufacturing	product_start		1
output: station spars manufacturing	parts_output_node		1
input: station integrated skin stringers manufacturing	product_start		1
output: station integrated skin stringers manufacturing	parts_output_node		1
Number of design points (product)			864

Table 6.4: Requirements imposed on product in the industrial use-case.

Type	ID	Stakeholder	Requirement text	Test case
Performance	R-1001	Client	The wingbox shall have a total cost of less than \$76000	CATMAC (cost tool)
Performance	R-1002	Client	The wingbox shall have a total mass of less than 215 kg	MPM (mass tool)
Performance	R-1003	Client	The wingbox shall have a number of shipsets per month of at least 7	Assembly model
Suitability	R-1004	Design team	The wingbox shall have passed all compatibility checks	Compatibility analysis
Suitability	R-1005	Design team	The wingbox shall have passed all manufacturing model validity checks	Manufacturing model

6.3. Results and discussion

6.3.1. DOE I: manufacturing model

The DOE resulted in a spreadsheet, with each row representing a design point in the study. A portion of this spreadsheet is shown in [Figure 6.4](#). The long spreadsheet has been split into three parts to improve readability. The design point index can be seen in column A of each sub-figure.

Selections made for each design variable for a given design point can be seen from columns D to P. Note that only design variables with at least two options (refer [Table 6.1](#)) are shown, as design variables with a single option remain constant.

The results from RVF include the requirement compliance check (True/False), and the compliance margin in percentage for all requirements in [Table 6.4](#), except for that on the production rate (which will be considered in the assembly model DOE). These can be seen in [Figure 6.4c](#). Note that a False requirement compliance check is represented by either an explicit "False" value, or by a blank cell.

For any further discussion, design points with a False compliance check for the requirement on compatibility (column Y) and manufacturing model validity (column Z) would be referred to as **"invalid"**. This is because these represent designs that cannot be manufactured, regardless of their cost or mass. On the other hand, design points that satisfy both of these requirement will be referred to as **"valid"**.

Lastly, the total cost and total mass can be seen in columns T and U respectively. Note that these have only been calculated for design points that are valid.

A	D	E	F	G	I	J	K
1	material_ribs	material_spar	material_string	spec_stringer	mfg_method_ribs	mfg_method_spars	mfg_method_stringers
1082	1336	AL_7075	CPS	I_stringer	mfg_method_rubber_forming	mfg_method_machining	mfg_method_tp_forming
1083	936	AL_7075	CPS	I_stringer	mfg_method_rubber_forming	mfg_method_tp_forming	mfg_method_tp_forming
1084	920	AL_7075	CPS	I_stringer	mfg_method_rubber_forming	mfg_method_hand_layout	mfg_method_tp_forming
1085	888	AL_7075	CPS	I_stringer	mfg_method_machining	mfg_method_tp_forming	mfg_method_tp_forming
1086	1084	AL_7075	CPS	blade_stringer	mfg_method_machining	mfg_method_tp_forming	mfg_method_tp_forming
1087	1085	AL_7075	CPS	blade_stringer	mfg_method_machining	mfg_method_tp_forming	mfg_method_tp_forming

(a) Portion of the spreadsheet showing results from the manufacturing model DOE - Part 1

A	N	O	P	T	U	Y
1	mfg_method_rib_skin_joints	mfg_method_spar_skin_joints	mfg_method_stringer_skin_joints	total_mass	total_cost	req_compatibility
1082	1336	mfg_method_mech_joint	mfg_method_mech_joint	227.24	73528.45	TRUE
1083	936	mfg_method_mech_joint	mfg_method_mech_joint	213.36	65688.5	TRUE
1084	920	mfg_method_mech_joint	mfg_method_mech_joint	213.36	66121.83	TRUE
1085	888	mfg_method_mech_joint	mfg_method_mech_joint	213.36	70996.32	TRUE
1086	1084	mfg_method_induction_welding	mfg_method_mech_joint	0	0	0
1087	1085	mfg_method_induction_welding	mfg_method_induction_welding	0	0	0

(b) Portion of the spreadsheet showing results from the manufacturing model DOE - Part 2

A	Y	Z	AA	AB	AC	AD
1	req_compatibility_check	req_mfg_model_check	req_total_mass_ch	req_total_mass_margin	req_total_cost_check	req_total_cost_margin
1082	1336	TRUE	FALSE	-5.694	TRUE	3.252
1083	936	TRUE	TRUE	0.764	TRUE	13.568
1084	920	TRUE	TRUE	0.764	TRUE	12.998
1085	888	TRUE	TRUE	0.764	TRUE	6.584
1086	1084					
1087	1085					

(c) Portion of the spreadsheet showing results from the manufacturing model DOE - Part 3

Figure 6.4: Screenshots showing a few design points in the resulting spreadsheet from the manufacturing model DOE. The spreadsheet has been split into three sub-figures to improve readability. The design point index can be referred to in column A in each sub-figure.

Trends in the DOE results

Out of the total design points considered in this DOE (1536), 240 are identified as valid. These are shown in various scatter plots in Figures 6.5, 6.6, 6.7 and 6.8.

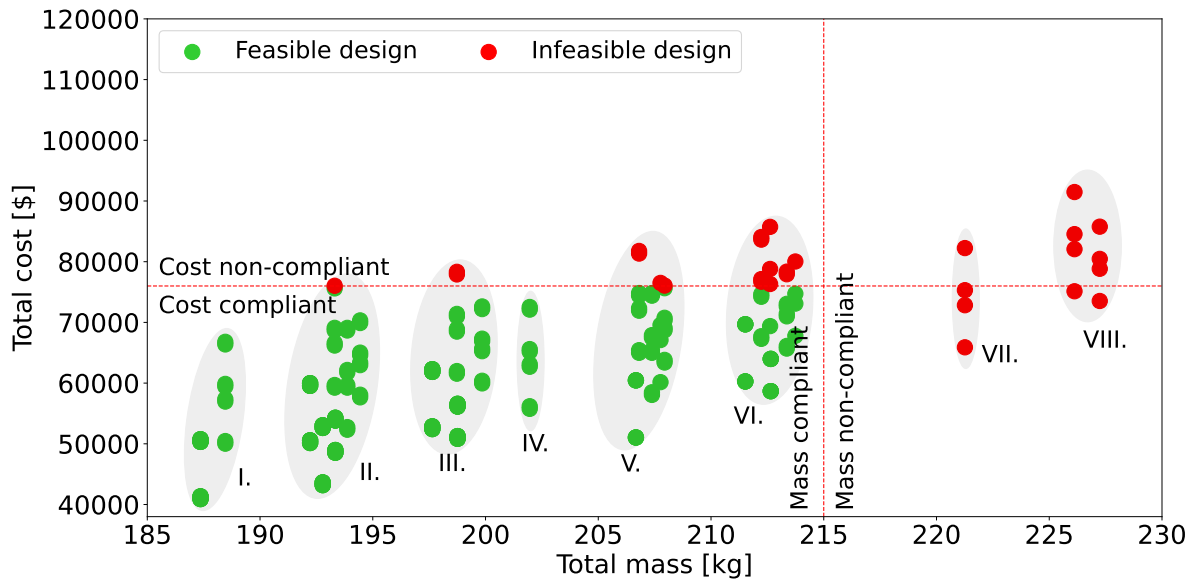


Figure 6.5: Scatter plot of all valid design points from the manufacturing model DOE. The requirements imposed on the total cost and total mass are represented by horizontal and vertical red lines in the plot respectively. All design points that satisfy these requirements are referred to as feasible, while the others are infeasible. Eight groups of design points, each with very similar total mass can be identified. Total cost is calculated by using CATMAC.

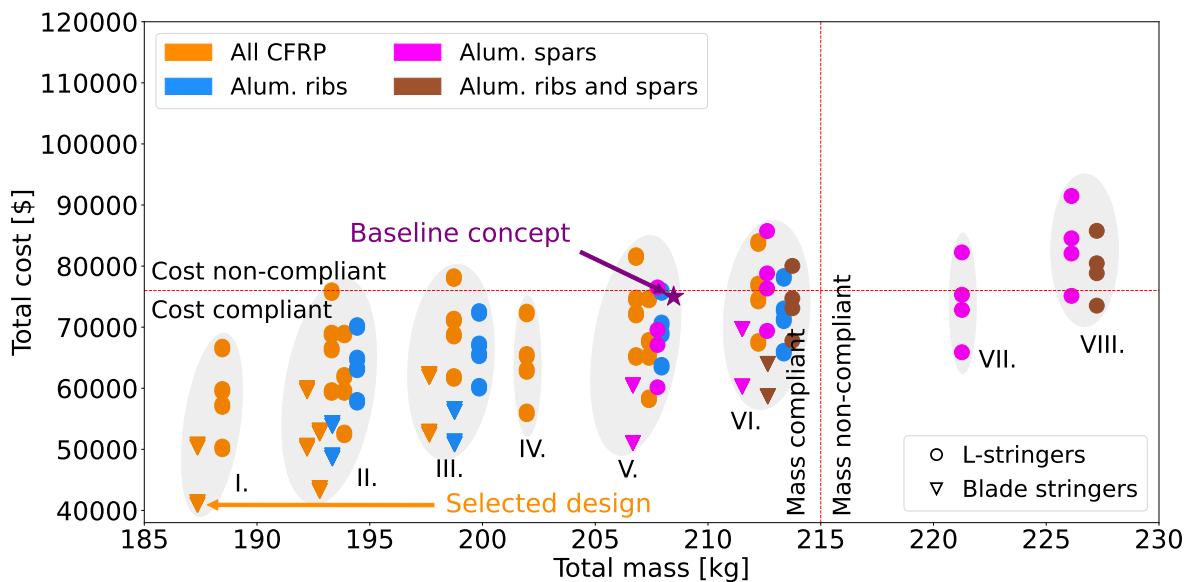


Figure 6.6: Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected part material. The default material for all parts is CFRP, unless stated otherwise for some parts by the legend key. The baseline design and design selected for the assembly model DOE are marked with an arrow. Two types of markers are used for the design points (circle/triangle), based on the selection made for the stringer design.

