UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND THE ENVIRONMENT

Computational Engineering and Design

Whole Life Cost Methods for Aero-Engine Design

by

James Stephen Wong

Thesis for the degree of Doctor of Philosophy

November 2012

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND THE ENVIRONMENT

COMPUTATIONAL ENGINEERING AND DESIGN

Doctor of Philosophy

WHOLE LIFE COST METHODS FOR AERO-ENGINE DESIGN

by James Stephen Wong

This research was motivated by the move of aero-engine manufacturers to provide services as well as products. With leasing arrangements such as TotalCare©, the aero-engine manufacturers are responsible for the operation, management and maintenance of their products while airlines pay a contracted rate for their use. As a result, aero-engine manufacturers need to minimise the cost their products incur over their lifecycle to increase profits. It is widely accepted that the greatest scope to reduce costs is in the design stage. Hence, the aim of this thesis is to create tools and methodologies for designers which will allow them to monitor the impact of design decisions on whole life cost.

Two different approaches to designing for whole life cost were presented. The Life Cycle Cost (LCC) approach and the comparatively novel Value Driven Design (VDD) approach. It was observed from research literature that models for both LCC and VDD need to be tailored to specific objectives in order to keep the scope of the model manageable. This makes generic LCC or VDD models unfeasible and consequently the reuse of these models is limited. With this in mind, a methodology was developed for creating integrated analyses models which were customisable, modular and transparent so as to facilitate future modification and reuse. It used a commercial software integration package called Isight, and modular analyses modules. Model Based Systems Engineering (MBSE) was also used in the development of the integrated model architecture. Case studies were performed for both LCC and VDD approaches to allow comparison of their respective merits and flaws. Finally several avenues of future work in VDD and MBSE were discussed.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, James Stephen Wong, declare that the thesis entitled Whole Life Cost Methods for Aero-Engine Design and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
  1. Journals
     - Cheung, J., Scanlan, J.P., and Wong, J.S. (2011) Application of value-driven design to commercial aero-engine systems. *Journal of Aircraft*, 1-35.
  2. Conference
     - Wong J.S., Scanlan, J.P., and Eres, H., "Modelling the Life Cycle Cost of Aero-engine Maintenance", 15th ISPE International Conference in Concurrent Engineering, Belfast, UK, 18-22 August 2008.

- Wong J.S., Scanlan, J.P., and Eres, H., “An integrated life cycle cost tool for aero-engines”, 6th International Product Lifecycle Management Conference, Bath, UK, 06 - 08 Jul 2009.
- Wong J.S., Scanlan, J.P., and Eres, H., “A Systems Engineering Approach to Aero-Engine Life Cycle Costing”, 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, USA, 04-07 Jan 2010.
- Cheung, J., Scanlan, J.P., Wong, J.S, Forrester, J., and Eres, H., “Application of Value-Driven Design to Commercial Aero-Engine Systems”, 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth, Texas, USA, 13 - 15 September 2010.

3. Internal Report
    - Collopy, P., Forrester, J., Wiseall, J., Hollingsworth, P., Scanlan, J.P., Eres, H., Cheung J., Wong, J., and Hoptroff, J., “Advanced Cost Modelling Methodologies: Value Driven Design – Scoping and Feasibility Study”, Technical Report, DNS153982, Rolls-Royce plc.

Signed: …………………………………………………………………….

Date:…………………………………………………………………………

Acknowledgments

First and foremost, I would like to thank my supervisors, Prof. J. P. Scanlan and Dr. H. Eres for their guidance and support. You two have shown me patience and encouragement without which this work would have been possible. I am forever grateful.

I would also like to thank Julie Cheung and Steve Wiseall for their invaluable feedback, discourse, and the many cups of coffee. Special thanks also go to Paul Collopy and Peter Hollingsworth for sharing their ideas about Value Driven Design.

Finally, I dedicate this work to my loving wife, my parents, my two sisters, and especially my dearest grandmother who I miss so much.

Nomenclature

ABC	Activity Based Costing
AI	Artificial Intelligence
AML	Adaptive Modeling Language
Bi	Biot Number
BPML	Business Process Modelling Language
BPR	Bypass Ratio
BSI	British Standards Institution
C_d	Coefficient of drag
CAD	Computer Aided Design
CAx	Computer Aided technologies
CBR	Case Based Reasoning
CBS	Cost Breakdown Structure
CE	Concurrent Engineering
CER	Cost Estimating Relationship
CFD	Computational Fluid Dynamics
CFF	Cooling Flow Fraction
D_n	Nacelle drag
DATUM	Design Analysis Tool for Unit-Cost Modelling
DEE	Design and Engineering Engine
DFC	Design for Cost
DOC	Direct Operating Cost
DOD	Department of Defense
DSL	Domain Specific Language
DTC	Design to Cost
ESA	European Space Agency
FAA	Federal Aviation Administration

FE	Finite Element
FLOPS	Flight Optimization System
FW	Fuel Weight
GDL	General-purpose Declarative Language
GE	General Electric
GUI	Graphical User Interface
HCF	High Cycle Fatigue
HP	High Pressure
HSCT	High Speed Civil Transport
ICARE	Illustration, Constraint, Activity, Rule, and Entity
ICES	Implied Cost Evaluation System
IEC	International Electro-technical Commission
INCOSE	International Council on Systems Engineering
IP	Intermediate Pressure
IPAS	Integrated Product and Services
IPT	Integrated Product Team
ISO	International Organization for Standardization
K_{cool}	Cooling flow factor
KBE	Knowledge Based Engineering
KNOMAD	Knowledge Nurture for Optimal Multidisciplinary Analysis and Design
LCC	Life Cycle Cost
MBSE	Model-Based Systems Engineering
MFOP	Maintenance Free Operating Period
MLW	Maximum Landing Weight
MML	MOKA Modelling Language
MOKA	Methodology and software tools Oriented to Knowledge-Based Engineering Applications
MPLW	Maximum Payload Weight
MTBR	Mean Time Between Removal

MTOW	Maximum Take-Off Weight
MZFW	Maximum Zero Fuel Weight
NASA	National Aeronautics and Space Administration
OEM	Original Equipment Manufacturer
OEW	Operational Empty Weight
OMG	Object Management Group
OOSEM	Object-oriented Systems Engineering Method
OPM	Object-Process Methodology
PLM	Product Lifecycle Management
Pr	Probability
PW	Pratt & Whitney
R	Reliability
RDTE	Research, Development, Test and Evaluation
RR	Rolls-Royce
RSM	Response Surface Model
RUP SE	Rational Unified Process for Systems Engineering
S	Wetted area of nacelle
SA	State Analysis
SAE	Society of Automotive Engineers
SE	Systems Engineering
SFC	Specific Fuel Consumption
SysML	Systems Modeling Language
t	Time
T	Temperature
TET	Turbine Entry Temperature
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language
V	Velocity

VDD	Value Drive Design

List of Symbols

λ	Weibull scale parameter
β	Weibull shape parameter
θ	Weibull location parameter
η_{int}	Internal cooling efficiency
η_{poly}	Polytropic efficiency
η_{th}	Thermal efficiency
ψ	Cooling flow fraction
ξ	Compression temperature ratio
ζ	Expansion temperature ratio
ε_0	Blade cooling effectiveness
ε_f	Film cooling effectiveness
ρ	Air density

1. Introduction

In the last ten years, the airline industry has come under pressure, due to falling revenues, to reduce operating costs [1]. This pressure has extended to aero-engine suppliers as well. Programs such as TotalCare® [2], offered by Rolls-Royce (RR), deviate radically from the traditional procurement method in which the airlines/operators own and maintain the aero-engines (Figure 1-1) of their fleet. When airlines/operators enter into a TotalCare® arrangement they pay the Original Equipment Manufacturer (OEM) a fixed rate for the missions being flown. Under this contract the OEM takes responsibility for maintenance and support services. Since both the OEM and the airliners/operators are contractually obliged to ensure that the engines are kept flying with minimal disruption and cost, the conflict of interest that arguably existed before is eliminated. This kind of arrangement has many implications for the OEM. The most significant of which is that the OEM now has to take financial responsibility for the operating performance and costs of its product. Any disruptive events which take an engine out of operation will cause the engine to cease generating income for the OEM. Additionally, the engine will become a cost drain as it incurs maintenance costs. It is thus incumbent on the OEM to undertake a deep analysis of risks and costs.

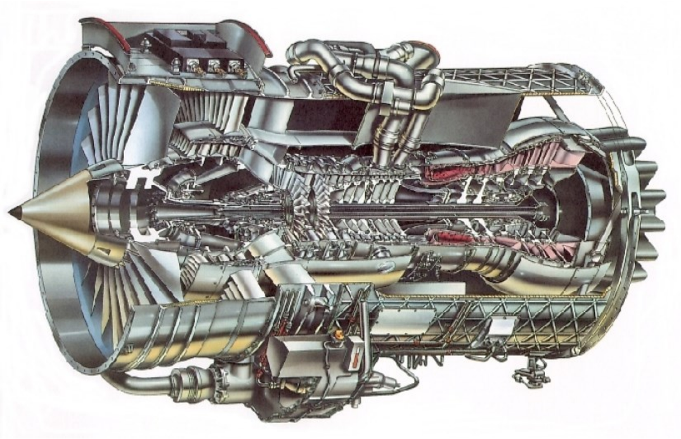


Figure 1-1: The BR715 engine [3].

In a scenario such as the one described above, where significant operating costs exist, a Life Cycle Cost (LCC) analysis can be used to determine the long term cost of ownership of a product or system. Life Cycle Costing takes into consideration all the costs that a product/system incurs throughout all phases of its life; from conception all the way to disposal. It provides the platform from which design trade-offs can be made with regards to costs accumulated over the lifetime of the product/system. A familiar conundrum to an aero-engine designer would be whether to choose a design which is cheap to manufacture but expensive to operate or vice versa. By the time full-scale development has been reached, a large proportion of LCC will already have been committed [4-7]. Therefore, it is critical that the necessary tools are made available to the designer to predict the effects of design decisions on LCC in all stages of the design process.

1.1. Research Aim

The main aim of this research is to develop tools and methods which will enable aero-engine designers to predict whole life cost. As this work focuses on producing cost estimates for the purpose of design, it is useful to understand that the design of a product/system involves several product design and development phases. In a Systems Engineering (SE) approach, there are six key lifecycle stages. These are: Concept, Development, Production, Utilisation, Support, and Retirement. RR has its own formal review process called the Derwent Process which is similar to the SE process; Figure 1-2. In this thesis, all mention of design stages will refer to the Derwent Process.

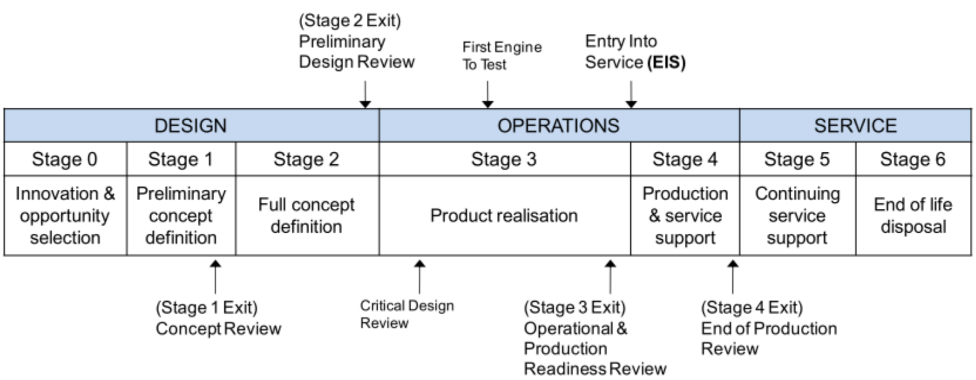


Figure 1-2: Rolls-Royce product design and development phases [8].

Figure 1-3 helps to illustrate the practice of how cost estimates are implemented in design. In the past, the process of developing cost estimates was sequential. The designer would submit a design to the cost estimator and the response time for the cost estimate would be in the scale of weeks. Presently, Concurrent Engineering (CE) practices, such as Integrated Product Teams (IPT), together with various technological advances (internet, email, computer-aided technologies) have improved the collaboration between designer and cost estimators to reduce cost estimation response times to a measure of days. In the future, Tammineni [9] envisions an environment where cost estimators will be responsible for the development of cost estimation tools which designers can use to generate their own quick cost estimates. It is with this philosophy that the objectives in the following section were derived.

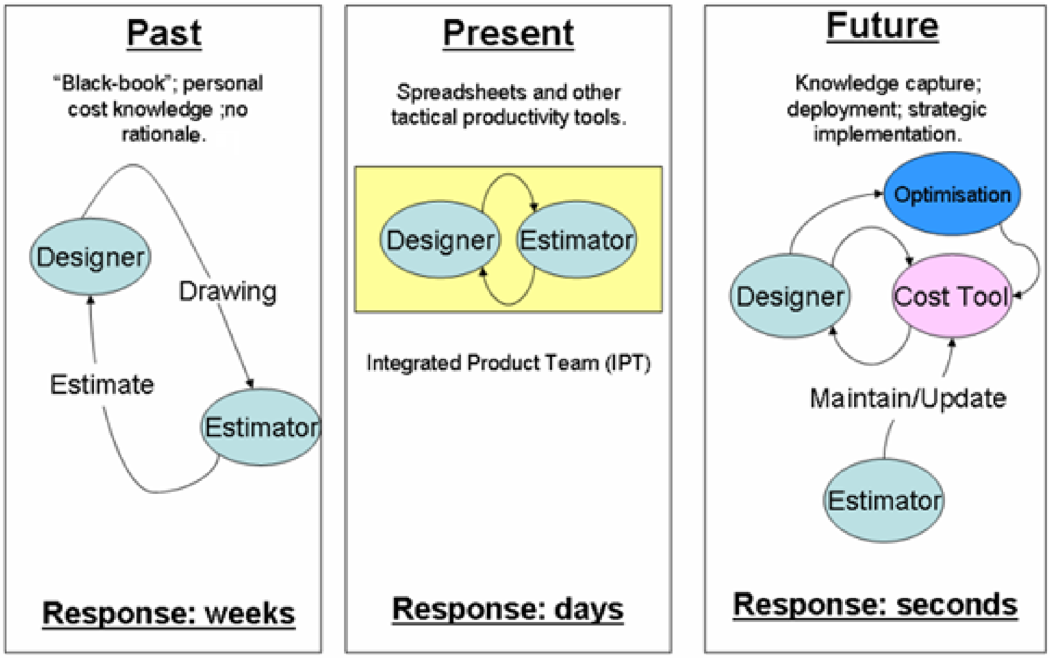


Figure 1-3: Trends of cost estimation response times [9].

1.2. Research Objectives

From the research aim discussed above, several objectives were formulated. These are:

Objective 1: Determine desirable characteristics for aero-engine Life Cycle Cost (LCC) design tools.

The concept of LCC was popularised when it was applied by the US Department of Defense (DOD) in the 1980s [10]. Since then, much research has been devoted to how LCC can be used to improve design decisions. In the case of RR, the practice of leasing contracts, such as TotalCare®, is relatively new. In the last 10 years of RR's civil large engine business, TotalCare® coverage of the installed fleet rose from 3% to over 65% [11]. Additionally, more than 90% of civil large engine orders from 2009 to 2012 included TotalCare®. When aero-engines were sold to airlines, it was common for engine manufacturers to sell at a loss and to generate their profits from the sale of spare parts. As the business model adopted by RR has changed it thus follows that the design approach taken must change as well. Existing LCC models were reviewed to determine what characteristics are desirable for aero-engine design.

Objective 2: Develop a methodology to link aero-engine design to LCC

A methodology was developed, incorporating the desired characteristics identified in Objective 1, to link aero-engine design parameters to LCC. The developed model focussed on one type of component, a turbine blade, but has the facility to extend the analyses to include other components as well.

Objective 3: Incorporate Model-Based Systems Engineering (MBSE) into the LCC process

Model-Based Systems Engineering (MBSE) is an approach which uses modelling in the specification, design, integration, validation, and operation of a system. Among its claimed benefits [12] include improvements in management of complexity, integration of multiple modelling domains across system life cycle, data capture and communication among system/project stakeholders.

As LCC is a system attribute, it is an emergent property that is influenced by the interactions between the different aspects of the system. As a result, developing a model to link the design parameters of interest to LCC requires the involvement of the relevant stakeholders (such as cost, geometry definition, stress, thermal, and manufacture). MBSE provides the framework to facilitate this collaboration. In future, MBSE also holds the potential to automate the construction of integrated analyses models via MBSE constructs to reduce the model development time [13]. A case study was performed to illustrate how MBSE can be used in the LCC process.

Objective 4: Compare Value Driven Design (VDD) approach with LCC approach

Value Driven Design (VDD) is a relatively new approach to design [14] which uses a mathematical value model and requirements flexibility to achieve an optimal solution with respect to performance, cost, schedule, and other measures important to the stakeholders. VDD does not set firm requirements on extensible system attributes (such as cost, life, and weight). This contrasts with a design for LCC approach which follows the traditional requirements based approach and only measures cost. As a VDD approach incorporates cost in its analysis, it is useful to compare its merits and flaws with the LCC approach. Additionally, the methodologies developed for the LCC approach can be applied to a VDD analyses as they both require integrated analyses models to model the complex relationships between design inputs and the respective outputs. A case study parallel to that of the LCC case study was performed to determine the differences between designing for optimum LCC and system value.

1.3. Thesis Layout

The remainder of this thesis is organised in the order of the research objectives laid out above.

Chapter 2 reviewed the relevant literature associated with cost modelling. It then explained the concept of LCC and described the process of a LCC analysis. A review of LCC models was performed to determine desired characteristics for an aero-engine LCC tool. A discussion around the issues regarding LCC modelling was presented. The concept of MBSE was also introduced and the INCOSE Object-oriented Systems Engineering Method (OOSEM) was summarised. Finally important future developments in MBSE were identified.

Chapter 3 detailed the methodology for developing a LCC model which incorporated the desired characteristics identified in the literature review. The integrated model was modular, extendable and transparent. The model was used to study the effects of turbine blade design on LCC. It linked manufacturing cost, deterioration mechanism life prediction and maintenance cost.

Chapter 4 used a case study to illustrate how MBSE could be used to support a LCC analysis. The study examined how LCC changed with turbine entry temperature (TET) and cooling flow fraction (CFF). In addition the study also demonstrated how the model developed in chapter 3 was adapted to address different study objectives.

Chapter 5 introduced the concept of VDD and provided an overview of the process. An overview of VDD models was presented and comparisons were drawn with the LCC methodology. Finally a case study, parallel to the one performed in chapter 4, was carried out to compare the LCC and VDD approaches.

Chapter 6 summarised the contributions and major conclusions of this research. Avenues for further research were also identified.

2. Literature Review

Cost is one of the important elements in design decision making in the field of engineering [8]; as illustrated in Figure 2-1. Modelling costs allows the cost impact of design decisions to be measured [15-17]. This chapter serves to introduce cost concepts to the reader as well as highlighting the importance of cost information in design. The most common costing estimation methodologies are presented and discussed. Life Cycle Cost is central to this thesis as it is a measure that can be used to show how design decisions impacts the long term cost of a product or service. The concept and process of LCC is introduced and the state of the art of LCC in aerospace is reviewed.

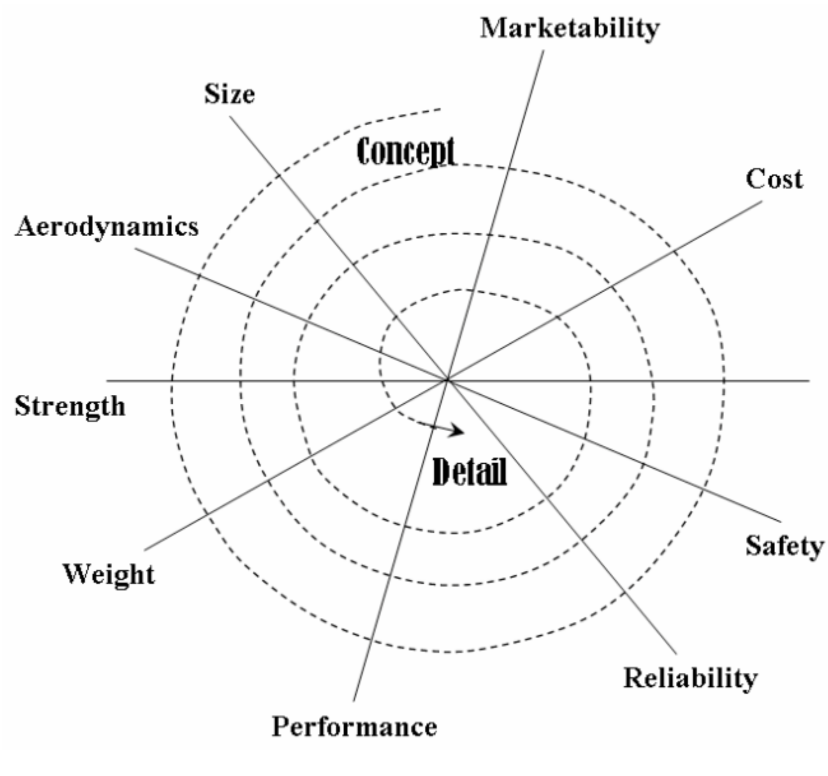


Figure 2-1: The design spiral [8].