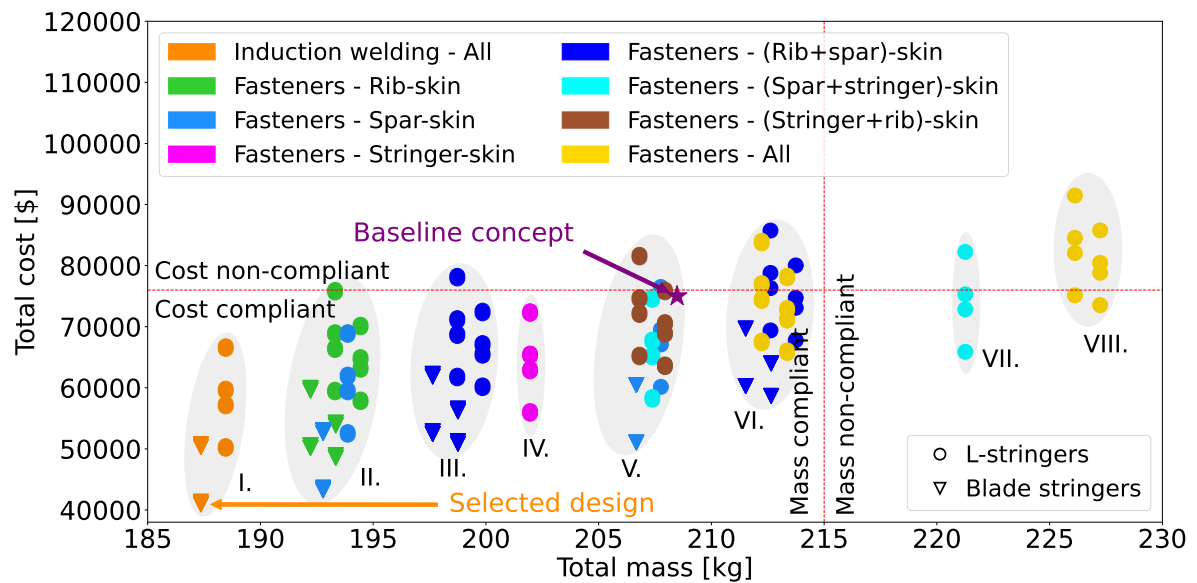


Figure 6.7: Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected manufacturing method for joints. The default joining method is induction welding, unless stated otherwise for some joints by the legend key. It can be observed that the joining method is closely related to the identified eight mass groups.

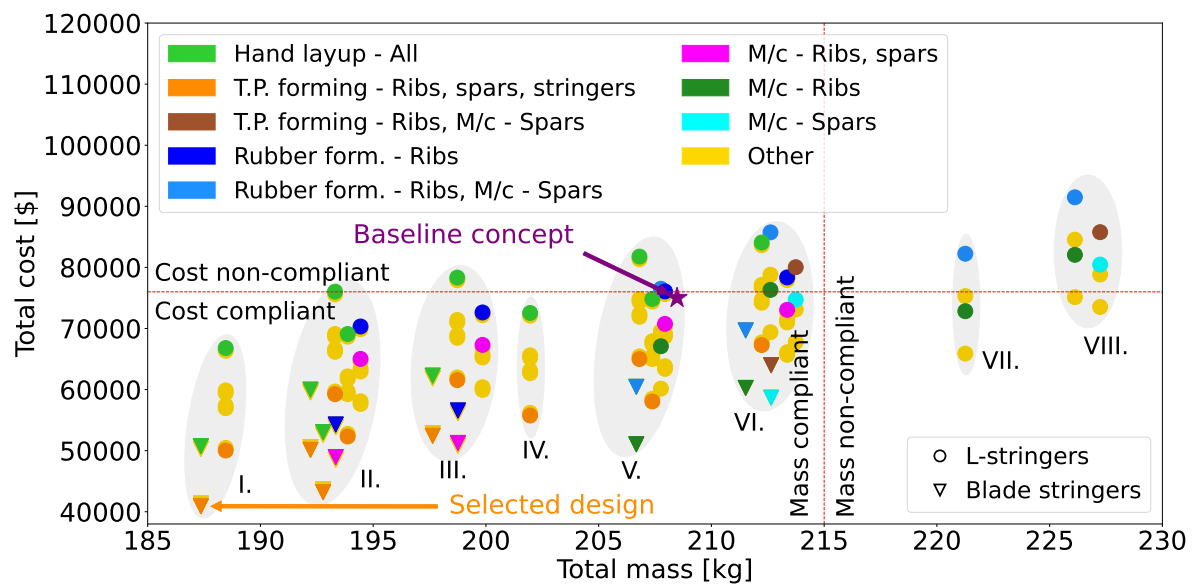


Figure 6.8: Scatter plot of all valid design points from the manufacturing model DOE, labelled based on the selected manufacturing method for parts. The default part manufacturing method is hand layup, unless stated otherwise for some parts by the legend key. No correlation between the part manufacturing method and mass groups can be observed. A relation between the part manufacturing method and total cost can be observed within each mass group. Note that not all possible combinations of selections are shown in the legend due limitation of space. These missing combinations are shown together with the legend key "Other". T.P. = Thermoplastic; M/c = Machining.

In all scatter plots, horizontal and vertical red lines mark the boundary of requirement compliance for the total cost and total mass. All design points that satisfy these requirements are referred to as feasible, while the others are infeasible. The first scatter plot (Figure 6.5) shows all valid design points, and marks them as feasible and infeasible. The second plot (Figure 6.6) marks all valid design points

based on the selected part material. The third plot (Figure 6.7) marks all valid design points based on the joining method. And the last plot (Figure 6.8) marks all valid design points based on the selected part manufacturing method. The baseline design, and the selected design for the assembly DOE (to be discussed later) are also marked with arrows in the last three scatter plots. Furthermore, two types of markers are used for the design points (circle/triangle) in these plots, based on the selection made for the stringer design (L or blade).

Some trends that can be observed in the scatter plots are discussed below:

- **The eight “mass groups”**

Eight groups of design points (labelled I. through VIII. in the plots) with very similar total mass can be observed in the scatter plots. The existence of these groups can be explained based on the selections for part material (Figure 6.6) and joining method (Figure 6.7) for the design points. It can be observed that these two plots have similar coloured nodes in the mass groups, indicating a relation between these choices and the mass groups. This relation is especially strong based on the selected joining method, as nodes of almost uniform colour can be observed in each mass group in Figure 6.7.

The “jump” from one mass group to the next can also be explained using Figure 6.7. The first mass group has the least mass, and has all induction welded joints (therefore no additional mass). The mass jumps to group II when fasteners are introduced for either rib-skin or spar-skin joints. When fasteners are used for both of these joints, the total mass moves to group III. The next group has fasteners for all stringer-skin joints. These joints lead to a higher mass than the previous joints because of the long joint length of stringers, combined with their large quantity (10) in the product model. This leads to a larger quantity of fasteners. A similar pattern can be followed for the rest of the groups.

Another interesting observation is that the baseline concept (which has all fastener based joints), falls in group V, instead of group VI or VIII, which have design points with all fastener based joints. This indicates that the factor that was used to account for the joint mass in the baseline concept (10% of total wing mass table D.1) may not be sufficient.

- **Effect of part integration**

The effect of part integration on the total cost and mass can be observed in the scatter plots. For example, consider mass group I. In this group, the part materials and joining methods are the same for all design points. All part materials are CFRP, and all joints use induction welding. Now consider the two green design points in this mass group in Figure 6.8. These have total costs approximately equal to \$50000 and \$68000, and in both, all parts are manufactured using the hand layup method.

But still, a difference in the total cost can be observed between these design points. The only difference between these two design points is that the one with the lower cost has blade stringers, whereas the other one has L-stringers. The specification of blade stringers means that they have been integrated with the skin panels, and therefore, there is no cost associated with the manufacturing of stringers or stringer-skin joints. This explains the cost difference between these two design points.

- **Correlation between total cost and part manufacturing method in each mass group**

Although there is no correlation between mass and the part manufacturing method, a relation between the latter and total cost can be observed. For example, it can be observed that in most of the mass group in Figure 6.8, the hand layup method is the most expensive, while thermoplastic forming is usually one of the cheapest options. Design points with machined parts can also be found closer to the least expensive options.

- **Correlation between total cost and total mass**

A correlation between the total cost and mass can be observed from Figure 6.5. The total cost increases or decreases with a similar change in the total mass. This is in contrast to the expected relation between these two parameters. For example, the use of composite materials may offer a lower mass, but these can cost more as they are more complex to work with, require tooling etc., in comparison to metallic parts. Some possible reasons for the observed trend in the scatter plots are:

- Non-recurring cost associated with the manufacturing methods is not considered as it is not supported by CATMAC. Therefore, the cost is limited to that of the materials, labour and machine usage. Manufacturing methods like the Tapas layup can require extensive tooling [20]. Therefore, not considering these aspects can give an inaccurate estimate for the actual cost.
- The cost model is not accurate (for the recurring cost). This aspect has been noted in [24] as well.

Selection of a design point for assembly model DOE

As per step 7 of Figure 6.1, one or more design points from the manufacturing model DOE have to be selected, for analysis in the assembly DOE. For this, the manufacturing model DOE result spreadsheet was filtered to only keep design points that comply with all requirements. These are the design points that are valid and feasible. The rows were then sorted to bring in the ascending order of the total cost and total mass, bringing designs with the lowest values for these to the top of the spreadsheet. The top 8 design points in this spreadsheet are shown in Figure 6.9.

Different groups of design points with identical values for total mass and cost can be observed next in this figure. These have been highlighted in green light and purple colours. The total mass is the same for all of them as their part materials and manufacturing methods for joints remain constant.

It can be observed that the total cost varies from one colour group to the next, but remains identical in the group. This is because selection of blade stringers indicate that no separate stringer parts exist in the product as they have been integrated with the skin panels. Therefore, selections made for the design variables stringer-skin joints and manufacturing method of stringers becomes irrelevant in these cases, resulting in identical cost within the group.

The total cost changes from one group to the next because of a change in the manufacturing method of another primitive in the product. Going from the green group to purple, the manufacturing method of spars changes from thermoplastic forming to hand layup (highlighted in dark purple in Figure 6.9a).

Based on the above discussion, the design point with index = 271 is selected for the assembly model DOE, as it offers the least values for the total cost and total mass. It is highlighted in dark green in Figure 6.9. Note that any design point from the light green group could have been selected, as they are equivalent designs. The point 271 has been selected in particular, as it has a uniform selection for the part and joints manufacturing methods. This point is also marked in all scatter plots with an arrow in orange colour. Some key characteristics of this design point are that stringers have been integrated with the skin panels, all joints to the skins use induction welding, all parts materials are composites, and the ribs & spars are manufactured using thermoplastic forming.

	A	D	E	F	G	I	J	K
1		material_ribs	material_spar	material_string	spec_stringer	mfg_method_ribs	mfg_method_spars	mfg_method_stringers
2	262	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_tp_forming	mfg_method_hand_layout
3	263	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_tp_forming	mfg_method_hand_layout
4	270	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_tp_forming	mfg_method_tp_forming
5	271	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_tp_forming	mfg_method_tp_forming
6	246	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_hand_layout	mfg_method_hand_layout
7	247	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_hand_layout	mfg_method_hand_layout
8	254	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_hand_layout	mfg_method_tp_forming
9	255	CPPS	CPPS	CPPS	blade_stringer	mfg_method_tp_forming	mfg_method_hand_layout	mfg_method_tp_forming

(a) Part 1

	A	N	O	P	T	U	Y
1		mfg_method_rib_skin_joints	mfg_method_spar_skin_joints	mfg_method_stringer_skin_joints	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
2	262	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
3	263	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
4	270	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
5	271	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
6	246	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
7	247	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
8	254	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints
9	255	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_induction_welding	mfg_method_mech_joints	mfg_method_mech_joints	mfg_method_mech_joints

(b) Part 2

	A	Y	Z	AA	AB	AC	AD
1		req_compatibility_check	req_mfg_model_check	req_total_mass_check	req_total_mass_margin	req_total_cost_check	req_total_cost_margin
2	262	TRUE	TRUE	TRUE	12.854	TRUE	46.176
3	263	TRUE	TRUE	TRUE	12.854	TRUE	46.176
4	270	TRUE	TRUE	TRUE	12.854	TRUE	46.176
5	271	TRUE	TRUE	TRUE	12.854	TRUE	46.176
6	246	TRUE	TRUE	TRUE	12.854	TRUE	45.606
7	247	TRUE	TRUE	TRUE	12.854	TRUE	45.606
8	254	TRUE	TRUE	TRUE	12.854	TRUE	45.606
9	255	TRUE	TRUE	TRUE	12.854	TRUE	45.606

(c) Part 3

Figure 6.9: Selected design point from the manufacturing model DOE (index 271 - in dark green), for analysis in the assembly model DOE.

6.3.2. DOE II: assembly model

Results from the assembly model DOE were also captured in a spreadsheet, and Figure 6.10 shows an excerpt from the same. Columns C to J show selections for the design variables, and columns R to AG capture results for each design point execution. Blank cells in columns R to Z indicate an invalid definition of the assembly model based on the set of design variables. The corresponding error message generated in the model is captured in column AG. Therefore, valid design points can be identified in this spreadsheet by filtering out the rows with an error. Furthermore, designs that comply with the requirement on the production rate can be identified by applying a filter on column Y.

A	C	D	E	F	G
	oper_relation_part_station	oper_relation_joint_station	input_rib_skin_joints	output_rib_skin_joints	input_spar_skin_joints
40	sequential	sequential	parts_output_node	rib_skin_joints_output	parts_output_node
53	sequential	sequential	parts_output_node	rib_skin_joints_output	rib_skin_joints_output
58	sequential	sequential	parts_output_node	rib_skin_joints_output	rib_skin_joints_output
63	sequential	sequential	parts_output_node	rib_skin_joints_output	rib_skin_joints_output
75	sequential	sequential	spar_skin_joints_output	product_end	parts_output_node
99	sequential	sequential	spar_skin_joints_output	product_end	rib_spar_joints_output

(a) Assembly model DOE result spreadsheet - Part 1

A	H	I	J	R	Y	Z	AG
	output_spar_skin_joints	input_rib_spar_joints	output_rib_spar_joints	shipsets_per	req_shipset	req_margin	error_as
40	product_end	spar_skin_joints_output	product_end				InvalidAssembly
53	product_end	spar_skin_joints_output	rib_spar_joints_output				InvalidAssembly
58	spar_skin_joints_output	spar_skin_joints_output	product_end	3.434730773	FALSE	-50.932	
63	product_end	rib_skin_joints_output	rib_spar_joints_output	3.434730773	FALSE	-50.932	
75	product_end	rib_skin_joints_output	rib_spar_joints_output				InvalidAssembly
99	product_end	rib_skin_joints_output	rib_spar_joints_output				InvalidAssembly

(b) Assembly model DOE result spreadsheet - Part 2

Figure 6.10: Screenshots of assemble model DOE results. The two sub-figures show a single design point, with the design point index shown in column A.

A	C	D	E	F	G
	oper_relation_part_station	oper_relation_joint_station	input_rib_skin_joints	output_rib_skin_joints	input_spar_skin_joints
751	all_parallel	all_parallel	spar_skin_joints_output	product_end	rib_spar_joints_output
841	all_parallel	all_parallel	rib_spar_joints_output	rib_skin_joints_output	rib_skin_joints_output
706	all_parallel	all_parallel	parts_output_node	rib_skin_joints_output	rib_skin_joints_output
711	all_parallel	all_parallel	parts_output_node	rib_skin_joints_output	rib_spar_joints_output
764	all_parallel	all_parallel	spar_skin_joints_output	rib_skin_joints_output	parts_output_node
803	all_parallel	all_parallel	rib_spar_joints_output	product_end	parts_output_node
332	sequential	all_parallel	spar_skin_joints_output	rib_skin_joints_output	parts_output_node
371	sequential	all_parallel	rib_spar_joints_output	product_end	parts_output_node
274	sequential	all_parallel	parts_output_node	rib_skin_joints_output	rib_skin_joints_output
279	sequential	all_parallel	parts_output_node	rib_skin_joints_output	rib_spar_joints_output
319	sequential	all_parallel	spar_skin_joints_output	product_end	rib_spar_joints_output
409	sequential	all_parallel	rib_skin_joints_output	rib_skin_joints_output	rib_skin_joints_output

(a) Part 1

A	H	I	J	R	Y	Z
	output_spar_skin_joints	input_rib_spar_joints	output_rib_spar_joints	shipsets_per_mo	req_shipsets_per	req_margin
751	spar_skin_joints_output	parts_output_node	rib_spar_joints_output	23.93263407	TRUE	241.895
841	product_end	parts_output_node	rib_spar_joints_output	23.93263407	TRUE	241.895
706	spar_skin_joints_output	spar_skin_joints_output	product_end	18.37232017	TRUE	162.462
711	product_end	rib_skin_joints_output	rib_spar_joints_output	18.37232017	TRUE	162.462
764	spar_skin_joints_output	rib_skin_joints_output	product_end	18.37232017	TRUE	162.462
803	spar_skin_joints_output	spar_skin_joints_output	rib_spar_joints_output	18.37232017	TRUE	162.462
332	spar_skin_joints_output	rib_skin_joints_output	product_end	9.575182545	TRUE	36.788
371	spar_skin_joints_output	spar_skin_joints_output	rib_spar_joints_output	9.575182545	TRUE	36.788
274	spar_skin_joints_output	spar_skin_joints_output	product_end	9.401496115	TRUE	34.307
279	product_end	rib_skin_joints_output	rib_spar_joints_output	9.401496115	TRUE	34.307
319	spar_skin_joints_output	parts_output_node	rib_spar_joints_output	9.401496115	TRUE	34.307
409	product_end	parts_output_node	rib_spar_joints_output	9.401496115	TRUE	34.307

(b) Part 2

Figure 6.11: Selected design point from the assembly model DOE (index 274 - in white text).

The spreadsheet was then filtered to only keep results with a valid assembly model, that also pass the imposed requirement on shipsets per month. The result is shown in Figure 6.11. Design points with identical production rate are highlighted with the same colour, and these include the green, purple, orange and blue groups. The reason behind identical production rate becomes clear from the station graphs. For example, consider the station graphs of design points in the green group, 751 and 841, shown in Figure 6.12 and Figure 6.13 respectively. The only difference in these graphs is the arrangement of the rib-skin and spar-skin joints stations. Since these stations lie on the same sequence, their order does not affect the critical manufacturing path, and hence the production rate is identical. The same reason holds for all other design points with identical production rates.