LCC modelling issues are identified from the literature and these issues provide the impetus for the work presented in this thesis. These issues are dependent on the how complex the relationships are between the design variables and cost elements.

The LCC models in the reviewed literature used different kinds of models of varied disciplinary fields. It becomes clear that designing for LCC of an aero-engine requires a framework which must support collaboration and interoperability at several levels: across global organisations, between disciplines involved in the systems development effort, among design teams within a given discipline, between design and analysis efforts, and between development and manufacturing. The framework must be capable of managing and integrating this collection of models. Model Based Systems Engineering (MBSE) has been championed as an approach to support this. An overview of MBSE and a discussion of how it can be used to resolve LCC modelling issues are presented.

2.1. Cost in the design stage

An oft quoted maxim in cost modelling literature is that 70-80% of avoidable cost is controllable in the concept design stage and that the opportunities to reduce costs diminish as the product/system progresses through its life cycle; this is illustrated in Figure 2-2. The authors of these papers generally use this statistic as the motivation for developing methodologies which will be able to generate more accurate cost estimates in the concept design phase.

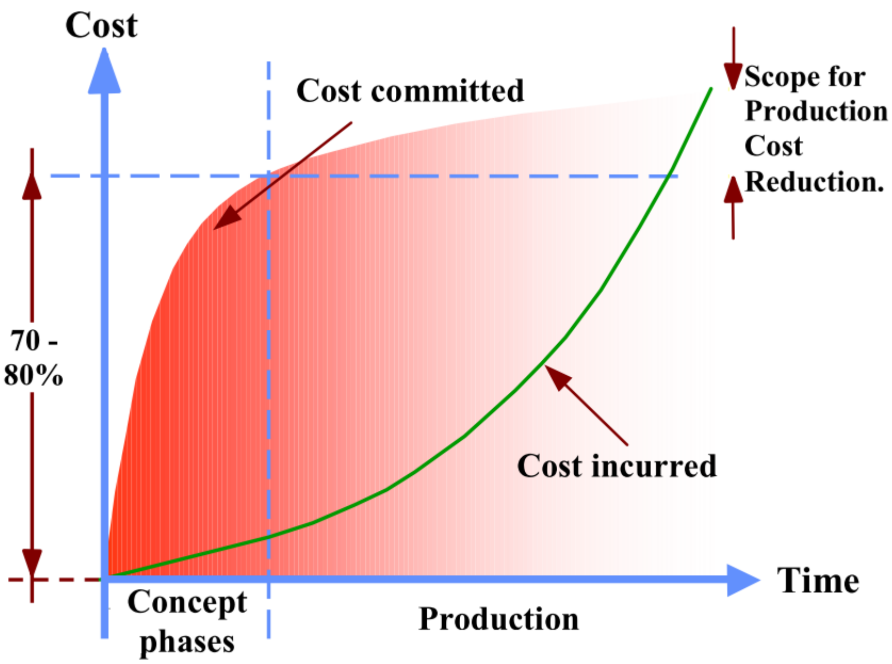


Figure 2-2: Cost commitment curve [18].

There are several authors who dispute the validity of this statement however. Ulrich et al. [19] found logical inconsistencies in the derivation of the cost percentages involved. Barton et al. [6] assert that there have been little or no basis for the cost percentages mentioned above. These authors argue that opportunities for cost saving are present in all stages of a product lifecycle. Despite these debates to the level of costs involved, there is little opposition to the belief that the design stage offers the most opportunities for cost reductions.

Where controlling cost in the design stage is concerned, there are two established methods: Design To Cost (DTC) [20] and Design For Cost (DFC) [21]. In the case of DTC, cost is controlled by establishing cost goals, and designs are adjusted iteratively until they satisfy the cost requirement. It is seen to be a management driven process since cost targets are set by derived requirements. The DTC problem can be represented mathematically as:

$$g(x) \geq r, \tag{2.1}$$

where x is a vector of cost drivers, $g(x)$ is the vector of equations which will allow the cost drivers to be compared to requirement goal values, and r is the vector of requirement goal values.

There are some disadvantages to DTC. If the design cannot meet the set requirements, the requirements have to be adjusted for the design to be feasible. As a result, DTC usually leads to a reduction of performance until the budget is met [20]. Additionally, DTC is highly dependent on how sensible the cost targets are. A poorly estimated cost target will lead to an unfeasible design.

DFC on the other hand is a philosophy which advocates a conscious effort to keep costs to a minimum. DFC aims to obtain the lowest possible cost for the given set of requirements. Where DTC is iterative in nature, DFC attempts to engineer low cost into the design from the start. DFC, unlike DTC, is hence seen to be engineering driven. The DFC problem can be expressed as:

$$\begin{gathered} minimise\ f(x) \\ subject\ to\ g(x) \geq r \end{gathered} \tag{2.2}$$

where x is a vector of design variables which will affect requirements, $f(x)$ is the cost equation, $g(x)$ is the vector of equations which will allow the cost drivers to be compared to requirement goal values, and r is the vector of requirement goal values.

Asiedu and Gu [4] advocate a more holistic approach using concurrent engineering and LCC analysis to provide cost information to designers. This approach corresponds well to a service provision business model such as TotalCare®. Curran et al [22] notes that both DFC and DTC tools are too specific and that an approach which can encompass various levels of cost evaluation for different purposes is required. They conclude that it is more important to give designers supportive costing tools which can help the production definition process by linking design decisions to estimated cost impact.

2.2. Cost Definitions

This section briefly describes some of the terms often used in cost engineering and are well documented in the associated literature [23, 24].

2.2.1. Non-recurring and recurring costs

Non-recurring costs, as its name suggests, refers to one-time costs incurred at some point in the lifecycle of a product/system. These costs are typically for development or investment and include costs for research, design, tooling acquisition/upgrade, and certification. Recurring costs are costs which are repeated throughout the lifecycle of the system/product. These costs usually relate to those which are required for production, consumables, labour and maintenance. These costs are included when modelling learning and improvement curves. Recurring costs, in particular, should be seen to reduce over time; a concept introduced by Wright [25].

2.2.2. Fixed and variable costs

Fixed and variable costs are usually used together with break-even analyses in high-level financial studies to make investment decisions. A sign of good economic health is when profit levels are higher than fixed costs [22]. Fixed costs are defined by Schiller to be the "costs of production that do not change when the rate of output is altered" [26]. These costs include general production costs sustained in keeping the company operational. In contrast, variable costs change with the rate of output. These costs include elements such as labour, material, and machining.

2.2.3. Direct and indirect costs

Direct costs are costs which can be classified into specific categories or causes. There are thus more easily identified and can be associated with a product, service, programme, function, or project [22]. Indirect costs are the opposite of this in that they are difficult to identify or to be assigned to a specific operation or category. These costs are hence often labelled as overheads. Examples of indirect costs include cleaning, building works, and misappropriation.

2.3. Cost Estimation

Cost estimation can be defined as the process of predicting or forecasting the cost of a work activity or output [27]. There are several well recognised cost estimation techniques mentioned by several authors who have reviewed and classified these techniques. Roy et al. [28] classifies these methods into qualitative and quantitative methods. Curran et al. [22] separates the methods into classic and advanced estimation techniques. Scanlan et al. [29] organises these methods as parametric or generative. Niazi et al. [30] provides a detailed classification which covers most of the cost estimation techniques; this is show in Figure 2-3.

The following sections give an overview of the more popular cost estimation techniques employed in aerospace engineering.

2.3.1. Analogous

The analogous estimation method is one of the most established and widely applied methods of costing; especially within the aerospace industry [22]. It takes a similar product or system to the one being developed and adjusts its cost based on the differences between the two [31]. The effectiveness of this method depends on the ability to correlate the appropriate cases and identify their differences [32]. A major disadvantage of this method is the high degree of judgment required [4]. Personnel with the expert knowledge and the required familiarity of the product and processes may not always be available.

Analogous costing is used in case based reasoning (CBR) tools. It typically has a database of past projects and formalises the captured knowledge into similarity functions and analogy rules [33, 34]. However, these kinds of tools can be complex and subjective. Therefore, its development and use is difficult and is dependent on the capability of the user.

The Federal Aviation Administration (FAA) Life Cycle Cost Estimating Handbook [35] recommends the use of this method for a new product or a system that is primarily a combination of existing sub-systems, equipment or components for which recent and complete historical cost data is available.

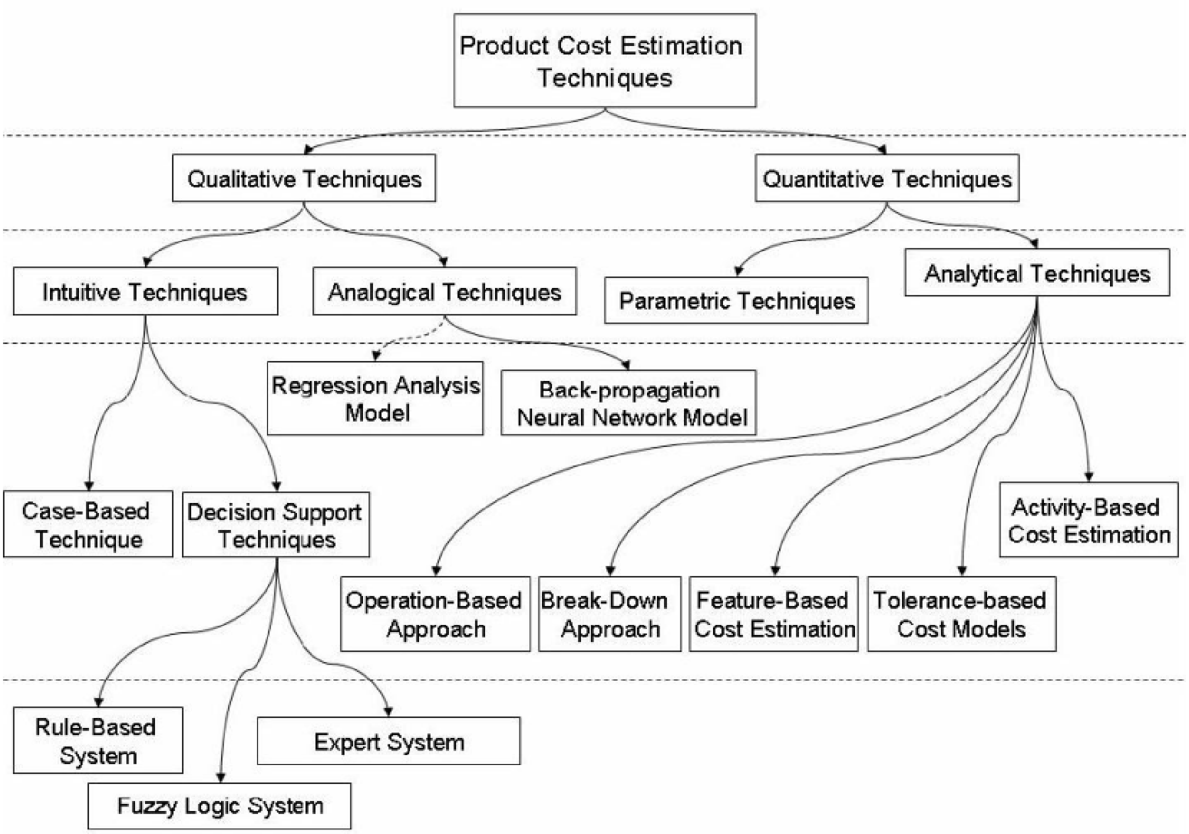


Figure 2-3: Classification of cost estimation techniques [30].

2.3.2. Parametric

The Parametric Cost Estimating Handbook [36] states that "A parametric cost estimate is one that uses cost estimating relationships (CER) and associated mathematical algorithms (or logic) to establish cost estimates." This technique typically uses regression analysis based on historical and technical data to form CERs.

CERs link changes in one or more parameters to a change in cost. These parameters are also known as cost drivers. The learning curve, demonstrated by Wright [25], was an early example of parametric cost estimating. Wright determined that when the total aircraft production doubled, the required labour time decreased between 10% and 15%. A relationship between the unit cost of the aircraft and the number of aircraft produced was hence formed. This CER was used extensively during World War II when there was an exponential increase in the production of military aircraft.

Parametric cost models can be used easily and at an early stage in the design process when there is little knowledge on product definition. However, these models are sensitive to the range of use. This is due to their inability to estimate the differences in the product definition which are not evident in the historical data [22]. Parametric models use regression models which can be linear, quadratic or multi-dimensional [37] depending on the number of cost drivers used. An example of a CER is a relationship that links the cost of a design process to the number of design drawings produced. The number of drawings increases with the complexity of the product/system. This is linked to design effort and time which ultimately translates to design cost. Parametric estimating is also often referred to as a 'top-down' estimating technique [4].

2.3.3. Generative

This method first identifies and sizes the component parts and task. Part and task estimates are then aggregated to form an overall estimate. The application of this method requires detailed design and configuration information for the various system components and accounting information for materials, equipment and labour [38].

The resultant model is very detailed with well understood cause and effects. However, this approach is the most time consuming and costly approach as compared to the above two methods. Substantial detailed expert data is also required.

2.3.4. Activity based

Activity Based Costing (ABC) is a method which assigns the costs of activities to products and/or services according to how much the product/service has consumed

[39, 40]. This is done so that the products/services which are unprofitable can be identified. ABC aims to objectively identify the cause and effect relationships so that costs can be assigned. When the costs of the activities have been identified, the cost of each activity is attributed to each product to the extent that the product uses the activity. As a result, ABC is able to identify areas of high overhead costs per unit where the appropriate action can be taken.

2.3.5. Feature based

This method relies on the concept that a design can be decomposed or defined as features, and these features can be associated with cost. There are two reasons why design features are often related to cost [41]. Firstly, cost functions can be derived for classes of similar objects which serve as key drivers of global cost estimation and are linked to the engineering domain. Secondly, the designer desires to know the causes of cost, so that when linked to design features, they are able to influence committed cost directly.

There are three components of cost which relate to design features [41]. These are:

1. Costs assigned directly to individual design features at a feature or assembly level;
2. Costs incurred for a collection of design features: at a component or batch level;
3. Costs assigned to design features at an order or facility level.

Table 1 shows how features can be categorized for the purpose of costing [42]. It also shows the several levels of feature definitions that can be used.

2.3.6. Fuzzy logic

The fuzzy logic method was formulated to address the uncertainty in non-deterministic parameters. Fuzzy logic is a mathematical discipline that was originally created to bridge the gap between the binary world of digital computing and that of continuous intervals displayed in nature [43]. Its major contribution is its capability of representing vague and imprecise knowledge, as applied to classification, modelling and control [44]. The fuzzy approach provides a methodology by which algorithms for the prediction or control of a system are arrived at through qualitative expressions which link linguistic variables [45].

Table 1: Definition of feature drivers [42]

Feature Type	Examples
Geometric	Length, width, depth, perimeter, volume, and area
Attribute	Tolerance, finish, density, mass, material, and composition
Physical	Hole, pocket, skin, core, pc board, cable, spar, and wing
Process	Drill, lay, weld, machine, form, and chemical-mill
Assembly	Interconnect, insert, align, engage, and attach
Activity	Design engineering, structural analysis, and quality assurance

There are three main procedures within fuzzy logic [22]. They are:

1. Recognise one or more assigned physical conditions which require analysis or control;
2. Process these as inputs according to fuzzy 'if-then' rules that are expressed linguistically;
3. Average and weight the outputs from all of the individual rules into a single 'defuzzified' output that results in the decisions and/or actions required of the system.

There are two situations where this method can be employed. First of which is in a very complex model where its understanding is limited or judgmental. The second is for a process where human reasoning, human perception, or human decision-making are inextricably linked. One of the advantages of using such a method is that the mathematical concepts used are kept simple and transparent. Another is that the method has the ability to match any set of input-output behavioural data. A fuzzy logic model can also integrate with traditional techniques and experiences. However, this method of cost modelling is not well established and it is subject to the domain rule of being limited through limited scope and application [22].

2.3.7. Neural network

Neural networks are based on the concept of a system which will learn to predict the effect on cost given a number of product attributes. This method derives its name from the way it arrives at its result. The model, given input data from historical case studies, has a hierarchy of neural like logic gates which run through procedural permutations and combinations. The model then trains itself repeatedly in order to arrive at a logical conclusion. Once trained, the attribute values can be supplied to the network of neurons, in order for it to apply the approximated functional steps in computing an expected resultant cost [22]. This method does not simplify analysis but what it does do is programmatically develop logic and rules.

While neural networks can be used identify obscure relationships in the given data set, it does require a large historic bank of data which is of similar information and form to the new products being analysed. As a result, the accuracy of the result depends very much on the quality, quantity and relevancy of the input learning data. This also means that this method is unsuitable for analysing novel or innovative products where the required input data is not available. Inherently, this method has a "black box" nature and is not suited for the design process which requires the reasons behind CERs to be transparent.

2.4. Accuracy in cost estimation

The importance of an accurate cost estimate cannot be understated. Accurate estimates affect the company both externally and internally. Externally, if a company's estimate is unrealistically low, the company may obtain the order but will suffer financial penalties. Conversely, if a cost estimate is too high, it will make the company's product uncompetitive. Internally, underestimated costs will result in initial plans for staffing, scheduling, machine processing, tooling, and similar operations to be unachievable. Overestimated costs will tend to result in the phenomenon of a "self-fulfilling prophecy," whereby money which is available to be spent is spent [46]. Therefore, a company's survival depends very much on accurate cost predictions. The Freiman curve [46] shown in Figure 2-4 shows a theoretical relationship between cost estimates and the actual induced cost. The shaded area below the curve is the region where cost estimates are the most accurate and also the region of minimum actual cost. Ultimately, the Freiman curve illustrates that accurate cost estimates reduces the costs wasted on addressing the initial inaccuracies.

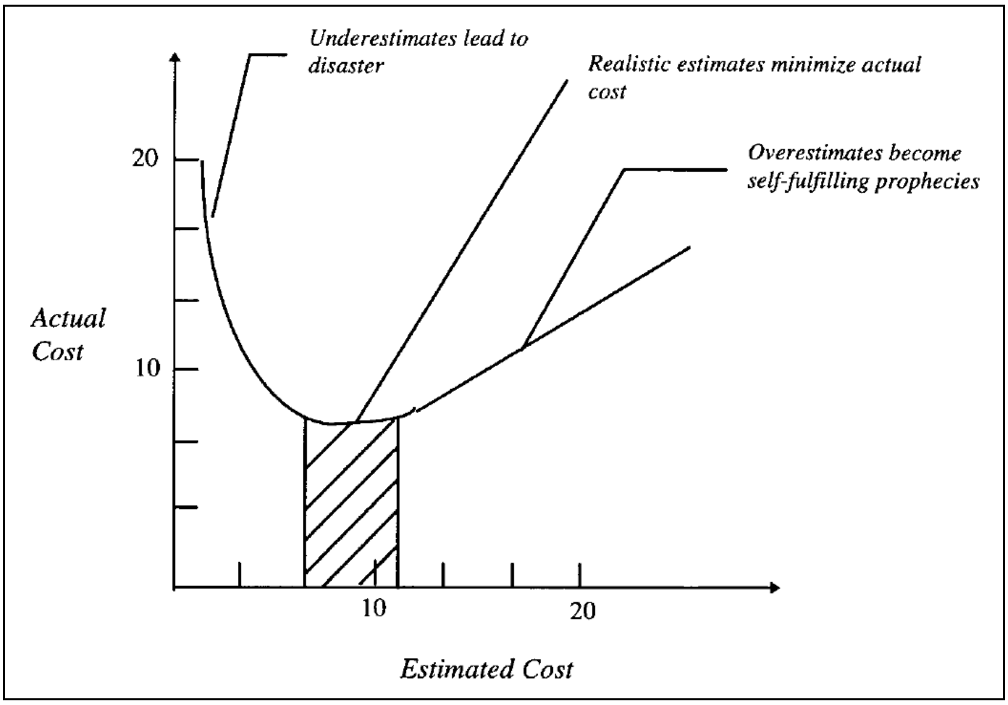


Figure 2-4: The Freiman curve [46].