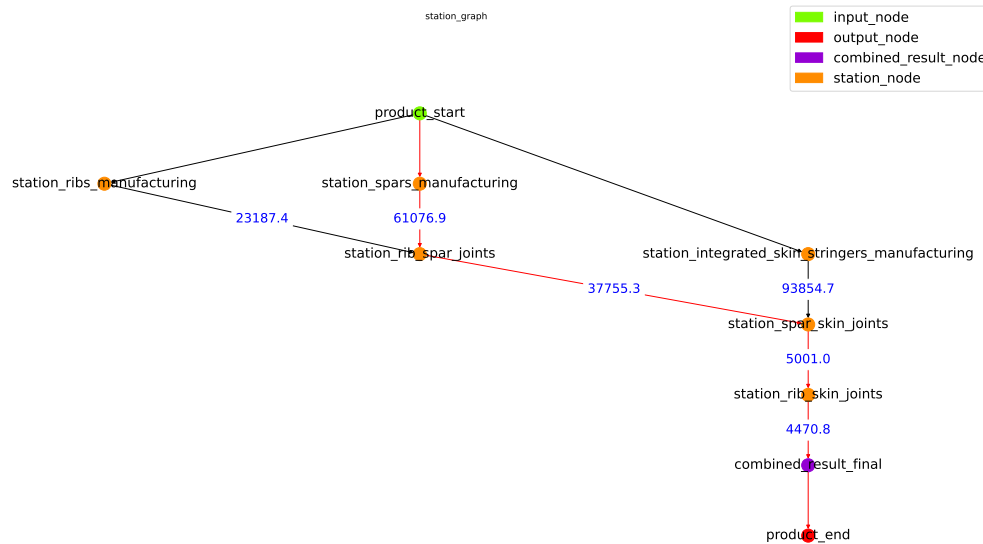


Figure 6.12: Station graph of design point 751 in the assembly model DOE.

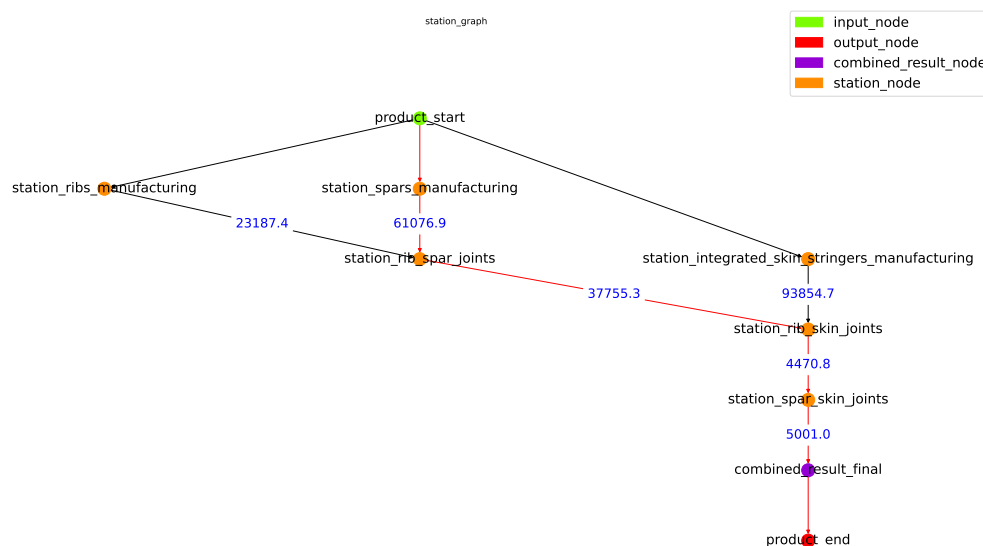


Figure 6.13: Station graph of design point 841 in the assembly model DOE.

Next, it must be noted that the filtered design points in Figure 6.11 may still represent infeasible assembly sequences, as currently, only a limited number of feasibility checks are performed by the assembly model (discussed in step 4 of fig. 3.11). Therefore, the final result will need to be verified by the user. Consider the station graphs of design points 751 and 841 again. By analysing the station arrangement it can be observed that both of these are infeasible, as the rib-spar joints are executed first, which is followed by either rib-skin or spar-skin joints. This second step will not be possible with skins on both sides, as it would lead to closure of the box section of the wing, making it impossible to remove the tooling used in the induction welding process.

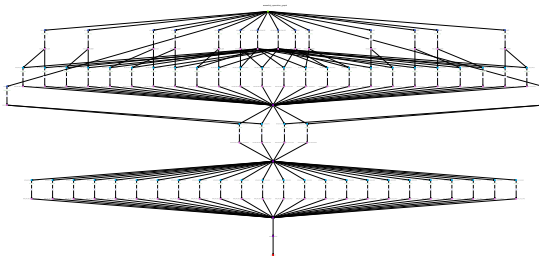


Figure 6.14: Operation graph of design point 706 in the assembly model DOE. The reader is suggested to zoom-in in the digital version of this report.

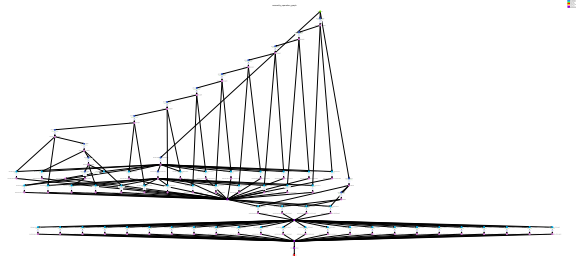


Figure 6.15: Operation graph of design point 274 in the assembly model DOE. The reader is suggested to zoom-in in the digital version of this report.

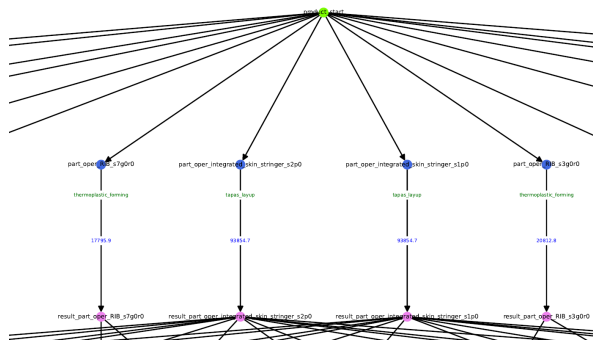


Figure 6.16: Operation graph of design point 706 in the assembly model DOE. Zoom level 1.

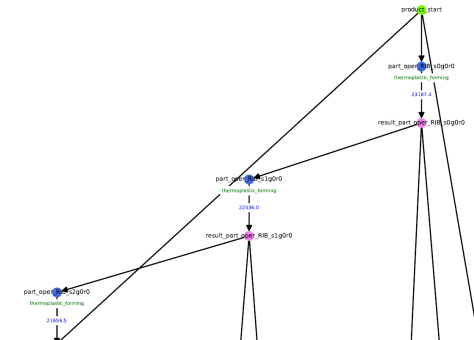


Figure 6.17: Operation graph of design point 274 in the assembly model DOE. Zoom level 1.

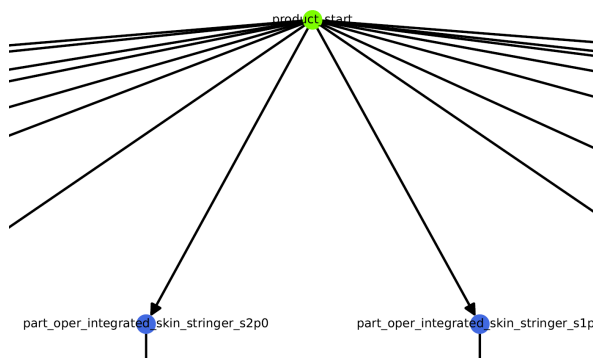


Figure 6.18: Operation graph of design point 706 in the assembly model DOE. Zoom level 2.

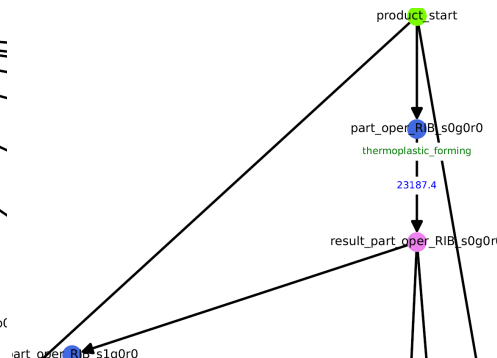


Figure 6.19: Operation graph of design point 274 in the assembly model DOE. Zoom level 2.

Similarly, on checking the remaining design points, the feasible design points in Figure 6.11 include 706, 711 (dark purple), and 274, 279 (dark blue). One design point from each pair can be selected, as each has the same production rate. Consider the points 706, and 274. These design points only

differ in the design variable that defines operation relations for part manufacturing stations (column C in Figure 6.11). This difference can be visually observed in the design point's operation graphs, shown in Figure 6.14 and Figure 6.15.

Although it is not possible to read these graphs in the printed version of this report, a clear difference in the arrangement of operations from the top (product start node) to the bottom (product end node) can be observed for the two. For design point 706, all part operations occur in parallel after the product start node, whereas, for 274, part operations occur in sequence for each part category. Zoomed-in versions of these graphs showing the top portion at two levels of zoom are also shown in figures 6.16 and 6.18 for design point 706, and in figures 6.17 and 6.19 for design point 274.

Finally, a selection can be made for the best design point between 706 and 274. The one with the higher production rate, and therefore higher requirement compliance margin may seem to be the obvious choice at first. But if just-in-time manufacturing is important for the client, then a compliance margin close to zero may be more desirable. Therefore, design point 274 is selected as the one that best satisfies the imposed requirements from this study.

Lastly, it should be noted that in case no valid design points that meet the imposed requirement could be found in this DOE, a different design point from the manufacturing model DOE would have to be selected, and the process repeated. The process can also be repeated to analyse the assembly process of a different manufacturing model design point, but the analysis will just be limited to one manufacturing model design point in this thesis due to time limitations.

6.3.3. Identification of unapparent designs - research question 2.a.i

Some unapparent design points can be identified in the manufacturing model DOE results from the scatter plot based on part material (Figure 6.6). In general, it is expected that design points with parts made from composite materials would lie towards the left side of the plot, while the aluminium parts are towards the right. This holds true for the spars, as all design points with spar material as aluminium are only present in mass groups V, VI, VII and VIII.

But the same is not true for ribs, as several design points can be identified with aluminium ribs (in blue) that lie in mass groups II and III. Between the blue points of mass groups II and III, some orange (all composite) points can be seen. This means the total wing mass of the blue points (with aluminium ribs) in mass group II is less than the total mass of some design points that have all composite parts.

Additionally, because induction welding cannot be used with metal parts, the ribs in these design points would be connected using mechanical fasteners. The latter is associated another penalty on the total wing mass compared to induction welding. Even then the total mass of these design points is lower than those that have composite parts. Therefore, the design points with metal ribs in mass groups II and III are found to be unapparent. This was verified with experts at GKN Fokker Aerostructures.

The reason behind the lower mass of these design points can be explained by comparing two design points, one "unapparent" blue design point from group II, and one "ordinary" orange design point from mass group III. Circular design points will be chosen for this comparison (an arbitrary choice), which represent L-stringers. One of the unapparent design points has an index 939. The index of an ordinary design point in group III is 256.

First, it must be established that the total mass of composite ribs is actually less than the total mass of aluminium ribs, based on their selected thicknesses. This value can be derived from MDM. The total mass for aluminium ribs is 16.41kg, and that of CFRP ribs is 15.28kg. Therefore, the unapparent design point does not have a lower mass just because of a variation in thickness and material.

By considering Figure 6.7, it can be observed that the unapparent design points in group II have fastener based joints only for rib-skin. Whereas, all design points in group III have fastener based joints for rib-skin and spar-skin. This is the main reason behind these unapparent design's lower mass. These unapparent designs have induction welding as the joining method for spar-skin, which makes them lighter compared to other designs.

Lastly, no unapparent designs were found from the assembly model DOE.

6.3.4. Identification of better designs - research question 2.a.ii

Designs that are "better" in terms of cost or mass can be identified using the scatter plots from the manufacturing model DOE. The baseline concept has been marked in figures 6.6, 6.7 and 6.8 with a purple arrow. This baseline concept falls in mass group V, and is right next to the total cost compliance boundary.

No specific region for “better” designs has been marked in these scatter plots. This is because, depending on the needs of the stakeholders, designs can be identified that may have a lower total cost, and/or lower or higher total mass, compared to the baseline. It may be that the design that is closest to the requirement of maximum total mass fits the needs of the stakeholders better, while still being compliant. This would mean that a design with a higher mass is selected.

The scatter plots, along with the spreadsheet, provide users the necessary tools to understand trends in the design space, and to rank designs based on their needs. This allows identification of designs that may be “best” for them, relative to the baseline concept.

Lastly, due to the unavailability of baseline data for assembly sequence or production rate, a comparison cannot be made with the results from the assembly model DOE.

6.3.5. Effect of production considerations on gauging product viability - research question 2.b

Part integration

To see how part integrated affects parameters of a product, two baseline cases with different types of joints will be compared to a case where parts have been integrated. For this analysis, it is considered that the material of all parts is CFRP, and they are manufactured using the hand layup method. Additionally, only L-stringers are considered, and mechanical rib-spar joints are always included in the analysis. Only total cost and mass are considered as they are the only relevant parameters of interested available within the scope.

Figure 6.20 shows the total mass and cost of three product definitions. There are two baseline cases, that each include L-stringers, skin panels and a joint between the two in the product definition. A case where the stringers and skin panels are integrated into a single part is also included in the plot.

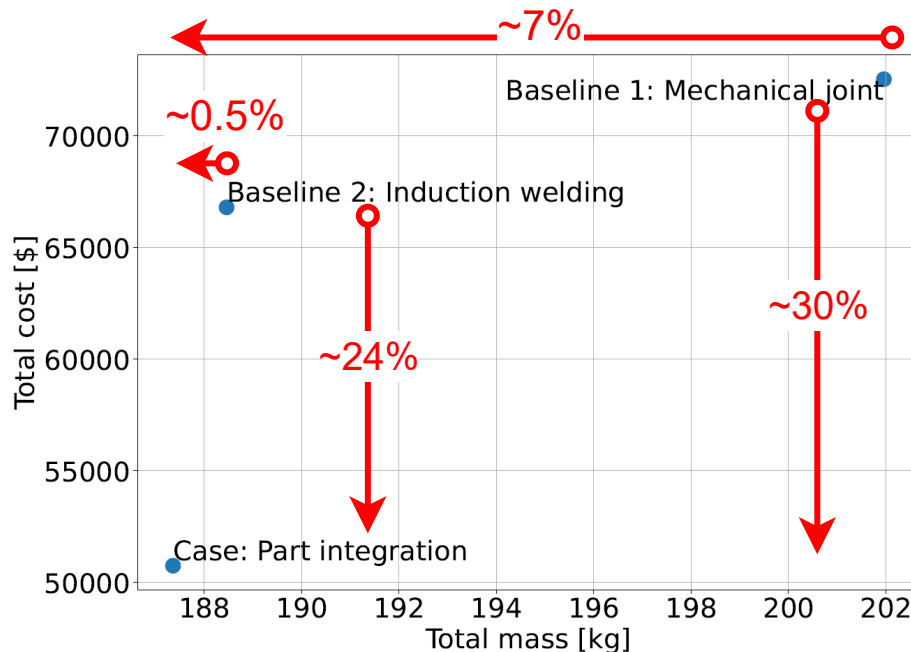


Figure 6.20: Effect of accounting for part integration in a product on the total cost and mass.

There is a considerable difference between both the total cost and mass of the case with part integration and baseline 1 with mechanical joints. On the other hand, the difference between part integration and induction welding is mostly limited to cost, as the induction welding process does contribute to the total mass of the product. The slight difference between the total mass of these cases can be attributed to the additional flange area of L-stringers in the induction welding case. With part integration, blade stringers are directly integrated with the skin in the layup process.

Overall, the integrated part costs approximately 24% less than the induction welded case, and

30% less compared to mechanical joints. The total mass for the integrated part is 0.5% and 7% less compared to induction welding and mechanical joints respectively. Except for the small mass difference between the integrated part and induction welded parts, the presented differences are not small enough to neglect this production considerations in the design process. Without them, engineers cannot be confident if their designs would meet the set requirements on these parameters.

Joints

The effect of joints on the product will be checked by analysing the contribution of joints to the available parameters (total cost and mass). For this, design points with different types of joining methods will be selected from the manufacturing model DOE, and the contribution of joints to the total cost and mass will be analysed. This is shown in [Table 6.5](#).

Note that each row represents one design point. The default joining method is induction welding, unless stated otherwise for some joints in the first column. All design points have part material as CFRP, have L-stringers, and always include mechanical rib-spar joints.

It can be observed that the contribution of joints to the total mass and joints increases as more fastener based joints are used in the product. Their contribution to the total cost is especially high, reaching almost 50% for the case with all fasteners. Whereas, their contribution to the total mass varies between 0.44% to 8.41%, depending on how many joints use fasteners. The large contribution of joints to the total cost indicates the importance of modelling them accurately, instead of relying on approximate factors that estimate their contribution.

As for the contribution to the total mass, the table shows that, again, it is important to consider the exact specification of the different joining methods. This is important to accurately calculate the total mass of the product, as their contribution can vary a lot based on the selected joining method. For example, a broad factor of 5 to 10% is suggested in [\[1\]](#) to account for joints. Though a lot of the cases in [Table 6.5](#) fall in this range, using such approximations can lead to overly conservative, or risky assessments of the total mass.

Assembly sequence and compatibility

Lastly, the effect of compatibility and assembly sequence will be discussed qualitatively only, as no related parameters of interest would be available if they are excluded from the design.

Not including assembly sequence would mean that no insights about the steps to manufacture the product will be available. Feasibility of product assembly would also have to be analysed completely manually, as shown by step 1c in [Figure 2.2](#). Additionally, parameters related to assembly, such as the production rate will not be available automatically in the design process. Similarly, not including compatibility analysis would mean that the related engineering rules would have to be manually included by the engineers to ensure validity of their designs (through the iterative steps 1b and 1c in [Figure 2.2](#)).

Overall, not automatically accounting for assembly sequence and compatibility can lead to bottlenecks in the design process at the conceptual design stage, as these steps then have to be completed manually in an iterative process. Thus, the number of options that can be explored get severely limited.

Table 6.5: Breakup of total cost and total mass to show contribution of joints, for some design points with different joining methods. All design points have part material as CFRP, have L-stringers, and mechanical rib-spar joints are always included. The default joining method is induction welding, unless stated otherwise for some joints in the first column.

Joining method in selected design point	Total cost [\$]	Total cost [\$]	Total joints cost [%]	Total mass [kg]	Total joints mass [kg]	Total joints mass [%]
Induction welding - All	66801.94	17782.19	26.62	188.46	0.83	0.44
Fasteners - Rib-skin	66621.25	27014.83	40.55	193.31	5.68	2.94
Fasteners - Spar-skin	69099.78	20080.03	29.06	193.87	6.24	3.22
Fasteners - Stringer-skin	72539.62	23519.87	32.42	201.96	7.58	3.75
Fasteners - (Rib+spar)-skin	71398.00	29312.67	41.06	198.73	11.10	5.59
Fasteners - (Spar+stringer)-skin	74837.45	25817.70	34.50	207.38	12.99	6.26
Fasteners - (Stringer+rib)-skin	72358.93	32752.51	45.26	206.81	12.44	6.02
Fasteners - All	74656.76	35050.34	46.95	212.23	17.85	8.41

6.3.6. Addressing limitations of the state-of-the-art

This section will discuss how limitations identified for the state-of-the-art approaches in [chapter 2](#) are addressed by MIM. This is discussed per limitation below:

- **Bottlenecks in the manual approach**

The main problematic steps of the design workflow in [Figure 2.2](#) are related to coming up with a valid starting point for the design (steps 1a, 1b, 1c), and to manually account for joints and part integration (steps 5a, 5b, 6). Both of these problems are addressed by MIM. A large set of valid design points were generated from the manufacturing model DOE. The manual and iterative of steps (1a, 1b, 1c) in the old workflow were removed with the use of compatibility analysis, and the assembly model, that automatically take these considerations into account. Lastly, joints and part integration were accounted for by updating the KBE tool's capabilities.