Achieving accurate cost estimates depends very much on the available data. As Scanlan et al. [29] note, accurate cost estimation requires a relatively rich and detailed product definition. This is in comparison to other parameter estimations such as aerodynamic and structural performance. Figure 2-5 illustrates this mismatch of parameter accuracy.

In general, limited product data and technical definition will be available at the conceptual stages of design. This is one of the issues that the Implied Cost Evaluation System (ICES) [29] research aimed to address. The work from ICES allowed the uncertainty associated with cost estimation to be quantified and displayed as a 'cost tolerance'. By doing this, the quality of design decision-making improves as the designers are made aware of the areas of the cost uncertainty and risk.

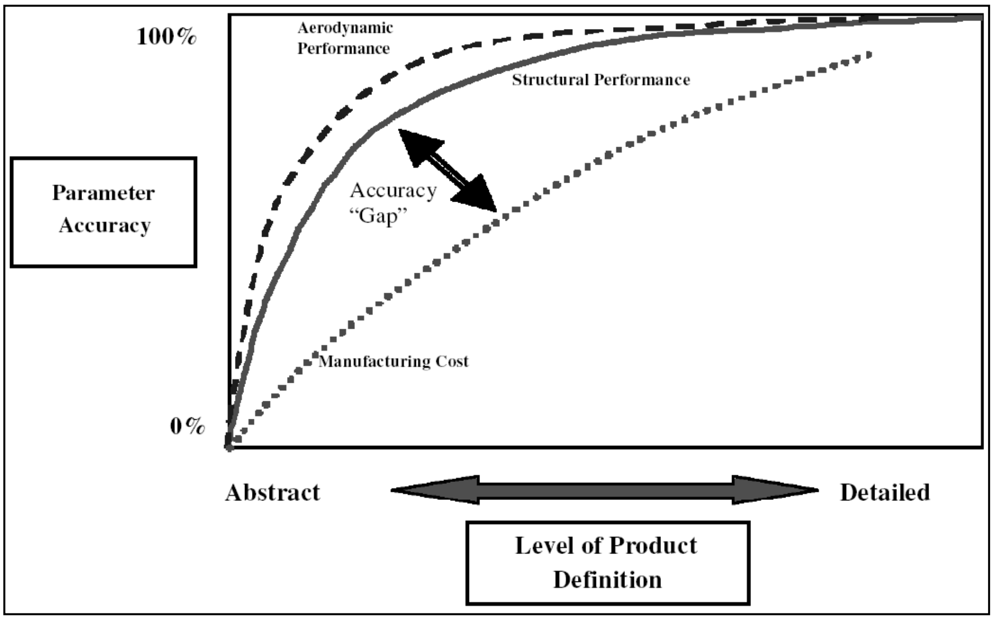
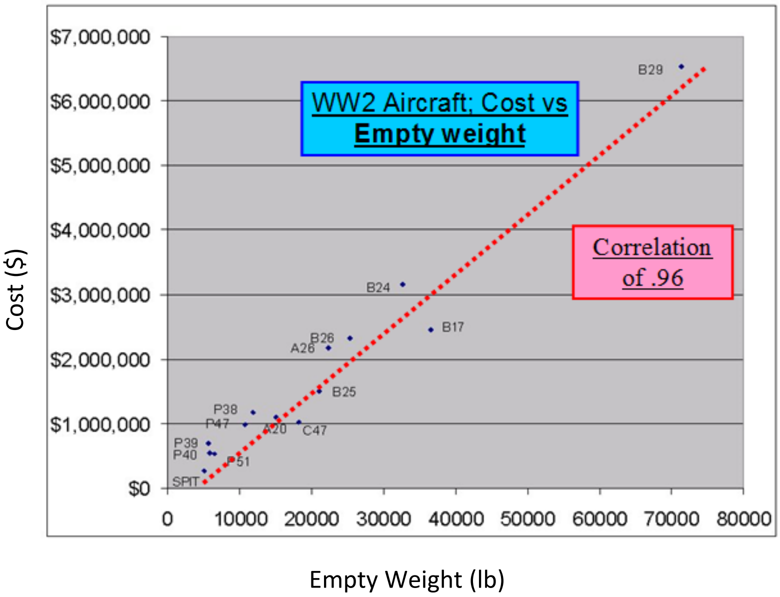
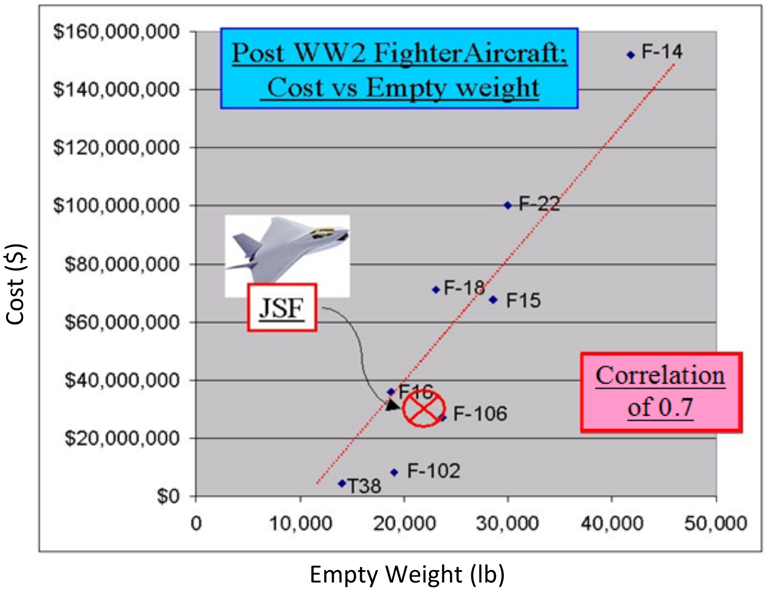


Figure 2-5: Accuracy mismatch between design parameter estimates [29].

It is also important to be aware of the limitations of the cost estimation method used. Results produced should be constantly verified and validated; especially in the conceptual design phases where high level parametric analyses are used. These methods can produce abstract relationships between cost driver and cost. Expert judgement is required to ensure that the results are sensible. Figure 2-6 [47] shows the relationships between weight and cost for WWII and post WWII military aircraft using linear least squares regression. The correlation coefficients for the two graphs are 0.96 and 0.7 respectively. The coefficients give an indication of how well the regression line approximates the real data points. The first graph shows excellent correlation between cost and empty weight for a wide range of aircraft. However the parametric relationship for post WWII aircraft has a poorer relationship. This is due to the technological diversity of the aircraft represented in the graph. Despite this, the lack of an alternative approach means that this analysis remains a useful guide in conceptual design.



(a)



(b)

Figure 2-6: Weight to cost correlation of (a) World War II and (b) post World War II military aircraft [47].

2.5. Life cycle cost

Life Cycle Cost (LCC) is defined by Fabrycky and Blanchard [48] to be the total cost associated with the acquisition and ownership of a product or system over its full life. There are several definitions of LCC from other authors in various disciplines but they do not deviate from Fabrycky's definition. LCC is a concept widely used in the aerospace, civil, manufacturing, defence and transport sector as a measure of the cost of ownership. It merges the details of conception, design, development, operation and disposal into a cost format that considers the time value of money.

The primary objective of a LCC analysis is to evaluate and/or optimise the LCC of a product or service. The aim of LCC is to allow informed decisions to be made based on costs. Some of the decisions LCC are used for are [49]:

1. Evaluation and comparison of alternative designs.
2. Assessment of economic viability of projects/products.
3. Identification of cost drivers and cost effective improvements.
4. Evaluation and comparison of alternative strategies for product use, operation, test, inspection and maintenance.
5. Evaluation and comparison of different approaches for replacement, rehabilitation/life extension or retirement of ageing facilities.
6. Allocation of available funds among the competing priorities for product development/improvement.
7. Long term financial planning.

As with any costing technique there are limitations to the use of LCC analysis. Barringer [50] gives a summary of the most often cited limitations. These are:

1. LCC outputs are only estimates and accuracy is dependent on the inputs provided;
2. LCC models require a large volume of data which can be difficult and expensive to retrieve;
3. LCC results are not good budgeting tools and are only effective as design comparison/trade-off tools;
4. LCC analysis requires a specific scenario and demands details such as how the model will age with use, how damage will occur, how long the model will survive and how cost processors (labour costs, material costs, parts consumption, maintenance costs) will function for each time period.

Just as it is in cost literature, most papers on LCC suggest that 70% of the total LCC of a product or system is determined by product design. Regardless of the contention of the costs committed in the design stage, a designer must know the cost implications of design decisions in order to reduce LCC. Tools which can help a designer accurately estimate LCC are vital to creating cost competitive products.

2.5.1. LCC Process

Kawauchi and Rausand [51] reviewed several LCC analysis processes (including the International Electro-technical Commission (IEC), International Organization for Standardization (ISO), Society of Automotive Engineers (SAE), and British Standards Institution (BSI)) and identified the basic stages involved. The stages are as follows:

1. Problem definition;
2. Cost element definition;
3. System modelling;
4. Data collection;
5. Cost profile development;
6. Evaluation.

This process is illustrated in Figure 2-7 with a summary of the sub-activities involved in each stage. A detailed description of these stages can be found in the BSI guide to life cycle costing [49].

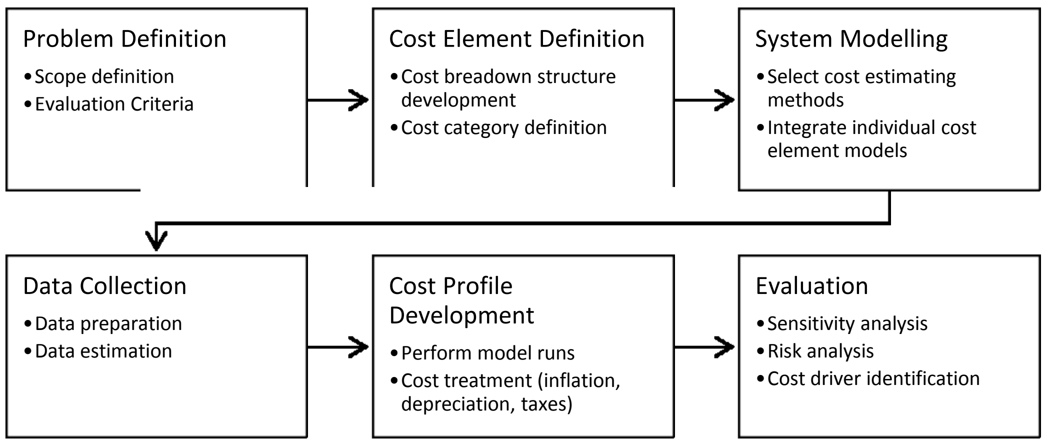


Figure 2-7: Life cycle cost analysis process.

2.5.2. Cost Breakdown Structure

To estimate total LCC, it is necessary to break LCC down into its various cost elements. The identification of the cost elements is subjective upon the purpose and scope of the LCC study. Barringer [50] advocates that a LCC study should only include the relevant cost elements and to discard elements which do not greatly influence LCC. Barringer lists various different LCC models. These are:

- LCC = non-recurring costs + recurring costs;
- LCC = initial price + warranty costs + repair, maintenance, and operating costs to end users;
- LCC = manufacturer's cost + maintenance costs and downtime costs to end users;
- LCC = acquisition costs + operating costs + scheduled maintenance + unscheduled maintenance + conversion/decommission.

An approach commonly taken is to initially categorise LCC elements into Acquisition costs and Ownership costs [5, 49-51]. Acquisition and Ownership costs can then be further broken down and charted into a cost breakdown structure (CBS). There is no universally accepted CBS of LCC given its application in disparate disciplines, so a CBS depends very much on the stage it is being used at and the kind of information it is required to present. Figure 2-8 shows an example of a CBS adapted from Gibbs [5]. It is by no means the most comprehensive or representative but it is the one which is most applicable to aero-engines. This CBS is split into the sub-categories of Research, Development, Test and Evaluation (RDTE), Investment, Operation and Support and Disposal.

While LCC is the total cost a system incurs over its entire life, there are differences between the cost issues that will be of interest to the person designing the product and the firm developing the product in a LCC analysis [4]. At the managerial level the total cost of the product will be of interest. Designers however, will only be interested in the costs that they can control.

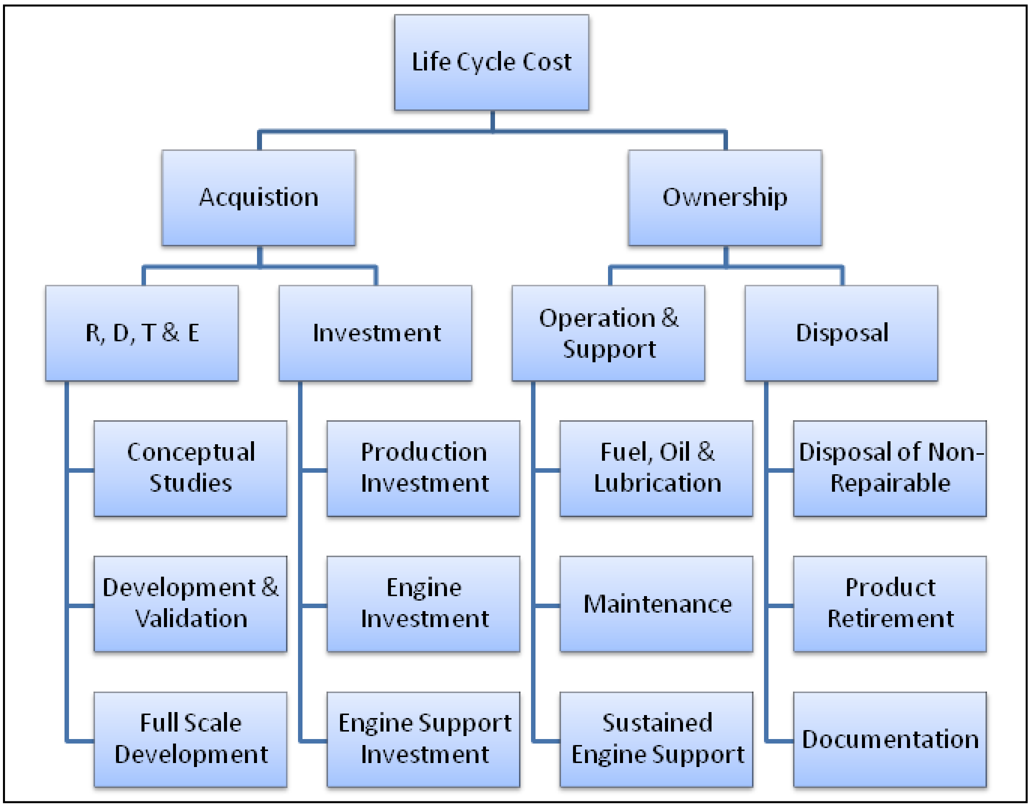


Figure 2-8: Life cycle cost breakdown structure (CBS) [5].

2.5.2.1. Research, Development, Test and Evaluation

This category considers all costs relevant to the research, design and development of the product and its support systems. These costs are managerial ones and will not be of interest to the designer [4]. However, although the costs from this phase will be beyond the influence of designers, it is precisely during this phase that decisions affecting the majority of LCC will be made [5]. As the design phases progresses, the opportunities for alternative approaches will diminish. It is therefore important that LCC is considered from the start of the design process.

2.5.2.2. Investment

The costs in this phase involve manufacturing, facility construction, process development, production operations, quality control and initial logistic support requirements [7].

Before in the RDTE phase, a LCC analyst would be looking into costs internal to the company. In this phase, the LCC analyst would focus on producing analyses for the customer in order to secure business. At this point in the system's life cycle, the RDTE costs would already have been well understood and easily quantifiable. Therefore, the biggest uncertainty will come from the estimation of the operation and support costs. Given the commitment involved in drawing up the customer's proposal, this stage is the one which bears the most risk in the entire process [5].

2.5.2.3. Operation and Support

Operation and support costs comprise the operations of the product in the field, product distribution, maintenance and logistic support. These costs are the most significant portion of the LCC and the most difficult to predict [4]. According to US government records, the cost of operating and supporting an item may exceed the initial purchase price of that item as much as ten times [52]. Products should be designed to reduce maintenance times to minimise the product out of service times and hence minimise the incurred financial loss [4]. At this stage, the LCC analyses should be extended to consider the implications of having to make product modifications [5]. The change in the LCC as a result of the modification will be used to determine the cost viability of different options.

2.5.2.4. Disposal

The true LCC of a product/system will only be known at this stage, as uncertainties in prior cost analyses diminish. Disposal costs can vary a great deal for the product being considered. A product/system involving radioactive materials will incur a high disposal cost. Disposal costs for defence aero-engines, in comparison, are minimal because they will be sold on to other parties, scrapped or recycled [5]. This is the same in the case of civil aero-engines. The largest proportion of disposal costs will come from the cost incurred during disassembly.

2.5.3. Dependability and Life Cycle Cost

Dependability is defined by the British Standards Institute (BSI) to be a collective term which describes a product's availability performance and influencing factors, such as reliability, maintainability and maintenance support [49]. Dependability can have a significant effect on LCC, especially for products with high operating costs. High acquisition costs may be able to improve the dependability of a product. This would in turn reduce failure and maintenance costs; and by extension, ownership costs. Conversely, lowering initial costs may have a detrimental effect on dependability. Since LCC aims to operate assets to minimum cost, the estimation of operating and maintenance costs is vital in performing a LCC analysis. In order to optimize product design to the lowest LCC, reliability, maintainability and other dependability management considerations should be an integral part of the design process.

Costs associated with dependability which need to be considered in LCC evaluations are [49]:

1. Unavailability costs (including costs associated with loss of product function);
2. Warranty costs and costs for warranty-type agreements;
3. Liability costs.

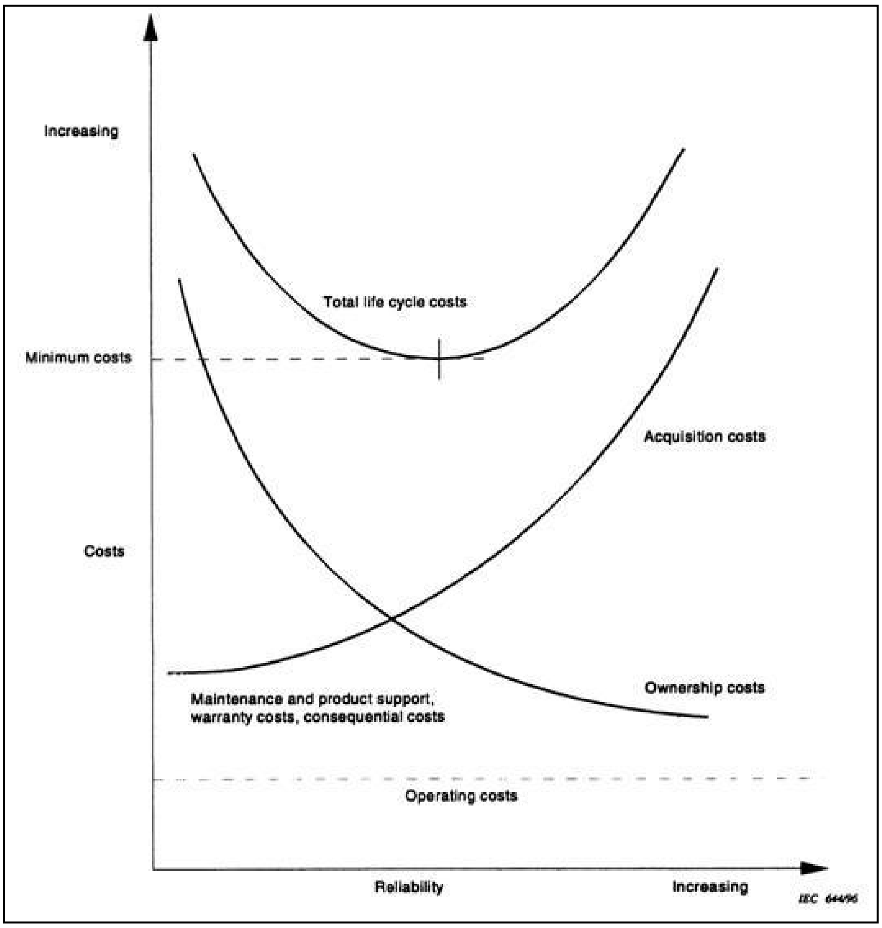


Figure 2-9: Simplified relationship between reliability and life cycle costs [49].