- **Accounting for production considerations only through associated cost [[4–6](#)]**

This limitation has been overcome by successfully including parameters other than the manufacturing cost using MIM. Namely, these parameters are the mass, production rate, and requirements compliance.

- **Non-generic solutions [[7–10](#)]**

Successful integration of MIM has been shown with a variety of product primitives in the verification and validation case, and the industrial use-case. Since MIM works with a KBE tool, there is a direct link to CAD geometry, and the methodology is not limited to the definition of specific products that are defined using just some parametric rules. Furthermore, the MIM database can be easily extended to include more options for the manufacturing methods, materials, site, and equipment. Lastly, MIM can be integrated with other ParaPy based KBE tools to work with entirely different products by following the steps shown in [Figure 4.1](#). However, it should be noted that these steps have not been verified on a use-case.

- **Methodology depends heavily on the use of some commercial software that don't support automation, or are unsuitable for conceptual design [[10–12](#)]**

No commercial analysis tools have been used in this thesis. Mass is calculated using an in-house tool at GKN Fokker Aerostructures (MPM), cost is calculated using an open source cost estimation tool (CATMAC) which can be easily extended to include new manufacturing methods, and the compatibility analysis and assembly model for production rate are ParaPy based tools that have been developed by the author.

- **Manufacturing and assembly are not considered in the same study [[5](#), [9](#), [13](#), [14](#)]**

Aspects of both manufacturing and assembly have been considered in this thesis through the manufacturing and assembly models respectively.

7

Conclusions and Recommendations

Considerations that decide producibility of a design form an important part of the design process, and must be included in the early stages of design to ensure that these designs can be produced within the set requirements. A methodology to capture these production considerations in a model, and to automatically include them in the conceptual design process has been presented in this thesis. It manages to automate the manual iterative steps in the state-of-the-art design process. The following conclusions can be drawn from this work:

- **The MIM provides a generic structure to capture and organise production related information in a product system**

- It has been seen from the verification and validation case, and from the industrial use-case, that the manufacturing model can be defined for a variety of product primitives, including parts, joints and integrated parts, without making any changes to its generic structure. The five information categories are able to capture all production related information regardless of the type of primitive, while maintaining a link to the primitive's design model.
- The manufacturing model is not limited by what selections can be made in each information category, as any (new) material, manufacturing method, site or equipment can be included, as long as a corresponding entry is also added in the MIM database for the same.
- The assembly model can also work with different types of product primitives. This is because the assembly model primarily depends on the manufacturing model, and the product definition model to collect all relevant primitives as manufacturing operations for assembly sequence planning.

- **The manufacturing model allows for clear identification of manufactured primitives in a product model**

A manufacturing model is only defined for manufactured primitives in a product. Therefore, both functional and manufactured primitives can co-exist in the same product without any confusion about their respective roles.

- **The manufacturing model offers a single source for accessing production related information for each manufactured primitive**

This information may be used for different purposes, such as for user reference to understand how a primitive will be manufactured, or by an analysis tool. Having a single source for this related information ensures that the correct, and most updated information is being accessed. The following software tools and models were successfully tested with MIM in this thesis: CATMAC (cost), MPM (mass), compatibility analysis, RVF (requirements).

- **The definition of a manufacturing model object can be highly automated, or completely manual, based on user requirements**

The information categories can be filled through automated methods that derive information from the design definition model, or they can be user defined. This ensures flexibility for different use-cases. For example, automated methods may be preferred in a DOE or optimisation study, but

manual entries may be preferred when trying to implement a new concept for the first time, or for a single point design.

- **The station and operation graphs from the assembly model together provide information about the steps involved in the complete production of a product in a concise format**

These graphs also have information about what subassemblies exist at different stages of the assembly process through the node attributes in these graphs.

- **The use of operation sets offer a good balance between the automated definition of operation relations, and user control over the assembly sequence**

The definition of an operation set is independent of the product topology (although, it is specific to a given product model / KBE tool). Therefore, there is no need to redefine them when the product's design definition model is changed, for example, when changing the quantity of a primitive, their location, or removing them altogether from the model etc.

- **The MIM allows for identification of trends, and to rank different manufacturing concepts based on the imposed requirements, which helps in making decisions**

- Results from the DOE provided data in a format that allowed quick comparisons, and identification of trends.
- This allowed for the identification of some designs were unapparent, or better (when compared to a baseline). This shows the methodology's potential to find anomalies in the design space, and to allow a further investigation of such designs by users.
- It is important to look at such unusual design points, as they might be a better fit in some cases. For example, by knowing that the mass penalty of using metal ribs is not that high, engineers can decide to include them in the product, if the overall requirement on the total mass is still satisfied. This is because there can be some advantages associated with parts made from metals, over composite materials. For instance, metal parts usually require less specialised equipment and processes, and therefore may be easier to outsource to external companies.

- **The methodology shows the importance of including production considerations on gauging product viability**

A large quantitative effect of joints and part integration could be seen on the total cost and mass, whereas, assembly sequence planning and compatibility analysis offer a reduction of manual iterative steps in the design process. Without accounting for these production considerations to a high level of detail, engineers cannot rely on the *estimated* parameters of interest for making decisions. Accounting for these considerations with high level factors, such as a percentage of the total cost/mass, does not offer the level of detail required to make decisions confidently because of the large uncertainty associated with these values.

Several areas for improvement of the capabilities developed in this thesis can be identified. These are listed below as recommendations for future work:

- **Improve readability of the assembly operation graph**

The operation graph for the overall assembly process is difficult to interpret because of its large size, and the presence of a lot of relations. A possible way to improve its readability is by using specialised graph visualisation software, such as Gephi¹, that offer a variety of features to ease visualisation of large, complex graphs.

- **Account for the effect of assembly sequence on other parameters in the product model**

- The assembly model rightly calculates high production rates when operations are executed in parallel, but the effect of such parallel execution of operations is not accounted for at other places in the product system. For example, with more parallel operations, the fixed cost associated with equipment, and factory floor area would also increase, but the cost tool considered in this thesis can only account for the recurring cost.

¹<https://gephi.org/> (accessed 22 August 2022)

- Additionally, how many operations can be executed in parallel also depends on the number of available workers in the company. Currently, the total number of required workers is not calculated from the assembly sequence, and therefore this also cannot be verified. However, all the information to perform such a calculation is available in the assembly and manufacturing models, and therefore can be implemented in the future.

- **Fully automate the identification of feasible assembly sequences**

- Currently, limited checks that ensure assembly sequence feasibility are implemented, and therefore some user intervention is required as a last step to verify feasibility of the generated assembly sequence. A possible solution for this can be through user-configurable KBE rules, that become a part of the assembly model input configuration. These rules would essentially provide information to perform additional checks on the assembly sequence. They would have to be user-configurable, as they would depend on the product system in consideration. For example, one such criteria that was discussed in the industrial use-case was the inability to remove induction weld tooling if the wing “box” is closed. Therefore, these configurable rules should allow for the definition of a rule that checks for this condition if induction welded joints are used, but the condition is ignored with other joining methods, such as mechanical fasteners.
- Additionally, it can be beneficial to include the infeasible design identified by these configurable rules in the final results, as it is possible that engineers might come up with new ways to make the infeasible assembly sequences “feasible”, with some new techniques or technologies. This can be useful, for example, if an assembly sequence is infeasible, but has a very desirable production rate.

- **Study the applicability of the methodology to more complex products**

Currently, the methodology has only been applied to a single use-case. Its application to different products, possibly with more primitives, and a wider range of design variables, will be helpful in analysing its applicability to such cases. Even though the new methodology automates a lot of aspects of the state-of-the-art design workflow, there are still some things that have to be manually set up or configured by the user. By applying it to more complex cases, its ability to handle such cases can be analysed, which will also possibly lead to identification of more areas for improvement.

- **Use MIM in MDAO workflows**

Currently, the MIM has only been implemented in a sequential design process, where choices for different disciplines, such as design, manufacturing and assembly are fixed at different steps in the process, before moving on to the next. This has the limitation of possibly missing out on good designs because of the choices made at the previous steps. A solution to overcome this is to use MIM in MDAO workflows, where different disciplines in the system are considered together to optimise the whole system. However, this might require development of some new techniques to account for major changes in the optimisation problem due to choices made in MIM. For example, part integration can lead to an entirely different set of primitives in the product.

- **Improve capabilities of CATMAC**

- A limitation of this cost analysis tool is that it does not allow for dynamic definition of manufacturing environments as a part of its input. A manufacturing environment in CATMAC is equivalent to a manufacturing site in the manufacturing model. With its current capabilities, the same manufacturing environment is applied to all primitives in a product, and it cannot be passed as an input for the cost tool *dynamically* for each primitive in the product. Therefore, manufacturing / assembly at different locations cannot be accounted for.
- Additionally, the batch size is defined in the manufacturing environment of CATMAC. Therefore, it is also not possible to consider different batch sizes for different primitives. In this thesis, the batch size was set to a constant value of 1 for all manufactured primitives. After dynamic definition of either the manufacturing environment or batch size is allowed, further work would be required to properly include its effect in the assembly model.
- Non-recurring cost should be included. The manufacturing model DOE results indicate that designs with integrated skin stringers made using the TAPAS-layup method, or designs with

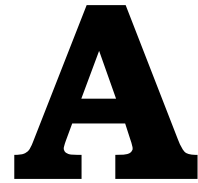
induction welded joints are the best in terms of both cost and mass. But in reality these manufacturing methods require a lot of specialised tooling compared to the hand-layup method or mechanical joints. The cost of this tooling has been neglected in the current analysis, as CATMAC can only considers the recurring cost.