Figure 2-9 shows a simplified conceptual relationship between dependability and LCC. It illustrates that minimising LCC requires a trade-off between ownership and acquisition costs.

2.5.4. Review of LCC Models

Asiedu and Gu [4] state that most LCC models fall into three general categories: conceptual, analytical and heuristic. Conceptual models encompass those which consist of a set of hypothesized relationships expressed in a qualitative framework. Analytical models are usually based on mathematical relationships which describe an aspect of a system/product under certain conditions. Heuristic models are ill structured analytical models which use a trial-and-error method of problem solving to generate a feasible solution. In reality, many LCC models are a combination of two or more of these categories. This is typified in Sandberg et al. [53] who developed a methodology to calculate the LCC of jet engines in the early design phase. It is able to assess how decisions between design, performance, manufacturing and maintenance activities affect each other. The developed approach used a parametric cost estimation technique which links to a geometric process definition and a knowledge-based database to support cost estimation activities. The rest of this section highlights other LCC models with an aerospace focus as aircraft design is the product of a complex process which involves many different fields and techniques [54].

Marx et al [55] developed a methodology which uses a hierarchical cost model structure to determine the life cycle effects of design and manufacturing alternatives for the major structural components in the wing of a High Speed Civil Transport (HSCT) aircraft concept. The model integrates a bottom-up cost estimation model of the HSCT wing with a top-level parametric LCC model. Through the integrated model, benefits in cost reductions incurred as a result of technology improvements could be directly assessed. Johnson [56] developed a methodology which would allow the identification of an aircraft concept which would meet mission requirements and have the lowest LCC. The approach integrated an existing conceptual design and analysis code, Flight Optimization System (FLOPS), with a LCC model. The LCC model was made up of cost models from various sources such as Science Applications Inc., Rand Corporation, American Airlines, and Lockheed-Georgia Company. With this model, a comparative study was performed which looked at optimising an aircraft for minimum LCC, gross weight, fuel burn, acquisition cost, and direct operating cost. This study demonstrated how the consideration of LCC can influence aircraft design. Tan [57] proposes an approach to LCC based on the Zachman framework model and object oriented modelling for the better management of complex systems. This approach was to be tested on an integrated aircraft wing. For a complex system, the relationships between multiple objects, attributes and operations of the LCC model are "complicated and interwoven". Each component in the wing is considered an object and is subjected to the 5W1H questions (why, who, where, when, what, and how) used in the Zachman framework. Once the cost estimation relationships (parametric, analogous, etc.) of each object are established, LCC is calculated through the aggregation of the individual cost elements. Curran presents a LCC methodology which facilitates the integration of design and manufacturing modelling at the concept design stage. It was developed to generate an aircraft fuselage panel design optimised for direct operating cost (DOC). This approach hinges on the integrated capabilities of the Dassault V5 platform. Firstly, a spread sheet model is used to generate the geometric definition of the panel optimised for DOC. The 3-D computer aided design (CAD) module, CATIA, is then used to create a solid model of the component of interest. Additional parameters can be modified through an interface coded in Visual Basic provided by the V5 platform. Finally, the information and data generated in the previous three steps is input into the DELMIA manufacturing simulation platform. Thokala et al. [58] developed a framework which could estimate the LCC of unmanned aerial vehicles (UAV). It integrates a hierarchical acquisition cost model with a discrete event simulation model which estimates maintenance and operation costs. The model was then used to perform trade-off analysis and cost-based optimization.

A survey of LCC models from all fields in the last ten years are presented in Newnes [59], Waghmode [60] and Cheung et al. [61]. The following section identifies several

LCC modelling issues that have been found in models developed for aerospace and other general fields.

2.5.5. LCC Modelling Issues

The LCC concept was popularised by the US Department of Defense (DoD), who were motivated by studies which showed operation and support accounting for up to 75% of typical weapon systems costs [10]. However the initial methodologies developed by the DoD were used for procurement rather than design. In 1984, the US National Science Foundation sponsored a conference in recognition of the need for economic based engineering methodologies. Out of 34 research opportunities identified and prioritised, the two research areas with the highest scores were 'economic evaluation of design trade-offs over the life cycle' and computer-aided cost estimating. Since then there has been significant work performed in these two areas which Asiedu and Gu [4], Curran et al. [22] and Newnes et al. [59] have reviewed in great detail. The sections below discuss some of the issues regarding LCC model development.

2.5.5.1. Application Specific Models

In general, costing tools have been found to be application specific and highly customised [4, 22] as these models are developed only for use in specific phases of the product life cycle or for specific operations in a particular life cycle phase. This trend reaffirms Dhillon's [62] belief that the development of a standard LCC model may be unrealistic and he suggests several factors to why this is the case. These are:

1. The diverse nature of the problem;
2. The use of different types of equipment, devices or systems;
3. The inclination of the user.

The general recommendation for LCC modelling is that the analysis should be tailored to the objectives and scope of the study to obtain the maximum benefit from the analysis effort [49]. This is because the modelling process can easily become very complicated by trying to undertake too large an effort or by proceeding in the wrong directions [7]. This presents a problem when creating tools for whole life costing as a new cost model will have to be constructed for every design scenario presented.

2.5.5.2. Modelling Approach

A LCC model can range from being a simple collection of cost estimation relationships to a complex computer simulation. In cost engineering, the classic cost estimating techniques used are the Analogous, Parametric and Generative/Bottom-up methods. Some of the more advanced techniques to have emerged include Feature-based modelling, Fuzzy-logic and Artificial Neural Networks. Each of these methods naturally has their own strengths and weaknesses, therefore the choice of a modelling approach will rely on the objectives, scope, level of detail and results desired; constrained by the resources available; such as finances, personnel, time and tools. Furthermore, Scanlan et al. [29] note that cost modelling tools need to "deal with the multiplicity of levels of abstraction associated with an emerging design."

With regards to concept design, the parametric cost estimating approach has been widely used in the aerospace industry [22, 63]. This approach uses Cost Estimating Relationships (CERs) and mathematical algorithms or logic to establish cost estimates [36]. This technique typically uses regression analysis based on historical and technical data to form CERs which link changes in one or more parameters, known as cost drivers, to a change in cost. This method is popular because of its ease of use and ability to operate with abstract design definition. Despite this, there are several limitations to its implementation. Firstly, parametric models cannot be used beyond the scope of the data used in its validation. Thus, it is unable to predict cost changes in novel designs where historical data is non-existent. Secondly, CERs do not always have direct links between cost drivers and cost and in doing so they can obfuscate genuine causes of cost changes. Collopy [64] discusses how weight is not a strong cause of cost despite the fact that it is used in popular aerospace cost models. He notes that in many cases design choices that reduce weight actually increase cost; such as adding cooling holes to a turbine blade. A generative approach may be able to address these issues as it relies on detailed engineering analysis to generate a cost estimate. These models show direct cause and effect between cost drivers and cost which is highly desirable in a design tool. On the other hand, this method is more difficult and time consuming to implement as it requires complete understanding of the cost driver to cost relationship. Consequently it also requires more detailed design information which may not be available at the early design stage. As LCC has such a broad scope, different cost estimating methods will need to be used to calculate the various cost elements; some examples are found in Marx [55] and Sandberg [53].

2.5.5.3. Model Complexity

For a product such as an aero-engine, causal relationships between cost drivers and cost can be complex. A large proportion of recurring costs of an aero-engine is dependent on its fuel burn and reliability. These properties are emergent properties which are consequent on the relationships between the various sub-systems involved. Modelling these kinds of properties may necessitate a multi-disciplinary analysis approach. Examples of this are evident in cost modelling literature; Thokala [58] developed a framework which links computational fluid dynamics (CFD) analyses, operations analyses and acquisition cost to calculate the LCC of Unmanned Air Vehicles (UAV). Sandberg [53] utilises a Knowledge Based Engineering (KBE) system to support computer aided design (CAD) tools which assess how geometric design changes made to a jet engine component influences its manufacturing cost and performance. Rao [65] developed a workflow which links CAD, finite element (FE) analysis, fatigue life prediction models and cost estimation models to investigate performance trade-offs between low-weight and low-cost designs of turbine disks.

With multi-disciplinary approaches, such as those described above, interfaces exist between the sub-components of the analyses. An interface is defined as “a common boundary or interconnection between systems, equipment, concepts, ...” [66]. As the model grows in complexity, so will the number of interfaces, interactions and dependencies making the model difficult to maintain and manage. Additionally, as the product becomes more defined, more design detail becomes available allowing for more accurate analyses to be performed. However as Price [67] notes, the relationships between the sub-components of the analyses are determined by analysis methodology type and fidelity. This is seen in cases where detailed analyses are needed to support the system definition at a higher level. Consequently, management of these relationships is required as interfaces, interactions and dependencies will change or even disappear as the cost model changes during the different design stages.

2.5.5.4. Model Transparency

As mentioned earlier, LCC tools can involve the integration of various types of analyses. These tools, such as those seen in Thokala [58] and Sandberg [53], have Graphical User Interfaces (GUI) which help to simplify the operation of the LCC model This unfortunately also obscures the mechanics of the model which encourages scepticism over the final results. In a design scenario, a transparent model is necessary to foster understanding of the cause and effects of design parameters on resultant LCC. This factor becomes even more important when dealing with complex models given the additional considerations of interfaces, interactions and dependencies and how they change as the product/system becomes more defined.

2.5.6. Desired Characteristics of LCC models

From the discussed issues of LCC models, several desirable characteristics for LCC models were identified. These are:

- Customisability – This is defined to be the ability to modify, make, or build according to individual specifications or preference. The LCC model needs to be customisable so that it can be adapted to suit different study objectives and approaches.
- Transparency – This is the property of being easily understood or obvious. A transparent model will also allow the model to be more easily debugged and maintained. Transparency facilitates future modifications as it will make it easier to identify where changes need to be made.
- Modularity – This is the degree to which the components of a system may be separated and recombined. In software design, modularity is used in the logical partitioning of the software. This helps to make complex software manageable for the purpose of implementation and maintenance. Keeping the LCC model modular will assist the management of the interactions within

the integrated analyses as it allows interfaces to be clearly defined. Modularity also allows varied modelling approaches to be added or removed as necessary as long as the interfaces with the other modules can be resolved.

2.6. Model Based Systems Engineering

Model Based Systems Engineering (MBSE) is an approach which makes modelling activity central to the Systems Engineering (SE) process. It will thus be useful to include a brief explanation of SE. The International Council on Systems Engineering (INCOSE) Systems Engineering Handbook [68] provides three definitions of what SE is. These are:

- "Systems engineering is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect." [68];
- Systems engineering is an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system. [69];
- Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. [70].

In essence, SE is a multi-disciplinary field of engineering which concentrates on how complex systems/products are designed and managed over their entire life cycle. Figure 2-10 gives an overview of the SE process. A discussion of the SE process is not within the scope of this thesis. Readers should consult the Systems Engineering Handbook [68] for further information.

MBSE is an approach which uses systems modelling in the systems engineering process in the specification, design, integration, validation, and operation of a system. It is an approach which seeks to replace the document-centric approach for describing systems, practiced by systems engineers in the past, with a model-centric approach. In doing so MBSE aims to impart the following benefits [71]:

- Development of a formalized practice of systems engineering through the use of models;
- Integration of multiple modelling domains across system life cycle;
- Improvement of communication among system/project stakeholders;
- Improvement of management of complexity;
- Enhancement of knowledge capture and reuse.

These are characteristics which would be of benefit to the LCC modelling process given the modelling issues laid out in section 2.5.5.

Estefan [73] summarises the leading MBSE methodologies. These are:

- IBM Telelogic Harmony,
- INCOSE Object-oriented Systems Engineering Method (OOSEM),
- IBM Rational Unified Process for Systems Engineering (RUP SE),
- Vitech Model-Based Systems Engineering Methodology,
- JPL State Analysis (SA),
- and the Dori Object-Process Methodology (OPM).

Several of these methodologies above are internally developed processes and methods from various companies and organisations. While these methodologies are not official industry standards, best practices could emerge from their application.

2.6.1. SysML™

In this thesis, system modelling is performed with the INCOSE Object Oriented Systems Engineering Method (OOSEM) as it is non-proprietary and utilises Object Management Group (OMG) SysML™; which is recommended by INCOSE to be the industry standard for MBSE [73]. SysML™ is a graphical modelling language developed by the OMG and INCOSE specifically for use in SE. It is an extension of a subset on the Unified Modelling Language (UML) designed to support the specification, analysis, design, verification and validation of a broad range of systems.

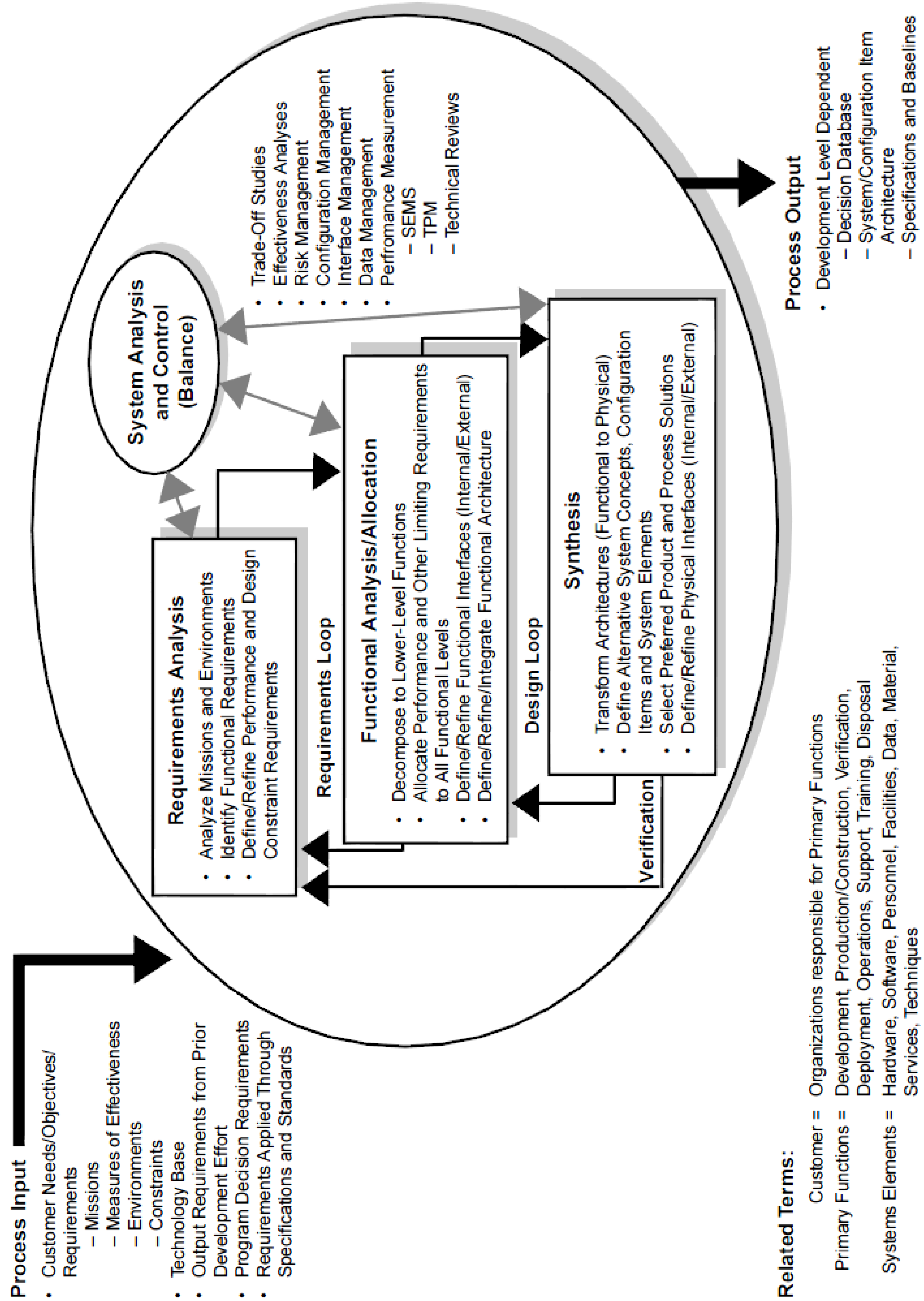


Figure 2-10: The Systems Engineering process [72].

SysML™ allows the system under study to be analysed from four aspects. These are:

1. Structure
2. Behaviour
3. Requirements
4. Parametric

Examples of each aspect are illustrated in Figure 2-11. SysML™ is a relatively new technology; OMG SysML™ 1.0 was only accepted by OMG for technology adoption in April 2006. Since then SysML™ has generated significant research interest and has been adopted in a broad range of aerospace and commercial applications [74].

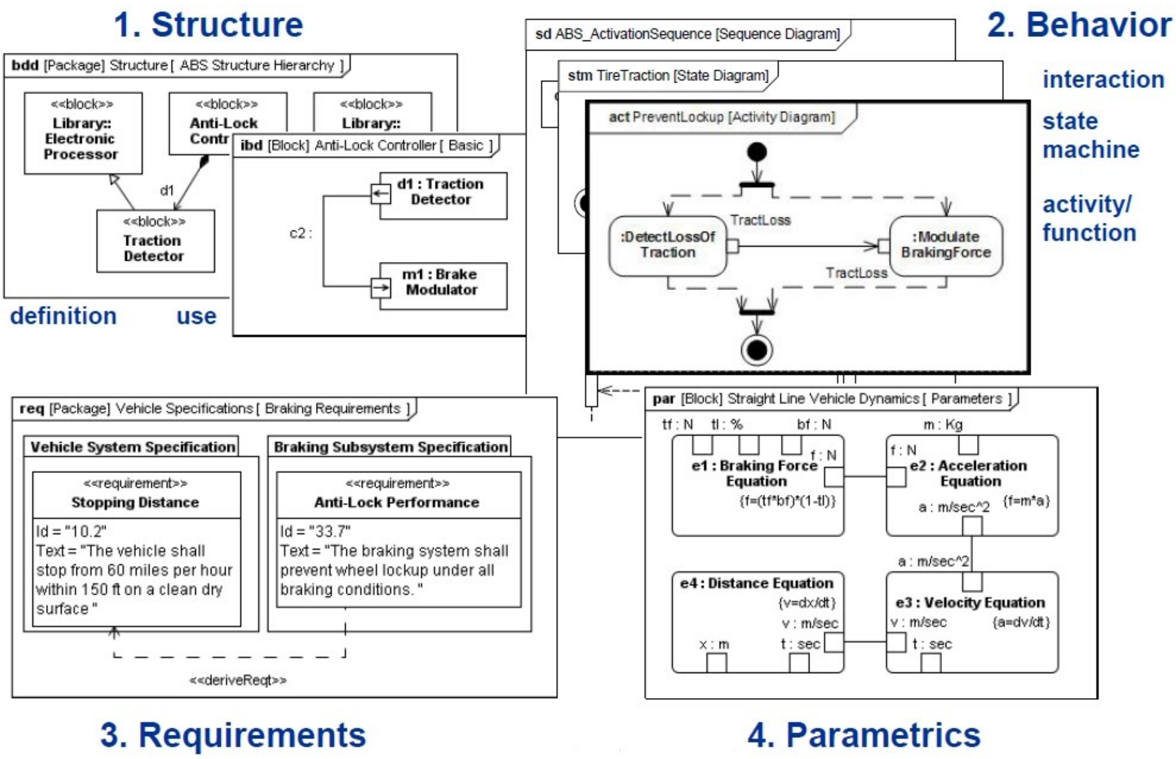


Figure 2-11: The four pillars of SysML™ [71].