- **Use RVF to setup design workflows from the requirements**

MIM was successfully integrated with RVF in this thesis for the verification of requirements, that also included test cases from the included production considerations. But the full capabilities RVF were not exploited, as it can also be used to create optimisation and DOE workflows automatically from the imposed requirements. Therefore, this is recommended as future work.

- **Improve compatibility analysis**

Only a few compatibility categories and criteria were implemented within the scope of this thesis as a proof of concept, but many more can be possibly included in the future. The current implementation of this analysis has all the necessary base classes for such an extension.



Glossary

Note: All definitions without a reference are formulated by the author.

1. **Production consideration**
A factor from the perspective of production that has an influence on the system design.
2. **Model-Based Systems Engineering (MBSE)**
...the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [18].
3. **Product primitive**
A parametric building block to define the product [26].
4. **Functional primitive**
A constituent of a product, identified based on its specific functions in the product.
5. **Manufactured primitive**
A constituent of a product, identified based on what object is manufactured.
6. **Manufacturing model**
A disciplinary view of a manufactured primitive in a product that complements its design specification to capture production related aspects of the primitive.
7. **Design definition model**
A product model with primitives arranged per a defined topology. It defines what primitives are present in the product, and their specifications (e.g. dimensions, location, quantity, type etc).
8. **Equipment set**
A set of equipment required for the completion of all steps of a manufacturing method.
9. **Material family**
A set of materials grouped together for convenient reference.
10. **Operation**
Execution of manufacturing process(es) that result in the materialisation of a manufactured primitive.
11. **Assembly station**
A physical location where a set of manufacturing operations take place, whose result (part/sub-assemblies) is either passed on to another assembly station, or is the final product itself.

12. Operation graph

A directed acyclic graph with nodes as manufacturing operations and their results, and edges representing the sequence order from one operation to the next.

13. Station graph

A directed acyclic graph with nodes as assembly stations, and edges representing the flow of subassemblies from one station to the next.

14. Operation set

A set of operations with defined interrelations (edges), with an overall input and output node.

B

ArchiMate Modelling Language Reference

A legend describing the ArchiMate modelling language¹ notation used in this report is shown in [Figure B.1](#)

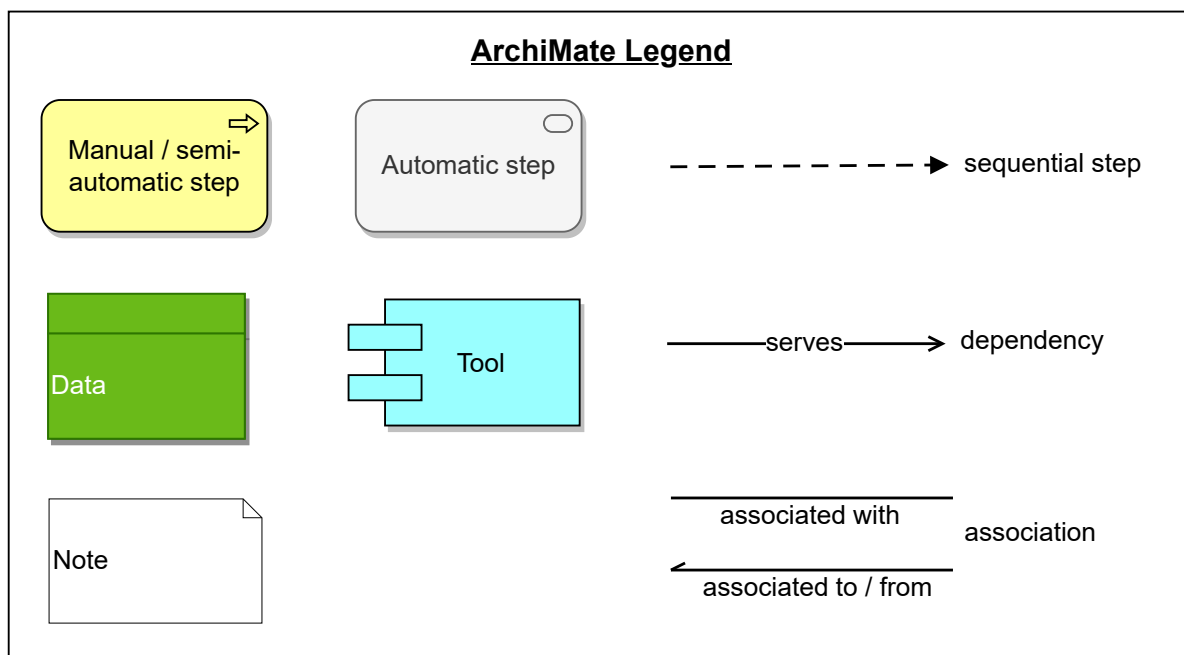


Figure B.1: Legend describing the ArchiMate modelling language notation used in this report.

¹<https://pubs.opengroup.org/architecture/archimate3-doc/toc.html> (accessed 24 September 2022)

C

Examples of User Inputs and Data Specifications in MIM

C.1. User input to define manufacturing model for a manufactured primitive

```
manufacturing_model_input = {  
  
  "uid": "mfg_model_spec_example",  
  
  "info_category_order": ["mfg_method",  
                           "material",  
                           "equipment_set",  
                           "mfg_site"  
                           ],  
  
  "mfg_method": {"info_source": "from_mfg_model_spec",  
                  "library_uid": "mfg_method",  
                  "uid": "hand_layup",  
                  },  
  
  "material": {"info_source": "from_definition_model"  
              },  
  
  "mfg_site": {"info_source": "from_mfg_model_spec",  
               "library_uid": "mfg_site",  
               "uid": "NL1_PPD",  
               },  
  
  "equipment_set": {"info_source": "from_mfg_model_spec",  
                    "library_uid": "equipment",  
                    "uid": "hand_layup equip_set_1",  
                    },  
  
  "manufacturing_time": {"method": "from_catmac"  
                        }  
  
}
```

C.2. MIM database file content examples

C.2.1. Material data file

```
{
  "material_mappings": {

    "disciplines": {
      "open_source_cost": [
        {
          "cost_model_material": "AL_2024",
          "material_uid_list": ["2024_T42", "2024_T46"]
        },
        {
          "cost_model_material": "AL_7075",
          "material_uid_list": ["7075_T7351", "7075_T73"]
        }
      ]
    },

    "material_families": [
      {
        "material_family_uid": "aluminium_7000_series",
        "material_selection": [{"library_uid": "metals", "uid_list": ["7075_T6",
          "7075_T65"]}]}],
      {
        "material_family_uid": "carbon_pps",
        "material_selection": [{"library_uid": "carbon_pps", "uid_list": ["CPPS_1",
          "CPPS_2"]}]}]
    },

    "material_material_compatibility": [
      {
        "criteria_uid": "galvanic_corrosion",
        "criteria_config": {
          "method": "by_mfg_method_uid",
          "method_config": {"mfg_method_uid": "mechanical_joint"},
          "interface_types": ["part_joint_interface"]
        },
        "compatibility_type": "incompatible",
        "valid_material_pairs": [
          {
            "group_1_materials": [
              {
                "method": "by_material_family_uid",
                "uid_list": ["aluminium_2000_series", "aluminium_7000_series"]
              }
            ],
            "group_2_materials": [
              {
                "method": "by_material_family_uid",
                "uid_list": ["carbon_pekk", "carbon_pps"]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    }
  ]
}
]
}

```

C.2.2. Manufacturing method data entry

```

{
  "uid": "machining",
  "type": "part",
  "method_specifications": {
    "cutter_radius": [6, 8],
    "tightest_tolerance": 1e-2
  },
  "valid_material": [
    {"method": "by_library_uid", "uid_list": ["metals"]}
  ],
  "mappings": {
    "open_source_cost": "Machining",
    "project_based_cost": {
      "name": "Machining",
      "process_code": "M"
    }
  }
},

```

C.2.3. Equipment set data file

```

{
  "equipment": [
    {
      "uid": "milling_centre_1",
      "equipment_specifications": {
        "allowable_part_dimensions": {
          "length": {"min": 0, "max": 8000},
          "width": {"min": 0, "max": 6000},
          "height": {"min": 0, "max": 1000}
        },
        "number_of_workers": 1,
        "supported_dof": "5",
        "tightest_tolerance": 1e-3,
        "valid_cutter_radius": [4, 6, 8, 10, 12],
        "valid_material": [
          {"method": "by_library_uid", "uid_list": ["metals"]}
        ]
      },
      "scrap_rate": 0.01,
      "operating_cost_per_hour": 100,
      "worker_cost_per_hour": 30
    }
  ],
  "equipment_sets": [

```



```

{
  "uid": "milled_machining_part equip_set_1",
  "mfg_method": {"library_uid": "mfg_method", "mfg_method_uid": "machining"},
  "equipment_uids": [
    "milling_centre_1",
    "deburring_tool_manual_edge_1"
  ]
}
]
}

```

C.2.4. Manufacturing site data entry

```

{
  "uid": "NL2_HGV",
  "location": "Hoogeveen",
  "equipment_config": [{"library_uid": "equipment", "equipment_list": []}],
  "equipment_set_config": [
    {
      "library_uid": "equipment",
      "equipment_set_list": [
        {"uid": "hand_layup_equipment_set_1", "quantity": 2},
        {"uid": "hi_lite_joint_manual_equipment_set_1", "quantity": 2},
        {"uid": "hi_lite_joint_automated_equipment_set_1", "quantity": 2},
        {"uid": "tapas_layup equip_set_1", "quantity": 1}
      ]
    }
  ],
  "worker_cost_per_hour": 30,
  "mappings": {
    "open_source_cost": "Netherlands_HLU_Batch1"
  }
}

```

C.3. User input for operation set definition

```

operation_set_1 = {

  "uid": "operation_set_1",

  "operation_selection": [
    {"type": "joint",
     "primitive_name": "rib_skin_joints",
     "config": {"method": "all"}
    }
  ],

  "operation_relations": {"method": "by_parallel_Sequences",
                         "config": {"number_of_sequences": 3,
                                    "sequence_order": "root_to_tip"
                                   }
                         },

  "input_node_uid": "operation_set_1_input",

```

```
    "output_node_uid": "operation_set_1_output"  
}
```

C.4. User input for assembly station definition

```
station_rib_installation = {  
  
    "uid": "station_rib_installation",  
  
    "operation_assignment": [  
        {"method": "by_operation_set_uid_list",  
         "uid_list": ["operation_set_1"]  
        },  
  
        {"method": "by_operation_uid",  
         "config_list": [  
             {"uid": "operation_rib_skin_joint_21",  
              "input_node_uid": "output_oper_set_stringers_installation_uid",  
              "output_node_uid": "product_end_uid"  
            },  
         ]  
        },  
  
    ],  
  
    "number_of_stations": 1,  
}
```

D

Baseline Concept Data for the Industrial Use-Case

Table D.1: Parameters of interest for the baseline concept finalised at GKN Fokker Aerostructures.