2.6.2. INCOSE Object-oriented Systems Engineering Method (OOSEM)

The Object Oriented Systems Engineering Method (OOSEM) "integrates a top-down, model-based approach that uses OMG SysML™ to support the specification, analysis, design and verification of systems" [73]. Additionally, object oriented concepts are used to enable the design of more "flexible and extensible" systems to support evolving technology and changing requirements.

OOSEM was founded on SE principles; therefore this methodology focuses on how a model based approach can be used to support existing SE practices. Estefan [73] states that there are six main development activities in OOSEM. These are:

1. Analyse Stakeholder Needs;
2. Define System Requirements;
3. Define Logical Architecture;
4. Synthesise Candidate Allocated Architectures;
5. Optimise and Evaluate Alternatives;
6. Validate and Verify System.

Figure 2-12 shows these activities along with MBSE artefacts which need to be produced listed in bullet points.

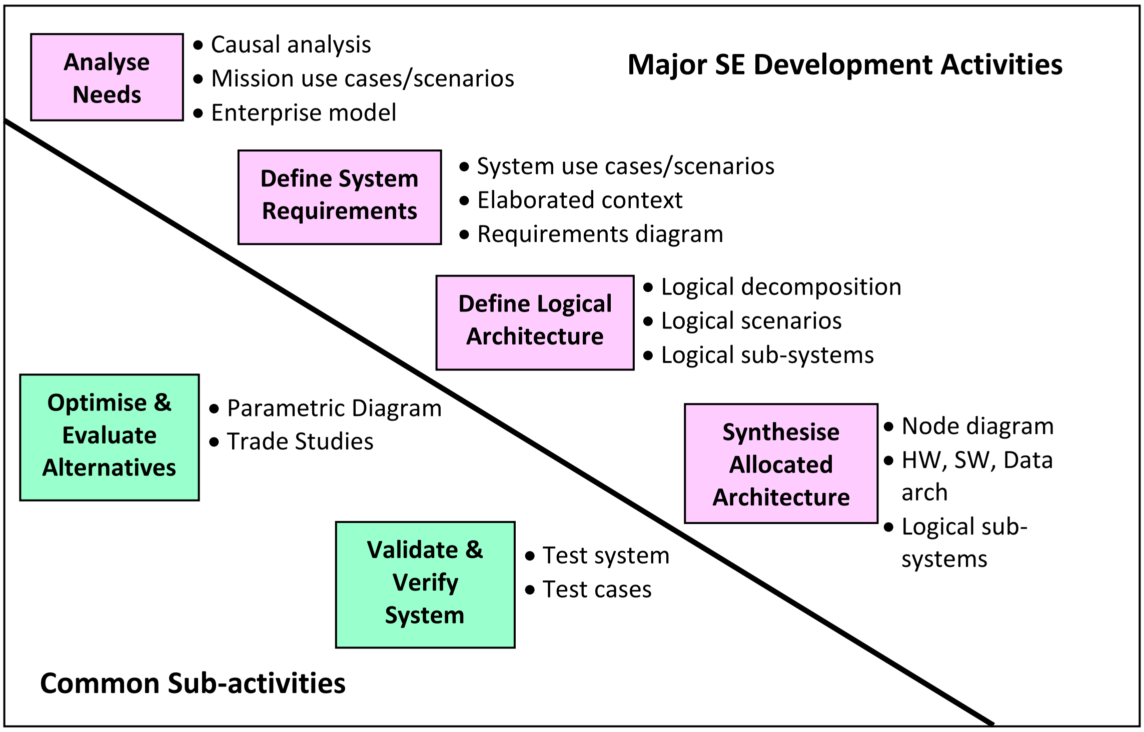


Figure 2-12: OOSEM activities and modelling artifacts [73].

The term "object oriented" used in the title refers to a philosophy of how the system is viewed rather than the programming paradigm used in software development. In OOSE, a system is regarded to be composed of "objects". Each of these "objects" will possess attributes, functions, parameters and constraints.

2.6.2.1. Analyse Stakeholder Needs

In the generalised OOSEM process, the purpose of this activity is to analyse the current status of the system or enterprise of interest in order to identify areas of limitation and potential improvement. In terms of developing integrated LCC models, this activity should be used to describe the context in which the LCC model is to be used and to assess and capture the stakeholders involved and what their roles are. Causal analysis techniques are also used at this stage to identify potential factors within the system which cause an overall effect. These will used as the basis for deriving the requirements of the LCC model to be created.

2.6.2.2. Define System Requirements

In this activity, stakeholder requirements and constraints are reviewed, assessed, prioritised, and balanced. These are then transformed into functional and technical statements which describe a system capable of meeting the demands of the stakeholders. System-level use cases and scenarios can be used to reflect the operational concept for how the system is to be utilised. An example is shown in Figure 2-13.

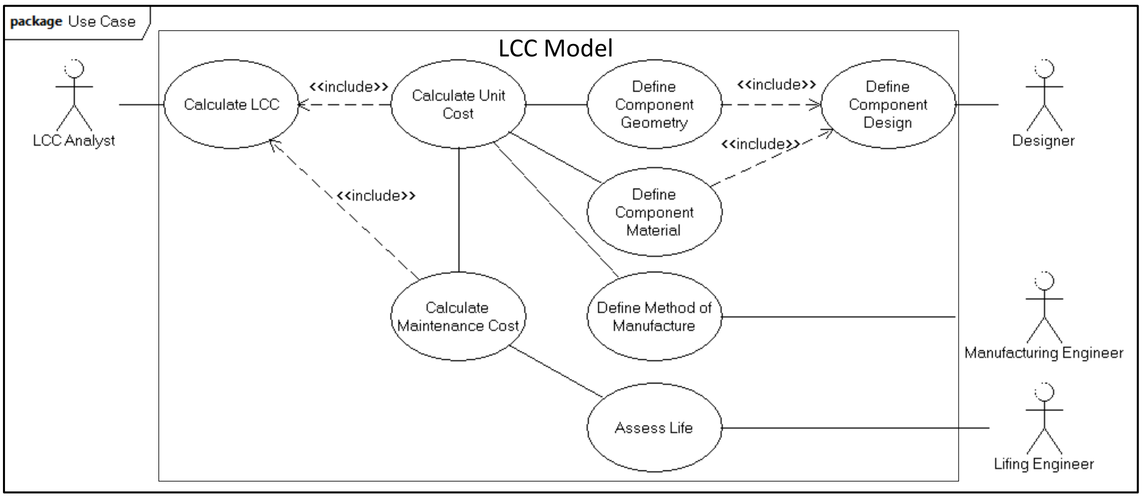


Figure 2-13: Use case diagram.

2.6.2.3. Define Logical Architecture

In this activity, the system is decomposed into logical components that interact to satisfy the system requirements. These logical components are used to capture the functionality of the system. The logical components are abstractions of the components which implement the system without imposing constraints on how the system is to be implemented. For example, the LCC model described in Figure 2-13 could have “Component Definition”, “Manufacturing”, “Lifing” and “Costing” components in its initial decomposition. These logical components are subject to refinement based on repartitioning of their functionality and properties.

2.6.2.4. Synthesise Candidate Allocated Architectures

The allocated architecture refers to the relationship between the physical components of the system; which can include hardware, software, data, and processes. The architecture is defined in terms of its physical components and relationships, and their distribution across system nodes. The system nodes represent a partitioning of components based on their physical location or other distribution criteria. A node physical architecture is defined where each logical component in each node is allocated to physical components. Analysis and trade studies are performed to evaluate, select, and refine the preferred architecture.

2.6.2.5. Optimise and Evaluate Alternatives

This activity occurs throughout the OOSEM activities as required. The aim is to optimise or to conduct studies to determine the most suitable candidate architecture of the system. This activity includes identifying the analysis that is needed, defining the analysis context, capturing the constraints in a parametric diagram, and performing the engineering analysis. For LCC models, this concerns the tools and methods to be applied and how they are to be integrated.

2.6.2.6. Validate and Verify System

This activity serves to ensure that the developed system meets it requirements. Plans, procedures, and methods are used to verify this. Some examples are inspections plans, demonstrations, analyses, and certifications tests. OOSEM supports this process through the use of system-level use cases, scenarios, and associated requirements. These form the basis for developing test cases and the associated verification procedures.

2.6.3. Using MBSE to Address LCC Modelling Issues

2.6.3.1. Application Specific Models

It was highlighted that LCC models were generally application specific which made the concept of a generalised LCC model impractical. This implies that a framework is required which would assist in the creation of application specific models. In MBSE, models are used in all areas of the design process to capture knowledge. The ability to reuse these models will help reduce the effort in developing new designs. Additionally, MBSE provides a consistent language to be used across the different system domains. This will aid communication and understanding in cases where the LCC model requires a multi-disciplinary approach.

2.6.3.2. Modelling Approach

LCC modelling approach varies with the information available, tools, and required fidelity. There is however a common element with these models; which is that they are all software models. This is relevant because MBSE was borne out of Model Driven Engineering; a software development methodology. Many of the MBSE methodologies presented later in this chapter were developed for software engineering purposes. The general purpose MBSE modelling language OMG SysML™ was adapted from OMG Unified Modelling Language™ (UML); a standardised general-purpose modelling language from the field of software engineering. As LCC modelling involves the creation of software tools, this makes MBSE suitable for the specification, design, and documentation of LCC models.

In situations where the system to be modelled is not software related, the use of a generalised systems modelling language may not be appropriate. For example, describing a hydraulic circuit with SysML™ instead of standard hydraulic circuit symbols would be laboured and inefficient. It would then be more effective and convenient to use a Domain Specific Language (DSL) modelling approach. DSLs provide suitable views and constructs for a particular domain. The DSLs would then have to be integrated with a general systems modelling language, such as SysML™, so that a coherent systems model can be created. There are currently two options to do this [75]. The first option involves an ontology based mapping of different DSL meta-models and linking the domain-specific models in a common data-centric model repository. The second requires the use of a language workbench, which is an integrated development environment specially dedicated to multiple domain-specific languages. The use of DSLs will not be considered in this thesis to keep the scope manageable.

2.6.3.3. Model Complexity

It was found that many LCC models involve multi-disciplinary analysis. As mentioned earlier, as these models grow in complexity, the number of interfaces, interactions and dependencies increase making the model difficult to maintain and manage. To help manage complexity, it is useful to consider the three categories of system complexity defined by Moses [76]. These are:

1. Behavioural complexity - A measurement of how difficult it is to predict the external behaviour of the system;
2. Interface complexity - A system has a complex interface if it has numerous connections to other systems;
3. Structural complexity - A system is structurally complex if the interconnections, interactions or interdependencies between its components are difficult to describe or understand.

MBSE allows management of complexities mentioned above through the formal representation of these aspects of the system engineering problem. SysML™, as discussed later, provides constructs to model system requirements, behaviour, structure, and parametrics.

2.6.3.4. Model Transparency

MBSE can help keep LCC models transparent as it is explicit and compact which enhances shared visualization, understanding and communication. In comparison, a document-centric approach utilizes prose which can be interpreted in different ways. MBSE also fosters collaborative environments across multiple disciplines as it provides a consistent language for the creation of integrated models.

2.6.4. Future of MBSE

At this point in time, MBSE is still a maturing concept. The Systems Engineering Vision 2020 [77] report noted that current MBSE processes and methods were generally practiced in an ad hoc manner. Cloutier [78] observed that there was no single view which defined MBSE. A lack of MBSE standards further exacerbates matters. The emergence of different MBSE tools means that various training requirements and methods have been imposed on the community. This also results in a lack of tool interoperability which has stifled the widespread deployment of MBSE.

The Systems Engineering Vision 2020 report goes on to highlight several key developments which need to be made in MBSE. These are:

- Domain-specific modelling languages and visualisation to enable systems engineers to better abstract the respective domains;
- Reuse of model libraries, taxonomies and design patterns;
- Standards which support integration and management across a distributed model repository;
- Reliable and secure data exchange via published interfaces.

Figure 2-14 shows the vision for MBSE where the above developments will help to create an integrated collaborative environment. With such capabilities, system development times could be drastically reduced as well as improving system quality and availability. LCC models, as mentioned before, are tailored to its objectives and can become multi-disciplinary. An MBSE based collaborative environment will be able to enhance the creation these kinds of models.

Work done by Peak et al [13] has added significant value to MBSE. SysML™ has, as they describe, “enough expressivity to be applied across diverse disciplines during system development”. On this premise they have developed a methodology for integrating a system model with multiple engineering analysis models and dynamic simulation models; Figure 2-15 shows the different tools used. The interfaces between the SysML™ tools and the analysis tools were developed using a combination of commercial interface and transformation tools, such as VIATRA and XaiTools, and custom built interfaces using C#, Java and Visual Basic.

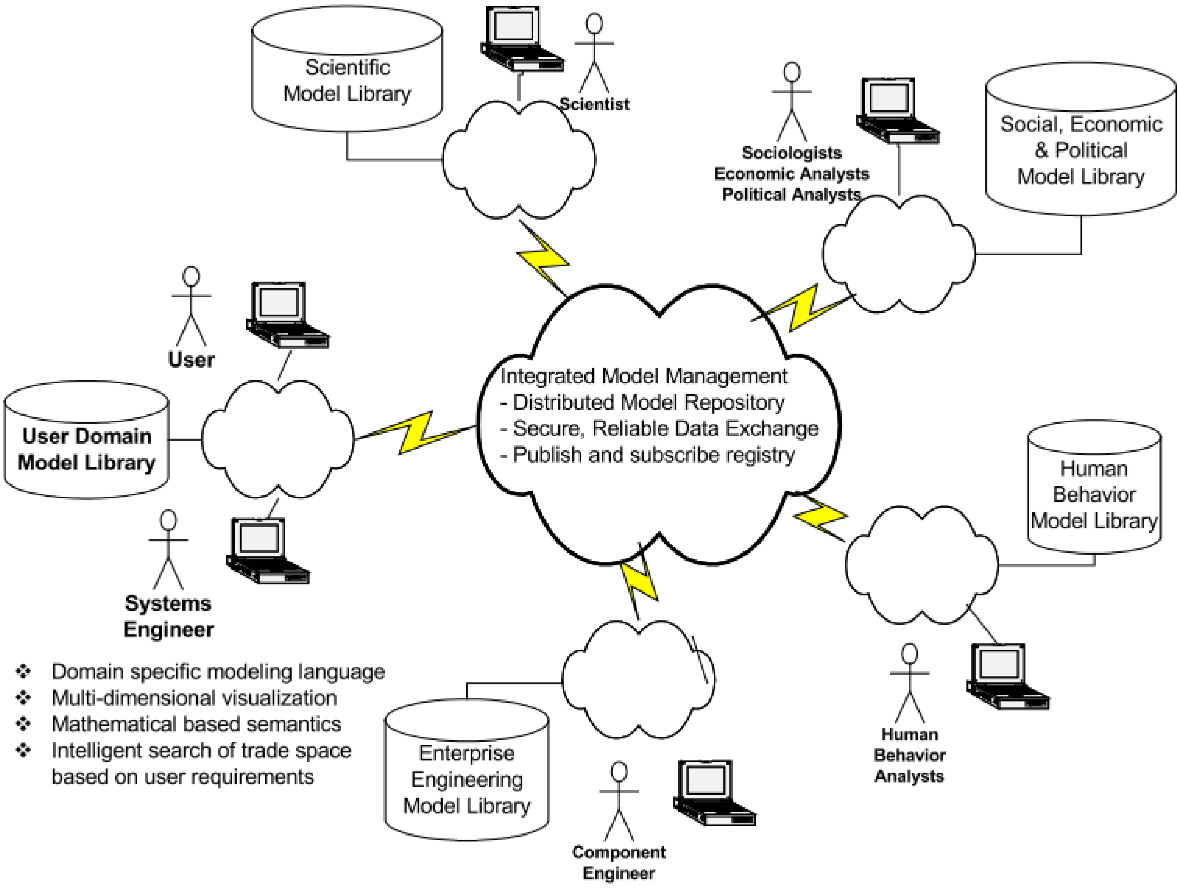


Figure 2-14: MBSE vision for cross-domain model integration [77].

The significance of this work is that, in addition to the benefits of MBSE discussed earlier, it is possible to use MBSE to automate the generation of integrated analysis models. This will significantly reduce design cycle time and increase design analysis and trade-space exploration.

As with any new emerging technology, there are inhibitors to the proliferation of MBSE. The practice of MBSE will involve a cultural change. MBSE is intended to be the primary artefact for system specification and design rather than the traditional document-based approaches used presently. There is also an inherent difficulty in implementing MBSE across an entire organisation. Finally, as mentioned earlier, the ad-hoc practice of MBSE and a lack of standards have given rise to a number of varying methodologies. This makes the education and training of MBSE even more difficult.

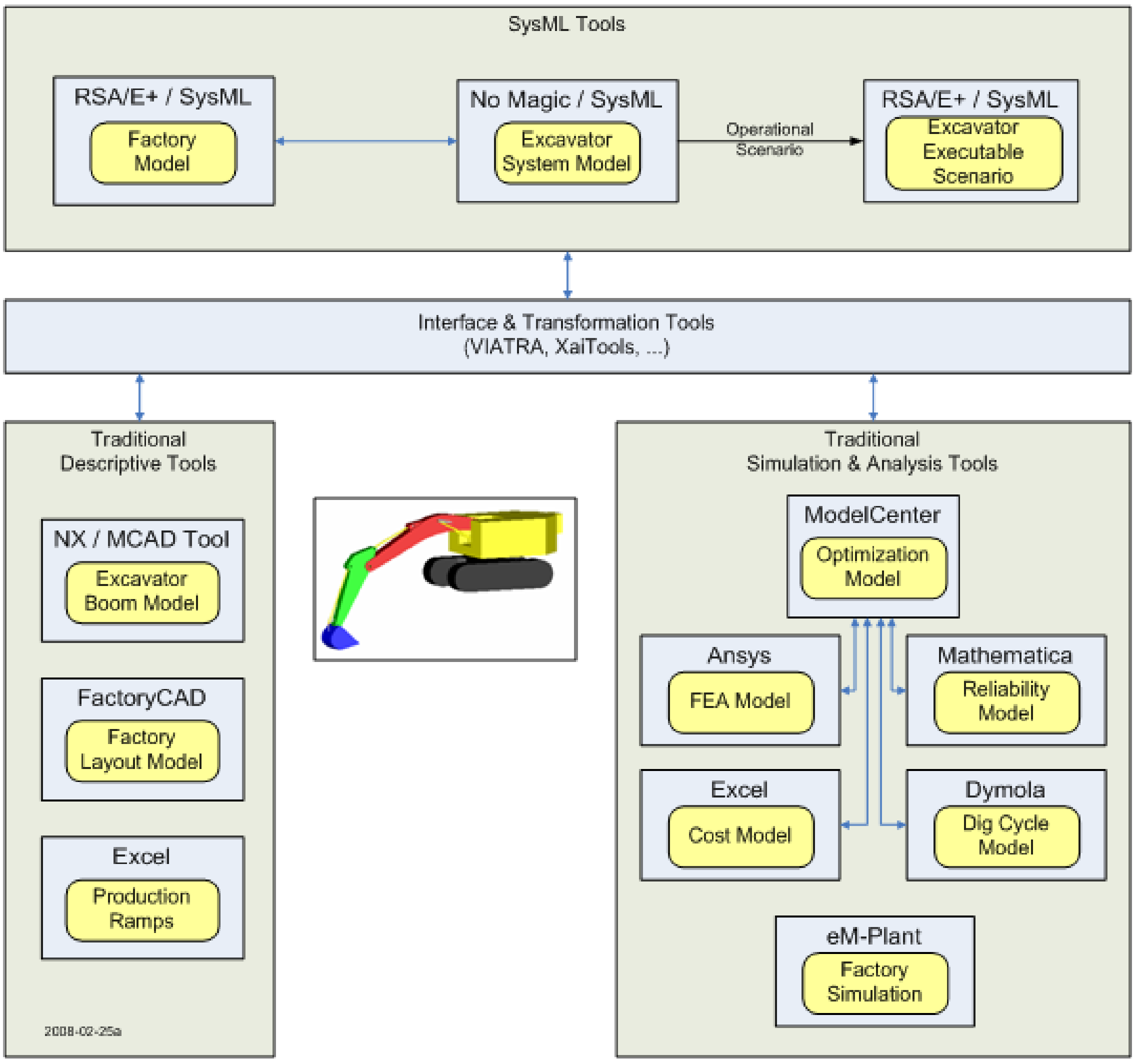


Figure 2-15: Connectivity between SysML™ tools and analysis tools [13].

2.6.5. MBSE and Knowledge Based Engineering (KBE)

Although MBSE and Knowledge Based Engineering (KBE) have different roots and aims, there are areas of overlap of these two fields of engineering. This section provides a brief overview of KBE tools and methodologies, and discusses how KBE and MBSE relate to each other.

2.6.5.1. Overview of KBE Tools and Methodologies

KBE is an approach which emphasises the capture and reuse of knowledge to reduce the time and effort to perform engineering analyses. KBE was established in the

1980s when efforts were made to apply Artificial Intelligence (AI) and Knowledge Engineering techniques in Computer Aided Design [79]. KBE has been used to support a wide array of activities in Product Lifecycle Management (PLM), Computer Aided technologies (CAx), and multi-disciplinary optimisation. Milton lists a number of KBE system and tools. Some examples are:

- Adaptive Modeling Language (AML) – An object-oriented, knowledge-based engineering modelling framework which enables multidisciplinary modelling, and integration of product and process development cycles.
- Knowledge Fusion – A KBE system integrated with the CAD system Unigraphics NX. It provides functionality to aid generation of geometry. These include the capture of engineering knowledge through expressions, formulas, and equations. Another feature is its ability to create libraries of intelligent part and feature families. A full list of its capabilities can be found in its product website [80].
- General-purpose Declarative Language (GDL) – According to its product website, GDL "is a generative application development system for creating web-centric Knowledge Based Engineering and Business applications" [81]. GDL has been used for modelling complex hierarchical systems, including three-dimensional geometric models. Its product website describes GDL as a system which combines the best aspects of a spread-sheet, a dynamic object-oriented programming language, a parametric CAD system, a web application server, and a Knowledge-management system.
- Design and Engineering Engine (DEE) – DEE is a software system developed for multidisciplinary, distributed design problems. It consists of design, modelling and analysis software tools (commercial off-the-shelf and bespoke) connected by a software communication framework. It was used in the design process of a wind turbine to support the automation of non-creative and repetitive design activities [79].

The systems and tools described above show some of the ways that KBE has been implemented.

There have also been a number of KBE methodologies to support the development of KBE applications and systems. The most well-known of these is the **M**ethodology and software tools **O**riented to **K**nowledge-Based Engineering **A**pplications (MOKA) [82]; developed around 1998. The MOKA Consortium consisted of BAE Systems, Aerospatiale Matra, BAE Systems, Daimler-Chrysler, PSA Peugeot Citroen, Knowledge Technologies International, Decan, and Coventry University. The aim of the MOKA project was the development of a methodology and tools to support the deployment of KBE applications. This project yielded the MOKA KBE life cycle as shown in Figure 2-16. Particular emphasis was given to the "Capture" and "Formalise" stages in the lifecycle. The "Capture" stage involves the collection of data in an "Informal Model". The "Informal Model" contains a set of Illustration, Constraint, Activity, Rule, and Entity (ICARE) forms. The ICARE forms help to structure the data for its implementation in formal models. The creation of formal models occurs at the "Formalise" stage of the lifecycle. "Formal Models" are generated using the MOKA Modelling Language (MML); which is adapted from the Unified Modelling Language (UML). The "Formal Models" organise the knowledge by providing meta views, such as Structure, Function, and Behaviour.

The emphasis of the MOKA methodology on the "Capture" and "Formalise" stages was driven by the assertion that KBE technology at the time was not able to handle the complex and diverse nature of engineering knowledge [83]. This was based on their observations that engineering problem solving involved people from different engineering disciplines. The resultant complexity of the knowledge and the engineering process produced a large diversity of special purpose engineering tools. Additionally, the consortium contended that information modelling approaches, such as STEP, CommonKADS, and other object-oriented were "not expressive and differentiated enough in order to fulfil the high knowledge modelling demands in engineering" [83].

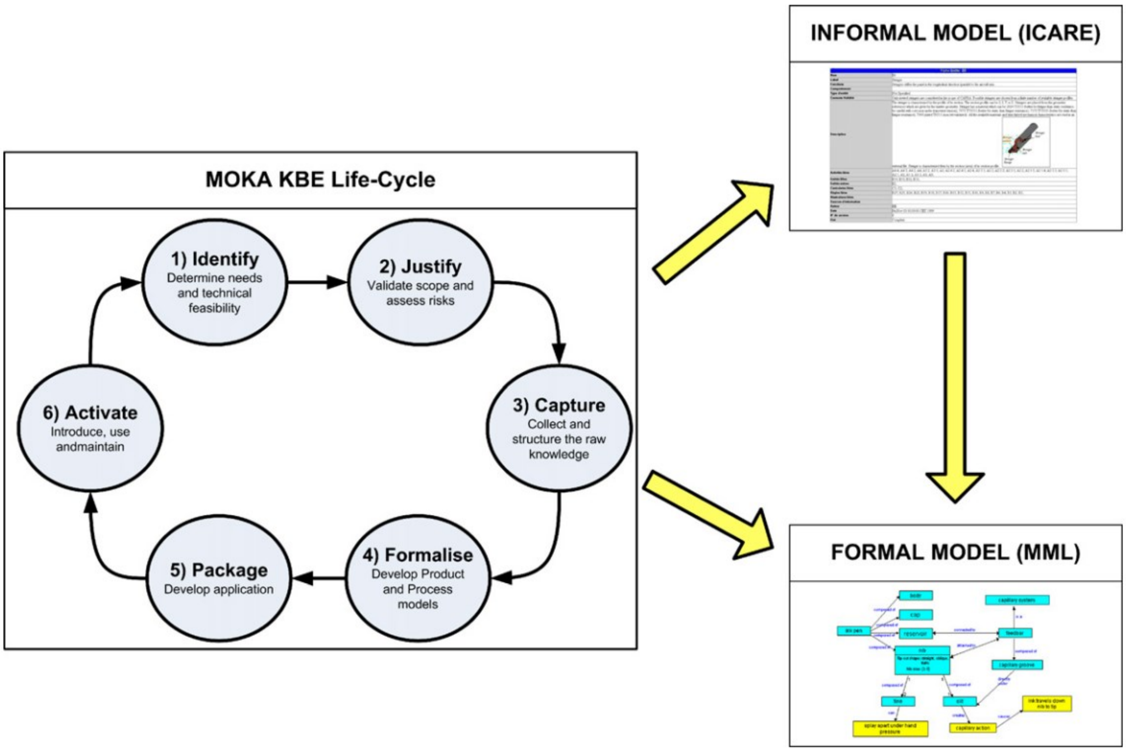


Figure 2-16: Elements of the MOKA methodology [82].

Another methodology for KBE is **K**nowledge **N**urture for **O**ptimal **M**ultidisciplinary **A**nalysis and **D**esign (KNOMAD). KNOMAD is a methodology for the analytical utilization, development and evolution of multi-disciplinary engineering knowledge within the design and production realms [84]. It was conceived to address the limitations of methodologies and tools like MOKA and DEE. The KNOMAD methodology has stages for formal knowledge capture and organisation. It uses System Engineering (SE) techniques like Functional Breakdown Structures, Functional Flow Diagrams, organograms, and N2-charts for knowledge capture and organises the data using an ontology. Elements of the DEE multidisciplinary, distributed design system are then used to implement the modelling and analysis stages of KNOMAD.

2.6.5.2. Relationship between MBSE and KBE

From the description of MOKA and KNOMAD, it is apparent that there are stages in the KBE lifecycle process where MBSE can be implemented. This is mainly in the knowledge capture and formalisation stages of the KBE methodologies. The benefits of MBSE mentioned in section 2.6 can address the issues that motivated the development of the MOKA methodology. Additionally, both MML and SysML are based on UML which means they share many of the basic diagrams and constructs. McGoey [85], in his view of the future of KBE, believes that the capture of design intent and process should be done via MML, SysML, and, Business Process Modelling Language (BPML) to enable creation of formal, executable design processes. KNOMAD uses SE techniques for knowledge capture which can also be done in SysML. A criticism of MOKA is that it does not demonstrate how the structured knowledge is used to aid or generate engineering analyses [84]. As mentioned in section 2.6.4, SysML has been demonstrated as a tool that can be used to integrate various engineering tool sets to perform multi-disciplinary analysis. This makes SysML a promising tool for KBE applications as it could potentially be used for the knowledge capture and model generation stages of the KBE lifecycle.

2.6.6. Advantages and Disadvantages of MBSE

The following summarises the main advantages and disadvantages of MBSE which need to be considered.

Advantages:

- MBSE formalises the development of complex systems through the use of models.
- Knowledge capture in visual models rather than in textual documents aids understanding and transparency.
- MBSE provides a common language which improves communication and understanding across the system stakeholders.
- MBSE assists in managing complex system development by supporting of multiple viewpoints to address the different interests of the stakeholders.

Disadvantages:

- MBSE requires a cultural change of an organisation as it requires the stakeholders to be trained in the tools and language.
- MBSE languages have their roots in software development. SysML, for example, can be unwieldy when modelling systems where domain specific modelling languages already exist, such as hydraulic or electronic systems.

- MBSE is not yet widely used, although there have been interest from NASA, ESA, Boeing and Lockheed Martin [86].
- There is an absence of convergent MBSE standards which impedes its adoption and imposes unique training requirements for each tool and method. This is evident in the various MBSE methodologies surveyed by Estefan [73].

Some of these disadvantages are due to the lack of maturity in the MBSE methodology. These have been identified by the MBSE practicing community and effort has been dedicated to allaying them, as discussed in 2.6.4.

2.7. Summary

This chapter provides an overview of the relevant concepts and methodologies used in engineering cost modelling. It was further established that the design stage offers the most scope for reducing costs. It is therefore essential that designers have tools which will enable them to predict cost at all phases of design. As this thesis aims to develop methodologies to design for whole life cost, there was a particular focus on LCC concepts, processes and models. Various issues with LCC modelling were identified from the reviewed literature. From these issues, several desirable characteristics of LCC models were identified. These were that LCC models should be customisable, transparent and modular. This chapter also introduces the concept of MBSE so as to assess how it can aid the LCC modelling process. Finally, a discussion on how MBSE can help to address the LCC modelling issues identified earlier was presented. As MBSE is still a maturing technology, future developments which will benefit LCC modelling were identified.

3. Life Cycle Costing of an Aero-Engine

In this chapter, the development of a methodology to assess design decisions on the LCC of aero-engine turbine blades is presented. An integrated model is created incorporating the desired characteristics identified in chapter 2. It integrates manufacturing cost, deterioration mechanism based life prediction and maintenance cost. The study also serves to illustrate the process and needs of a LCC analysis on an aero-engine.

3.1. Introduction

This case study looks at how the design of a turbine blade can affect aero-engine LCC. Turbine blades are responsible for extracting energy from the high temperature, high pressure gas produced by the combustor. The design of turbine blades is complex as it involves different design disciplines such as stress, thermal, and mechanical analysis, and aerodynamics. All these aspects of the component can affect LCC in one way or another. This study focuses on how geometric changes to a component can change LCC.

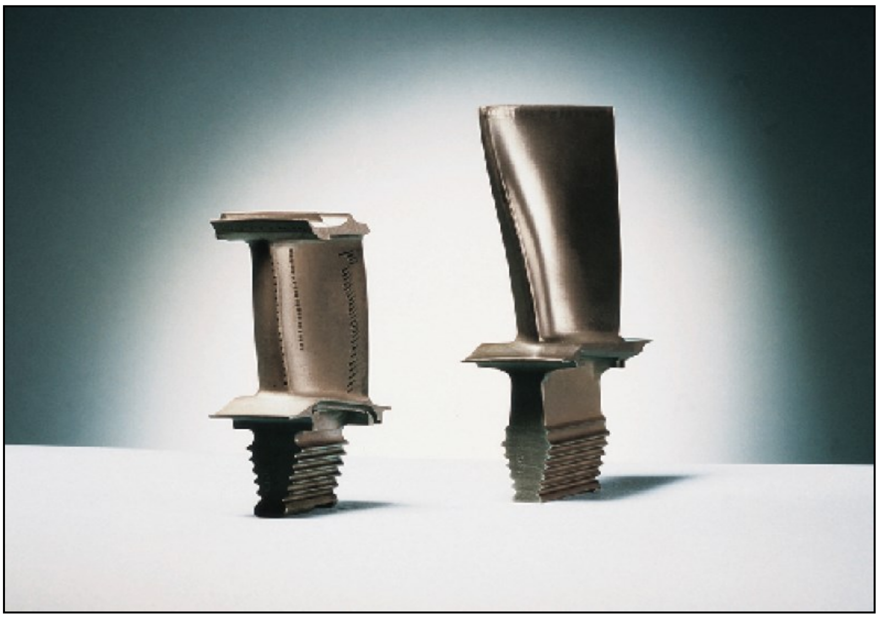


Figure 3-1: BR715 turbine blades [3].

From the literature review, it was seen that LCC can be segregated into acquisition and ownership costs. These costs can be further broken down into a Cost Breakdown Structure (CBS) as was seen in Figure 2-8. It was also mentioned in section 2.5.5.1 that for LCC modelling it is recommended that each study is tailored to its objectives. Barringer [50] advocates that the calculation of LCC should only include cost elements where changes in cost will be incurred. To keep the scope of the study manageable, it is assumed that the only cost element affected by the design changes will be maintenance cost. Maintenance cost is a critical component in the calculation of operating costs. Figure 3-2 shows that direct maintenance costs make up 6-8% of the total operating costs of low cost and major carrier airlines [1]. While this is a relatively low figure, engine maintenance plays a significant factor in other important operator cost drivers. The following are the cost drivers mentioned by Harrison [1]:

1. Range and Payload – This is vital for an airline to achieve their target route structure. Proper maintenance is required to ensure that the engine continues to deliver the thrust required throughout its operational life.
2. Safety – Maintenance is vital to the safe performance of an engine. Flight mishaps attributable to engine failure will have dire consequences.
3. Schedule Reliability – This is a major driver of consequential operating costs and it influences revenue opportunity. Some operators value the cost of an overnight flight delay at about $1 million, non-overnight flight delays could be penalised at a rate of $50,000/hr.
4. Life Cycle Fuel Burn – Fuel costs can be 2 to 4 times more than maintenance costs. Proper engine maintenance can increase fuel burn efficiency and control engine deterioration.
5. Engine Overhaul – These costs are the minimum investment costs necessary to ensure the operation of the engine.

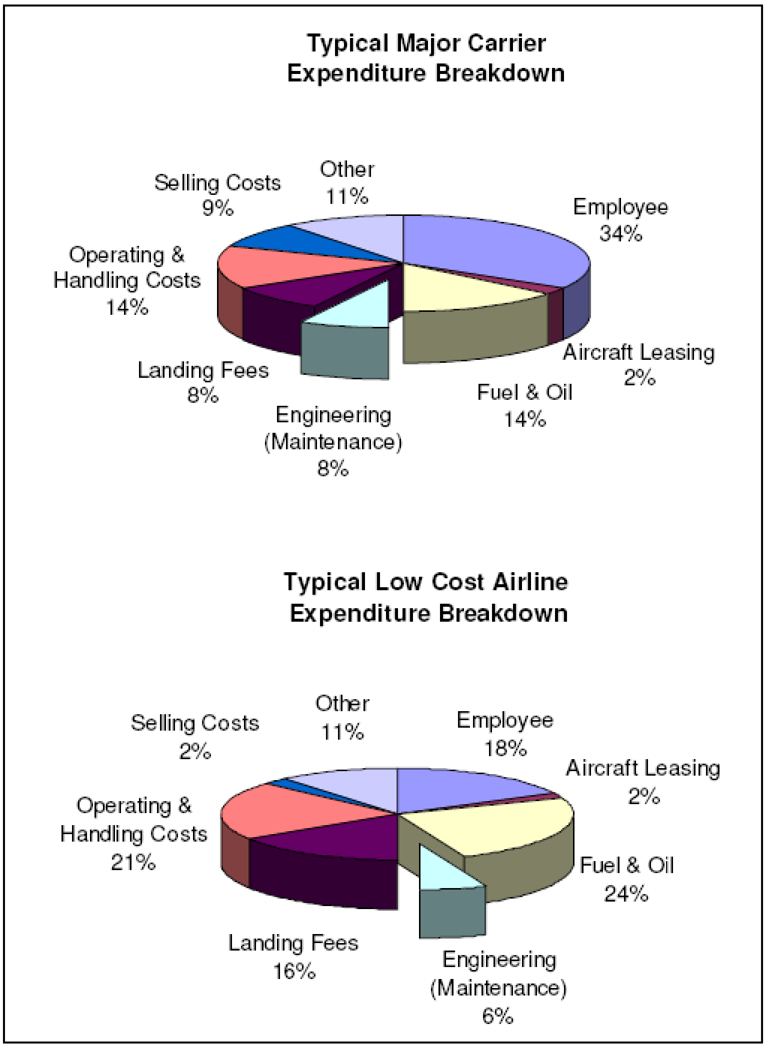


Figure 3-2: Operating costs for low cost and major carrier airlines [1].

The following sections detail the integrated LCC model that was constructed to perform the study as well as the individual component models used.

3.2. Integrated Life Cycle Cost Model

To link LCC to the design of a turbine blade, relationships between the design parameters and LCC must be identified. These relationships are illustrated in Figure 3-3; which also show the different models required for the calculation.

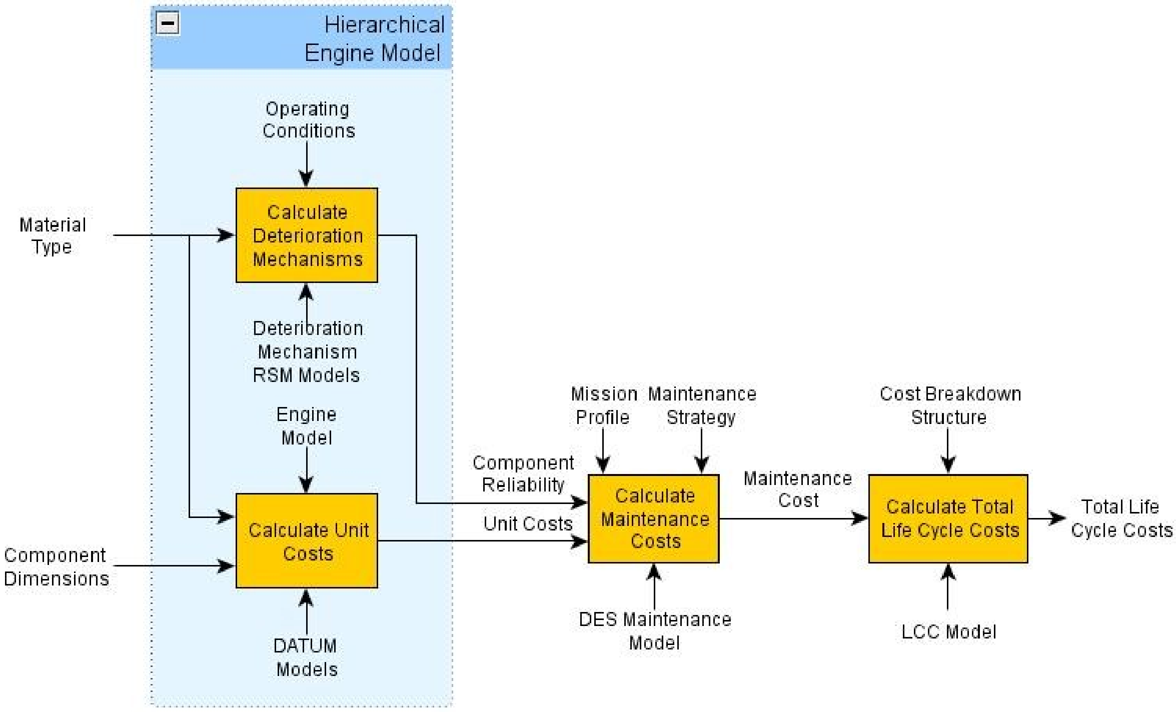


Figure 3-3: Interactions and data flows within the integrated model.

The unit cost and deterioration mechanism life prediction models were constructed using Vanguard Studio. The maintenance model uses a discrete-event simulation software tool called ExtendSim®. These models were integrated using Isight, a commercial software integration package [87]; a screen shot of the integrated model is shown in Figure 3-4. The box drawn around the deterioration mechanism life model component in the workflow figure indicates that it was developed externally, as described in 3.3.4.