Parameter	Value
Total cost	\$75008
Total mass	208.48 kg ^a

^a Only the total mass of parts was available for the baseline concept, which is equal to 187.63 kg. The total wingbox mass was estimated by considering that the joints account for 10% [1] of the total wing mass.

Table D.2: Requirements imposed on the baseline concept finalised at GKN Fokker Aerostructures.

Parameter	Requirement text
Total cost	The wingbox shall have a total cost of less than \$76000
Total mass	The wingbox shall have a total mass of less than 215 kg

Table D.3: Manufacturing details of product primitives in the baseline concept finalised at GKN Fokker Aerostructures.

Primitive	Manufacturing method	Material
Skin panels	Hand layup	CFRP ^a
Stringers	Hand layup	CFRP
Ribs	Press forming	CFRP
Spars	Hand layup	CFRP
Stringer-skin joints	Mechanical joint	CRES ^b / Ti ^c
Rib-skin joints	Mechanical joint	CRES / Ti
Spar-skin joints	Mechanical joint	CRES / Ti
Rib-spar joints	Mechanical joint	CRES / Ti

^a CFRP: carbon-fibre-reinforced polymer^b CRES: corrosion-resistant steel^c Ti: Titanium**Table D.4:** Reference material thickness values for part primitives for the baseline concept at GKN Fokker Aerostructures. Note that the average thickness is shown due to the presence of material zones of varying thicknesses on each part.

Primitive	Material	Average thickness [mm]
Skin panels	CFRP ^a	3.35
Stringers	CFRP	2.48
Ribs	CFRP	2.48
Ribs	Aluminium 7075	1.50
Spars	CFRP	1.84
Spars	Aluminium 7075	2.10

^a CFRP: carbon-fibre-reinforced polymer

Bibliography

- [1] E. Torenbeek, *Advanced aircraft design: conceptual design, analysis and optimization of subsonic civil airplanes* (John Wiley and Sons, Inc, 2013).
- [2] A. R. Kulkarni, G. La Rocca, T. van den Berg, and R. van Dijk, *A knowledge based engineering tool to support front-loading and multidisciplinary design optimization of the fin-rudder interface*, in *Aerospace Europe 6th CEAS Conference* (2017).
- [3] A.-L. Bruggeman, B. van Manen, T. van der Laan, T. van den Berg, and G. La Rocca, *An MBSE-Based Requirement Verification Framework to support the MDAO process*, in *AIAA AVIATION 2022 Forum* (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2022).
- [4] H. Bao and J. Samareh, *Affordable design: A methodology to implement process-based manufacturing cost models into the traditional performance-focused multidisciplinary design optimization*, in *8th Symposium on Multidisciplinary Analysis and Optimization* (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2000).
- [5] K. Gantois and A. Morris, *The multi-disciplinary design of a large-scale civil aircraft wing taking account of manufacturing costs*, *Structural and Multidisciplinary Optimization* **28**, 31 (2004).
- [6] S. Castagne, R. Curran, A. Rothwell, and A. Murphy, *Development of a Precise Manufacturing Cost Model for the Optimisation of Aircraft Structures*, in *AIAA 5th ATIO and 16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, Vol. 3 (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2005) pp. 1327–1336.
- [7] M. Rais-Rohani, *A Framework for Preliminary Design of Aircraft Structures Based on Process Information Final Report-Part 1*, Tech. Rep. (1998).
- [8] J. Ceisel, P. Witte, T. Carr, Pogaru Satya, and Mavris Dimitri N., *A non-weight based, manufacturing influenced design (mind) methodology for preliminary design*, in *28th Congress of the International Council of the Aeronautical Sciences* (2012).
- [9] M. Hoefer, N. Chen, M. C. Frank, and M. Frank, *Automated Manufacturability Analysis for Conceptual Design in New Product Development*, in *Industrial and Manufacturing Systems Engineering Conference Proceedings and Posters* (2017).
- [10] D. J. L. Siedlak, O. J. Pinon, P. R. Schlais, T. M. Schmidt, and D. N. Mavris, *A digital thread approach to support manufacturing-influenced conceptual aircraft design*, *Research in Engineering Design* **29**, 285 (2018).
- [11] F. Mas, M. Oliva, J. Ríos, A. Gómez, V. Olmos, and J. García, *PLM Based Approach to the Industrialization of Aeronautical Assemblies*, *Procedia Engineering* **132**, 1045 (2015).
- [12] R. Arista, F. Mas, M. Oliva, J. Racero, and D. Morales-Palma, *Framework to support Models for Manufacturing (MfM) methodology*, *IFAC-PapersOnLine* **52**, 1584 (2019).
- [13] X. Zha, H. Du, and J. Qiu, *Knowledge-based approach and system for assembly oriented design, Part I: the approach*, *Engineering Applications of Artificial Intelligence* **14**, 61 (2001).
- [14] R. Viganò and G. Osorio Gómez, *Assembly planning with automated retrieval of assembly sequences from CAD model information*, *Assembly Automation* **32**, 347 (2012).
- [15] D. Jiwang, H. Qichang, and F. Xiumin, *Automating generation of the assembly line models in aircraft manufacturing simulation*, in *2013 IEEE International Symposium on Assembly and Manufacturing (ISAM)* (IEEE, 2013) pp. 155–159.

- [16] A. Desai, *Ease of product assembly through a time-based design methodology*, [Assembly Automation](#) **39**, 881 (2019).
- [17] Z. Chen, J. Bao, X. Zheng, and T. Liu, *Assembly Information Model Based on Knowledge Graph*, [Journal of Shanghai Jiaotong University \(Science\)](#) **25**, 578 (2020).
- [18] INCOSE, [Systems Engineering Vision 2020](#), Tech. Rep. (2007).
- [19] A. Hiken, *The Evolution of the Composite Fuselage: A Manufacturing Perspective*, in [Aerospace Engineering](#) (IntechOpen, 2019).
- [20] J. W. van Ingen, J. E. A. Waleson, A. Offringa, and M. Chapman, *Double curved thermoplastic orthogrid rear fuselage shell*, in *SAMPE Europe Conference 2019 Nantes - France* (2019).
- [21] S. R. Shankar and D. G. Jansson, *A generalized methodology for evaluating manufacturability*, in [Concurrent Engineering](#) (Springer US, Boston, MA, 1993) pp. 248–263.
- [22] M. Rais-Rohani, [NASA Grant NAG-1-1716, NASA Langley](#), Tech. Rep. (NASA, 1996).
- [23] R. Curran, S. Raghunathan, and M. Price, *Review of aerospace engineering cost modelling: The genetic causal approach*, [Progress in Aerospace Sciences](#) **40**, 487 (2004).
- [24] T. van der Laan, A. Johman, T. van Puffelen, S. Nolet, B. van Maanen, E. Daugulis, and T. van den Berg, *An open source part cost estimation tool for MDO purposes*, in [AIAA AVIATION 2021 FORUM](#) (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2021).
- [25] J. P. Risueño and B. Nagel, *Development of a Knowledge-Based Engineering Framework for Modeling Aircraft Production*, in [AIAA Aviation 2019 Forum](#) (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2019).
- [26] T. van den Berg and A. van der Laan, *A multidisciplinary modeling system for structural design applied to aircraft moveables*, in [AIAA AVIATION 2021 FORUM](#) (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2021).
- [27] T.-C. Horng, *A comparative analysis of supply chain management practices by Boeing and Airbus : long-term strategic implications*, [Master thesis](#), Massachusetts Institute of Technology (2007).
- [28] W. Oosterom and R. Vos, *Conceptual Design of a Flying-V Aircraft Family*, in [AIAA AVIATION 2022 Forum](#) (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2022).
- [29] S. Peters, C. Fortin, and G. McSorley, *A Novel Approach to Product Lifecycle Management and Engineering Using Behavioural Models for the Conceptual Design Phase*, in [IFIP Advances in Information and Communication Technology](#), Vol. 565 IFIP (Springer, 2019) pp. 159–169.
- [30] M. R. Bahubalendruni, B. B. Biswal, and G. R. Khanolkar, *A Review on Graphical Assembly Sequence Representation Methods and Their Advancements*, [Journal of Mechatronics and Automation](#) **1**, 16 (2014).
- [31] J. Munroe, K. Wilkins, and M. Gruber, [Technical Report: NASA/CR-2000-209337](#), Tech. Rep. July (2000).
- [32] P. Sanders and C. Schulz, *High Quality Graph Partitioning*, in [Graph Partitioning and Graph Clustering - 10th DIMACS Implementation Challenge Workshop](#), Contemporary Mathematics, Vol. 588 (American Mathematical Society, 2013).
- [33] Y. Ege, M. Azizoglu, and N. E. Ozdemirel, *Assembly line balancing with station paralleling*, [Computers and Industrial Engineering](#) **57**, 1218 (2009).
- [34] G. La Rocca, *Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design*, [Advanced Engineering Informatics](#) **26**, 159 (2012).