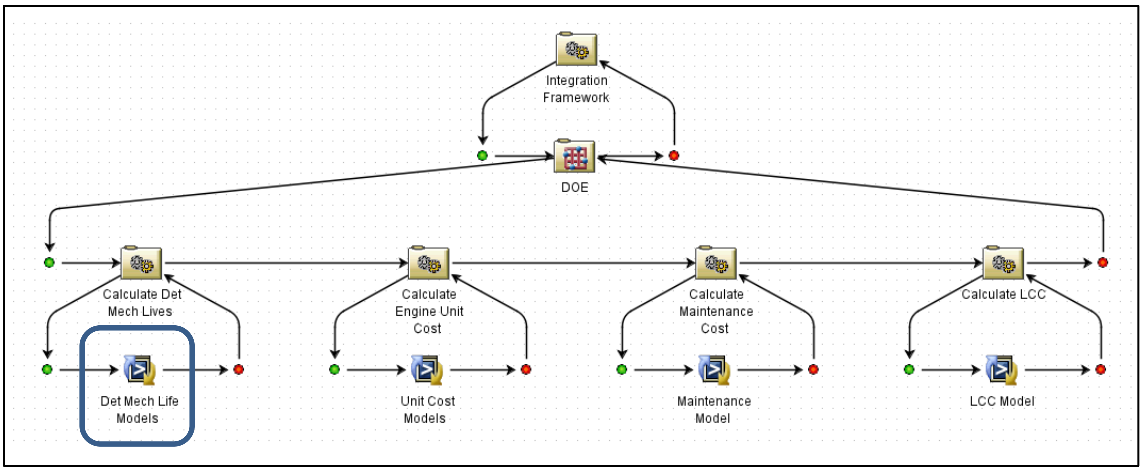


Figure 3-4: Implementing integrated model in Isight.

To generate a work flow, the user has to drag and drop various activity blocks provided by the software. To insert a model into the workflow, a "Simcode" block is inserted which runs a command to execute the desired model with input and output data exchange. Once the work flow has been generated, the input and output data flows between the various models can be mapped; this operation is shown in Figure 3-5.

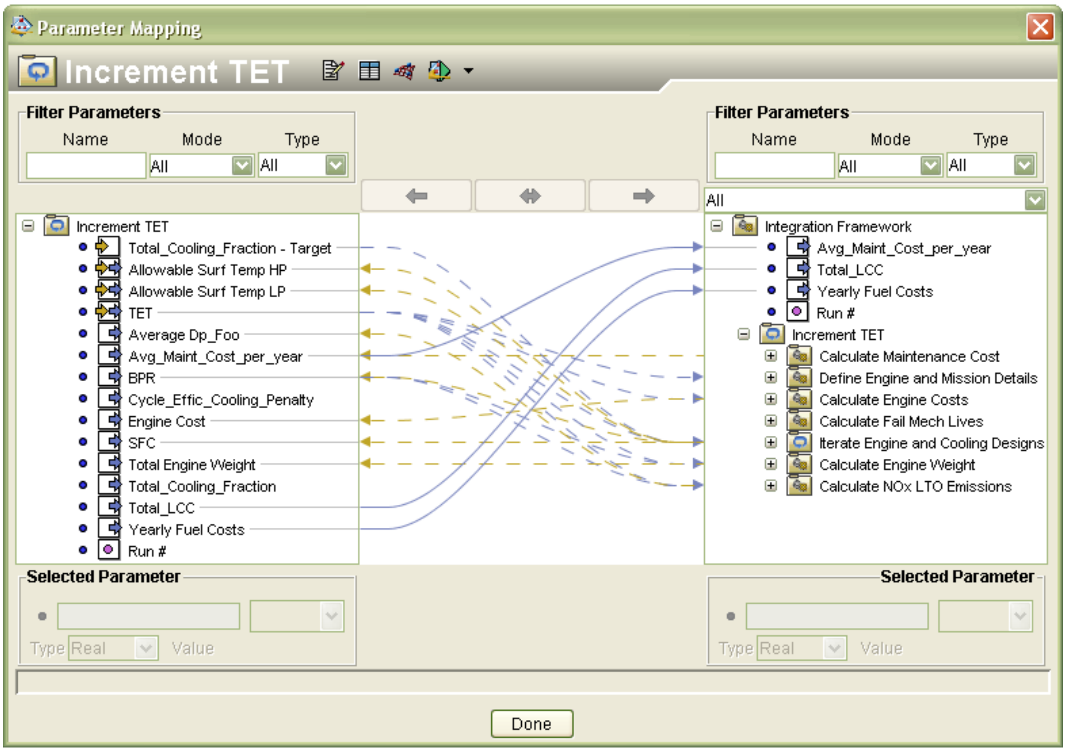


Figure 3-5: Parameter mapping in Isight.

3.3. Hierarchical Models

To ensure that the developed model remains flexible and extensible, a hierarchical approach was adopted in defining the characteristics of the aero-engine. A hierarchical tree is an appropriate representation for an aero-engine given its modular design. Figure 3-6 shows an exploded-view diagram of an aero-engine. An aero-engine typically comprises various sub-assembly systems called modules. Examples of modules are the fan, intermediate pressure (IP) compressor, high pressure (HP) compressor, combustor, IP turbine and HP turbine. Each of these modules consists of different components such as discs, stators, nozzle guide vanes and blades. Figure 3-7 shows a schematic of how the hierarchical model is organised.

In addition to defining the physical structure of the aero-engine, the hierarchical model contains information regarding component cost and reliability.

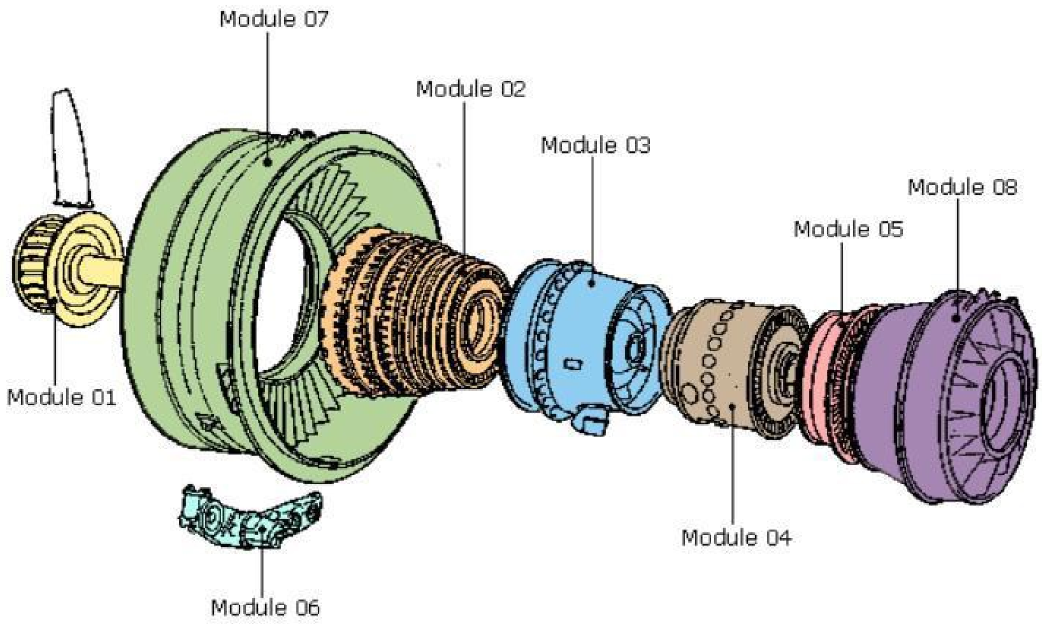


Figure 3-6 Breakdown of an aero-engine [88].

The developed hierarchical models follow the philosophies established in the Design Analysis Tool for Unit-Cost Modelling (DATUM) project [47]. The aim of DATUM was to establish a costing capability to support design decision making for all phases of the product development process. One of the most important aspects of DATUM concerned how knowledge is captured, represented and deployed. This resulted in the innovative use of a generic financial modelling tool, Vanguard Studio (known as DecisionPro in the DATUM project), to solve engineering cost estimating problems. Vanguard Studio was found to be highly suitable for this application because of the way it modelled hierarchical structures.

This structure encouraged users to decompose a problem into a logical series of steps. Additionally, the software was found to be:

- Easy to use;
- Able to perform stochastic and analytical calculations;
- Deployable through standard web browsers;
- Able to store models in a neutral, text based format; which facilitates integration and "future-proofing".

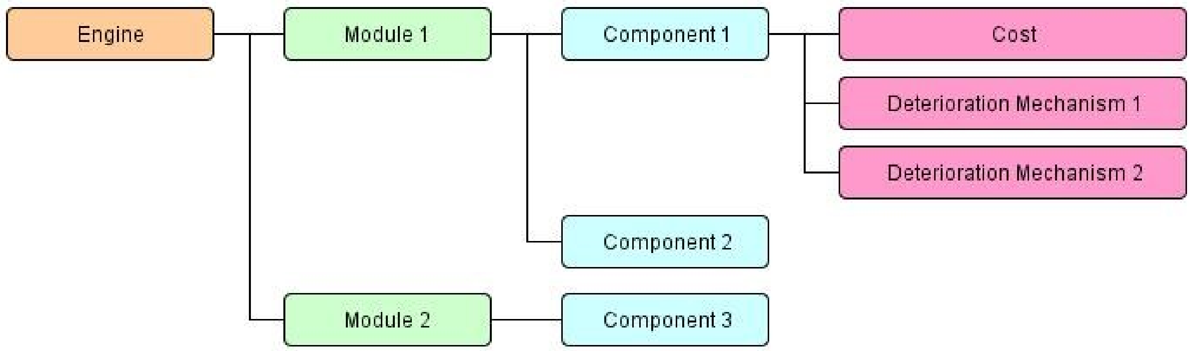


Figure 3-7: Schematic of hierarchical model.

A particularly important enhancement developed in Vanguard Studio concerned the use of the sophisticated library function which would provide an object-oriented capability to allow classes and instances to be defined. The particular principles of the object oriented approach used were those of encapsulation, abstraction and inheritance. With regards to encapsulation, each model has its own private and public methods and variables. This allows data protection and implementation protection. In terms of abstraction, each model contains essential characteristics which distinguish it from all other kinds of objects and thus provide defined conceptual boundaries. Finally, the principle of inheritance was used to ensure that models were consistent with each other. Each library model has a base class that maintains its own data and methods. Models derived from this base class are able to access the public methods and variables of the base class. In the context of this application, object oriented models impart the following benefits:

- Re-use of data, equations and logic;
- Simplified maintenance;
- Consistency of model structure and standards.

As models can be embedded as objects within another model, they provide a natural metaphor for the various components in a gas turbine engine. The product structure of the gas turbine can hence be created using these models. Figure 3-8 shows how the hierarchical models are linked together to represent the structure of the gas turbine.

Several libraries of models were developed using these philosophies. The libraries of models include engines, modules, components, manufacturing cost, deterioration mechanisms and repairs. Details of these models are described in the sections below.

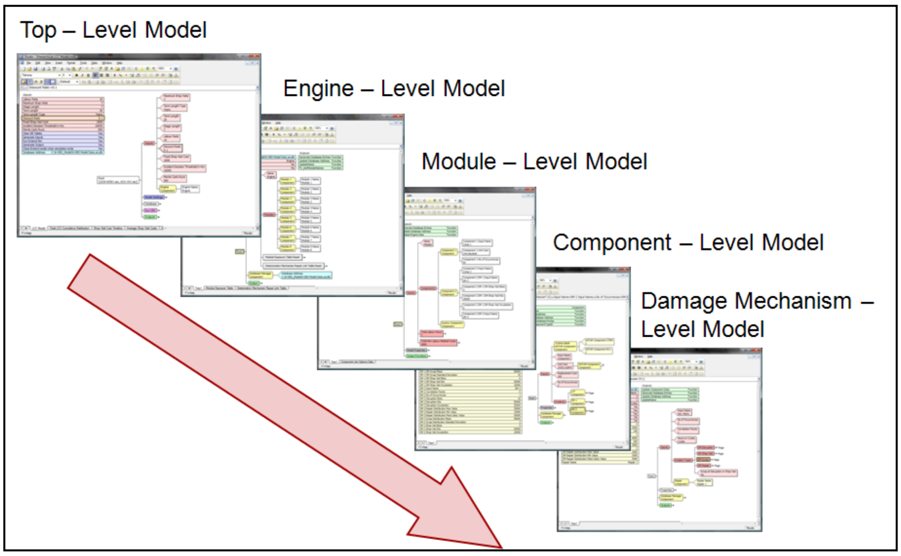


Figure 3-8: Linkage of Vanguard Studio models.

3.3.1. Engines

The primary function of this model is to define how the engine is structured. Along with the required engine characteristics (engine type, assembly rules and costs), this model contains objects representing the various modules within a gas turbine engine. Figure 3-9 shows the engine model.

3.3.2. Modules

The module level models are a level of abstraction lower than the engine model. They describe the sub-systems within the engine, such as the compressors, turbines and fan. These sub-systems contain objects representing the components which make up a module. For example, the high pressure (HP) module contains the HP compressor blades, stators, HP turbine blades, discs, nozzle guide vanes and combustors. Figure 3-10 shows the module model.

3.3.3. Components

At this level, each model represents a component in the engine. These include fan, compressor and turbine blades, stators, nozzle guide vanes, shafts, discs and fan casing. Within each of these models, there are objects which calculate the manufacturing cost and deterioration mechanism lives of the component. Figure 3-11 shows the component model.

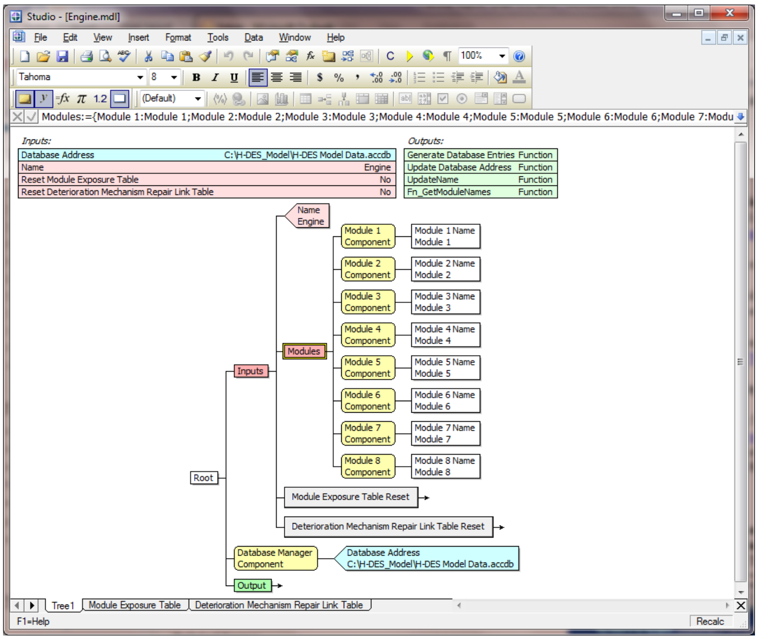


Figure 3-9: Engine model.

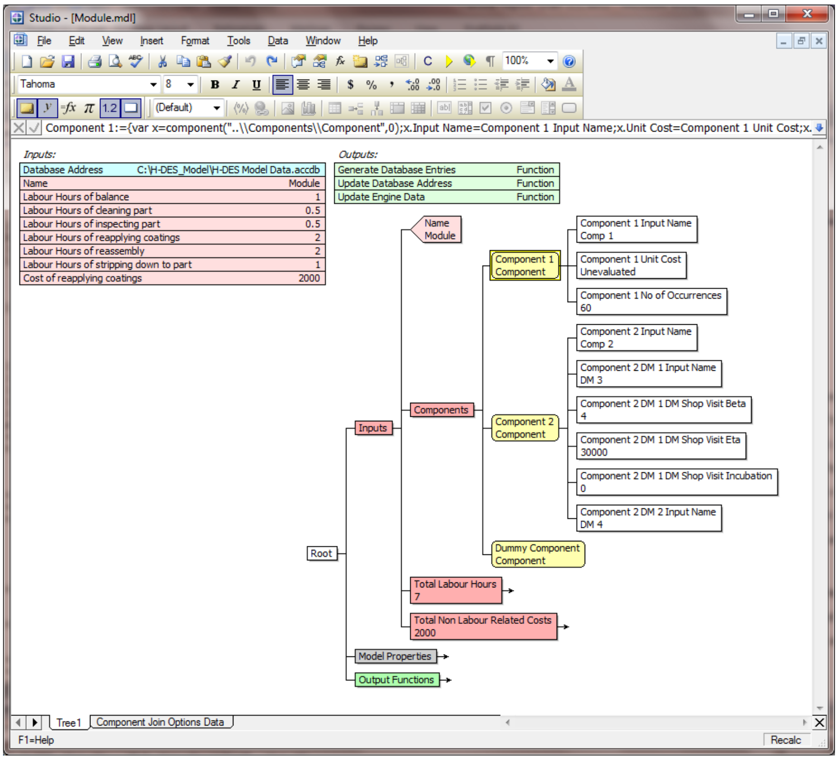


Figure 3-10: Module model.

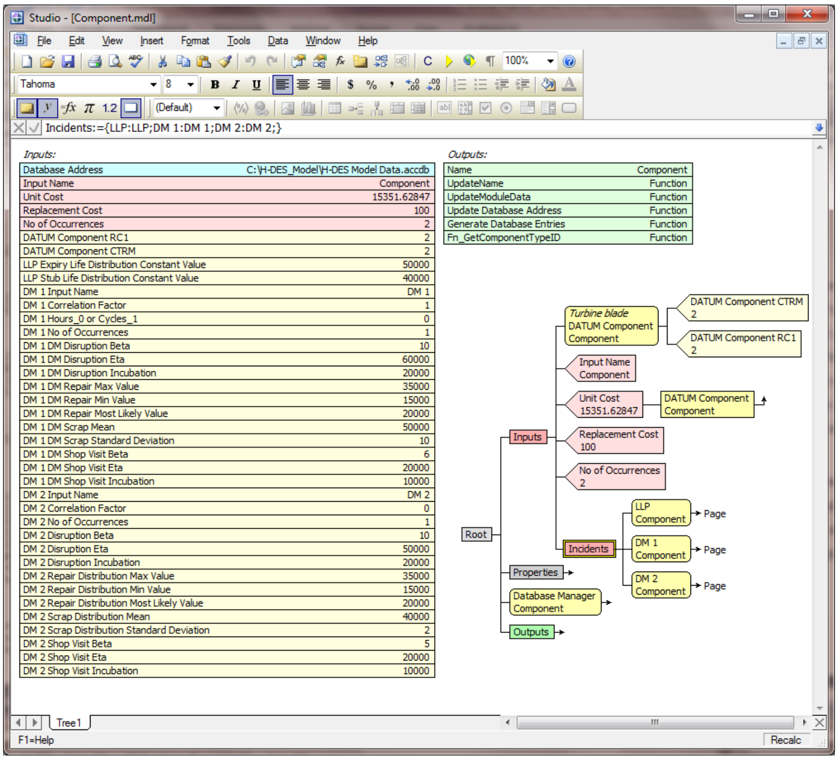


Figure 3-11: Component model.

3.3.4. Deterioration Mechanisms

Two common deterioration mechanisms associated with gas turbine blades are fatigue and creep. High cycle fatigue (HCF) has been observed to be the single largest cause of turbine blade failure [89, 90]. In general, fatigue is the progressive and localized structural damage that occurs when a material is subjected to cyclic loadings. HCF failures occur for cases with more than 10^4 cycles and where deformation is primarily elastic. Creep on the other hand is the tendency of a solid material to slowly deform permanently under the influence of stresses. It occurs as a result of long term exposure to levels of stress that are below the yield strength of the material. These deterioration mechanisms life prediction models developed for the Integrated Product and Services (IPAS) project [91] were integrated into the system model. They used response surface models (RSMs) generated from experimental rig data [92]. Figure 3-12 shows the RSM developed in the Vanguard Studio tool.

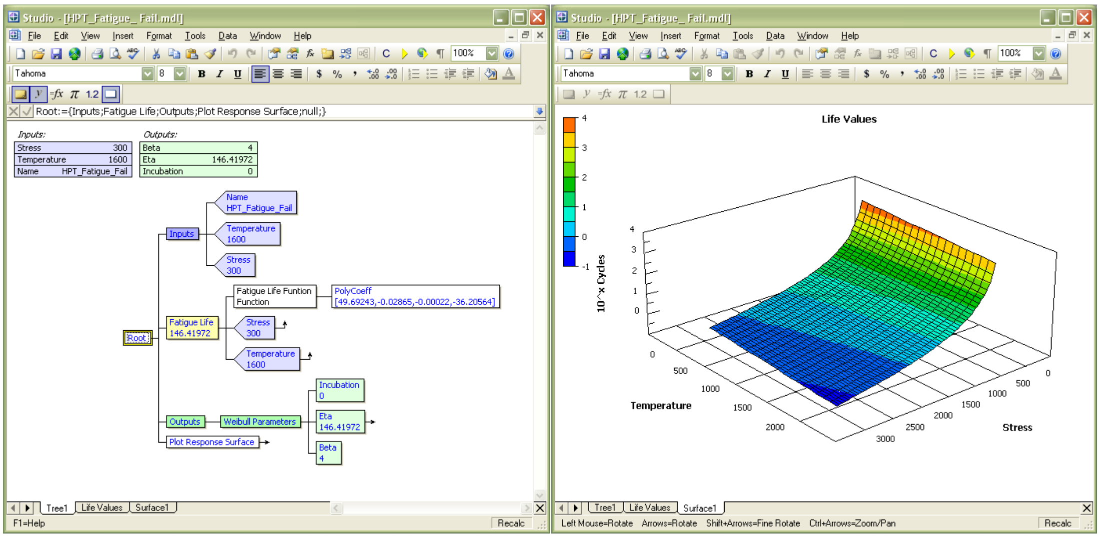


Figure 3-12: Physics based response surface model.

3.3.5. Unit cost

These cost models were developed using the methodology established in the DATUM [47] project. The manufacturing cost of gas turbine components was estimated using a feature-based approach. This required the mapping of manufacturing processes to the appropriate features of the component. The cost of a manufacturing feature is the cost of resources expended in making the transition from state $n-1$ to n as shown in Figure 3-13. This state change was associated with change in geometry (milling) or achieving a product characteristic (heat treatment) that provided a certain function in the component [9]. The final manufacturing cost was the aggregation of all the manufacturing feature costs. Figure 3-14 shows a HPT blade cost model, with the values removed due to confidentiality.

3.3.6. Repairs

These models contain information of repair tasks that are performed according to the type of deterioration mechanism suffered by each component. For example, the operations required of a component suffering from erosion would be material addition and polishing. Information in this model typically includes the repair cost, time, yield (percentage of repairs are successful), limit (how many times can a repair be carried out) and quality (percentage of total life is regained by repair). A screenshot of the model is shown in Figure 3-15.

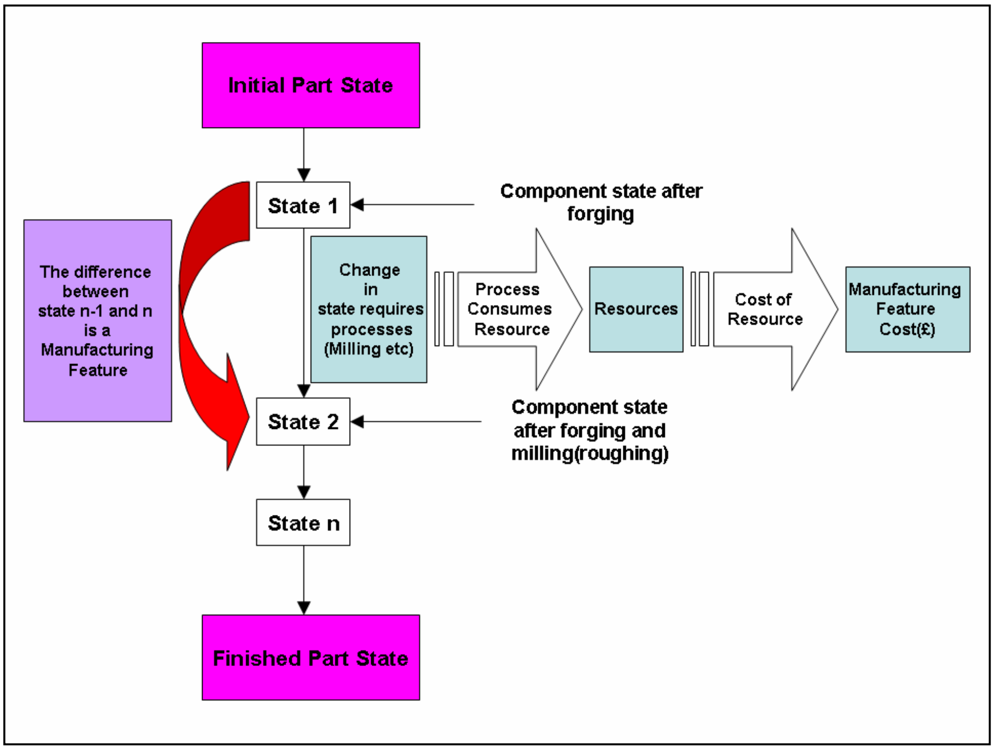


Figure 3-13: State transition and manufacturing feature costs [9].

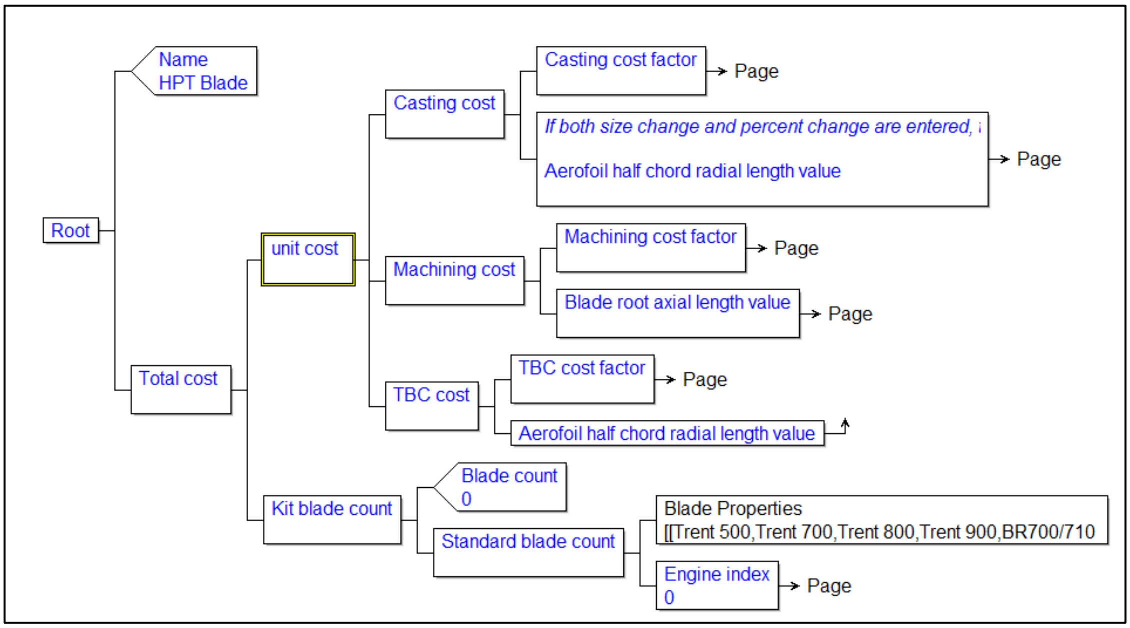


Figure 3-14: Screenshot of HPT blade cost model.

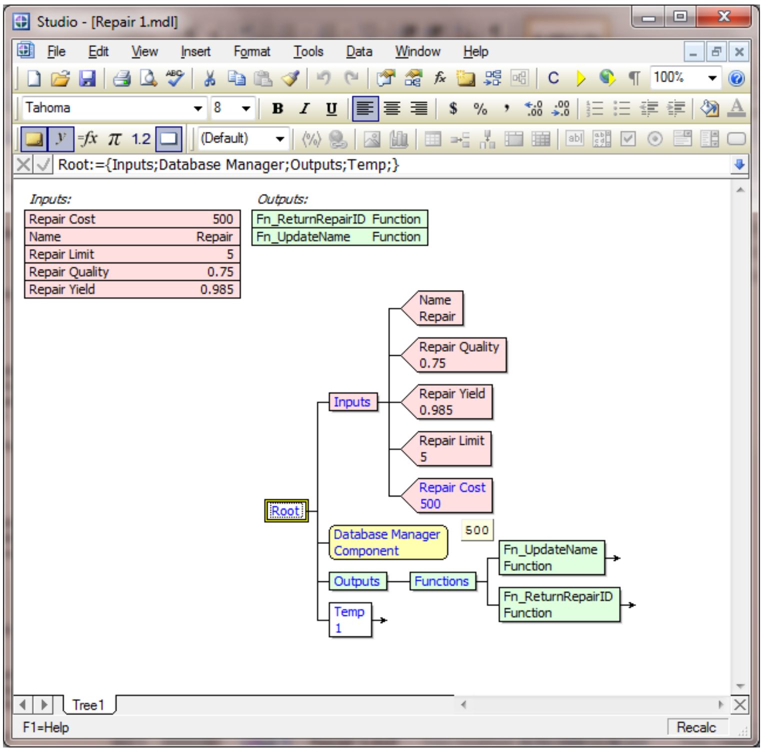


Figure 3-15: Repair model.

3.4. Aero-Engine Maintenance Cost

In general, total maintenance cost is the aggregation of cost elements such as repairs, component replacement, tooling, inventory and labour. In the case of aero-engines, the calculation of maintenance cost is a complex task and is not simply a matter of summing static estimations of successive annual cost and projecting them to their present values [93]. Neither is it a reactive task of responding to equipment failures and scheduling routine preventive maintenance [94]. Engine maintenance has inherent dynamics which affect its performance over time. It is a function of engine reliability and the selected maintenance strategy [95]. The cost of maintenance is typically non-linear and accumulates at different rates over the life of the engine. Figure 3-16 [96] shows a schematic of an aero-engine maintenance system. It shows the key features of the system; such as Mission Definition, Events, States, Resources and Rules.

The mission data of the engine includes information such as the number of flight cycles that the engine has to perform in a month and the number of hours in each flight cycle. This data will determine when the engine has to be inspected, repaired and overhauled. These actions are represented in the maintenance model as events. The reliability characteristics of the modelled engine control the different states of deterioration that the engine will progress through. Based on the state of deterioration and the maintenance rules applied, various maintenance actions are performed. These actions will consume resources which can then be linked to overall cost calculations.

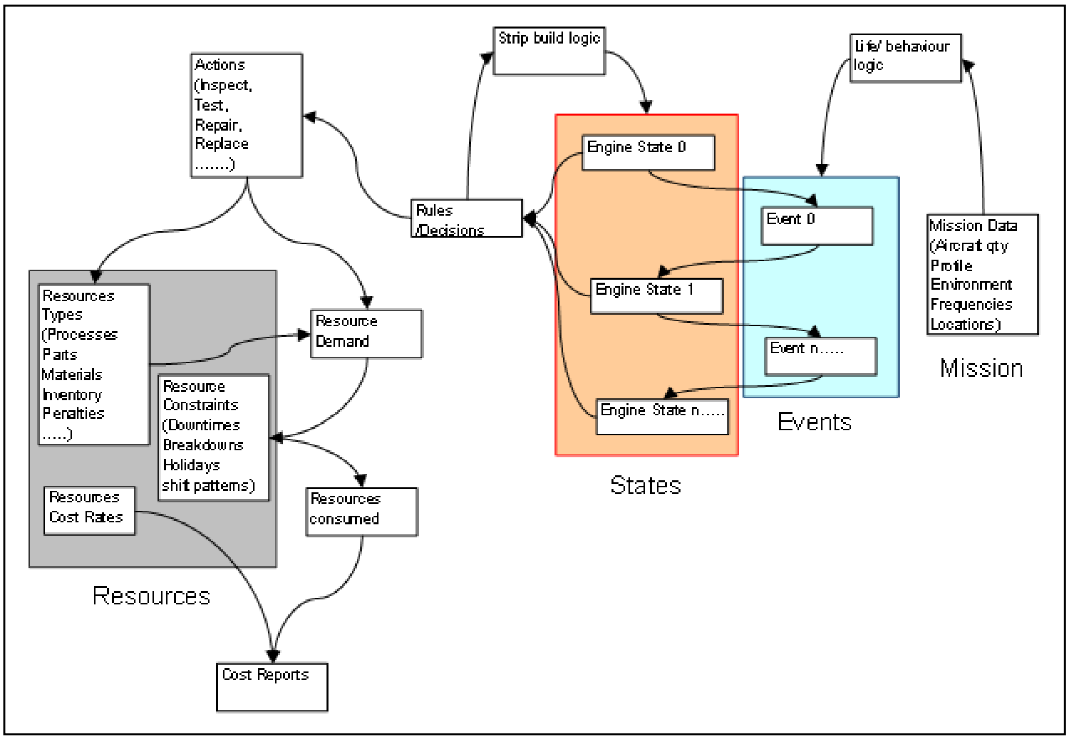


Figure 3-16: Engine maintenance system model [96].

3.4.1. Reliability

The reliability of the system being modelled is central to the calculation of maintenance costs. Reliability, in engineering terms, is defined as the probability that a system or component will perform its required functions under stated conditions for a specified period of time [97]. Mathematically, this may be expressed as,

$$R(t) = Pr\{t_{failure} > t\} = \int_t^{\infty} f(x)dx, \qquad (3.1)$$

where $f(x)$ is the failure probability density function, $t_{failure}$ is the time to failure and t is time.

Equation 3.1 is known as the reliability function [98]. Reliability of a system has to be described in probabilistic terms because system failure times are not deterministic by nature. Quantifying the reliability of a system hence requires the use of statistics and probability theory.

The field of Reliability Engineering deals with the study of system reliability. Some of the statistical distributions used for reliability measures are Exponential, Normal, Log Normal, Weibull, and Gamma distributions. In reliability engineering, the Weibull distribution is one of the most widely used as it is versatile enough to take on the characteristics of other types of distributions [99]. The Weibull probability density function is given by:

$$f(t) = \frac{\beta}{\lambda}\left(\frac{t-\theta}{\lambda}\right)^{\beta-1} e^{-\left(\frac{t-\theta}{\lambda}\right)^{\beta}}, \qquad (3.2)$$

For:

$f(t) \geq 0, t \geq 0 \; or \; \theta;$

$\beta > 0;$

$\lambda > 0;$

$-\infty < \theta < +\infty.$

Where:

$\lambda =$ scale parameter, or characteristic life;

$\beta =$ shape parameter;

$\theta =$ location parameter.

By changing the shape parameter, β, the Weibull distribution interpolates between the exponential distribution (β = 1) and the Rayleigh distribution (β = 2); shown in Figure 3-17.

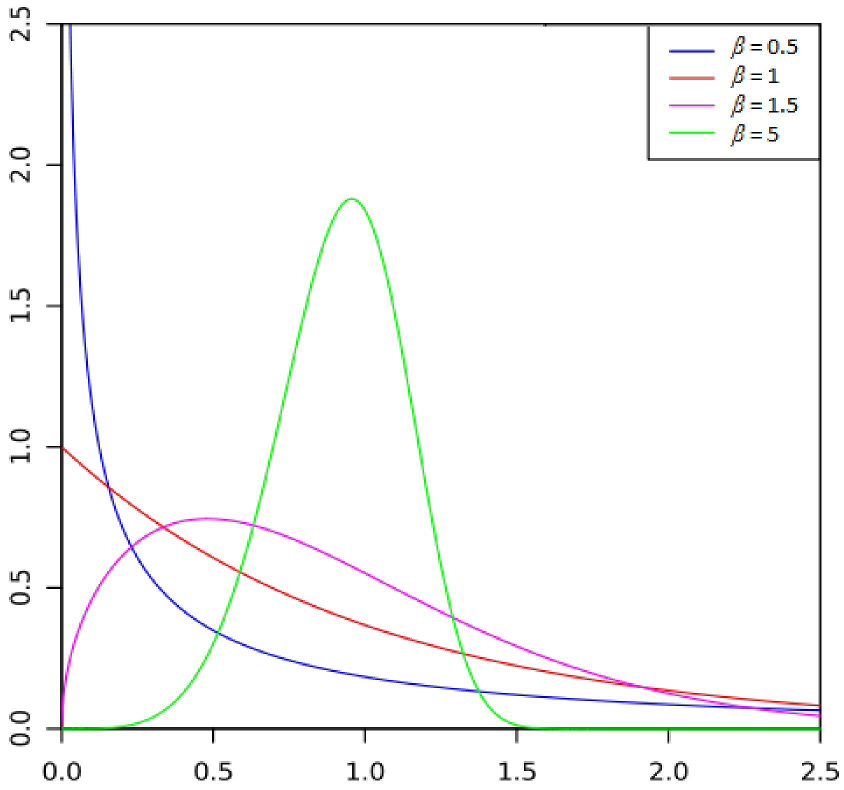


Figure 3-17: Weibull probability density function.

If t is taken to be the time of failure, a Weibull distribution with $\beta < 1$ shows a failure rate which decreases over time. This is used to represent "infant mortality" with products/systems failing early and then the failure rate decreases over time.

When $\beta = 1$, the failure rate is constant over time. This represents products/systems failing randomly. When $\beta > 1$, the failure rate increases with time. This represents products/systems going through an aging process where they fail after a period of time.

3.4.2. Strategy

Schedule reliability, as mentioned above, is an important consideration in aero-engine maintenance. Failure to meet schedule reliability results in considerable financial penalties. Maintenance strategies need to be centred on maintaining schedule reliability by ensuring a window of engine operation. Different metrics such as Mean Time Between Removal (MTBR) [95] and Maintenance Free Operating Period (MFOP) [94] have been used to measure this window of engine operation. Maintaining the window of engine operation is determined by [100]:

1. The ability of the engine to avoid failure.
2. The anticipation of failure and taking appropriate actions during the maintenance period.
3. The delaying of maintenance, following failure, to a subsequent maintenance period.

Figure 3-18 illustrates how these actions affect the window of engine operation.

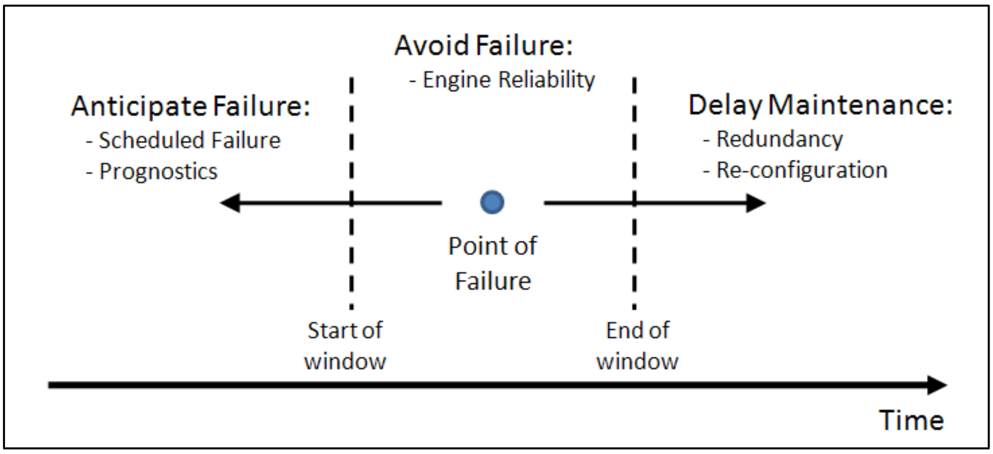


Figure 3-18: Moving potential failures outside a window of MFOP [100].

3.5. Discrete Event Simulation Maintenance Model

As mentioned earlier, the calculation of maintenance cost needs to consider system reliability and maintenance strategies. Simulation is suitable for this kind of problem as it is able to handle the required statistical and logical demands. Many examples of maintenance cost simulations have been found in applications such as manufacturing, aerospace and building design [53, 94, 95, 100-108]. The reasons commonly cited for the use of simulation are that: simulation models can handle more complex requirements than analytical models, simulation allows for experimentation with various system designs and that simulation provides better understanding of the consequences of design decisions. Some of the papers mentioned above deal specifically with aero-engine maintenance. However, what these models fail to do is to incorporate the wider life cycle perspective and how it can be used to make decisions and trade-offs at the early design stage. In some LCC models, maintenance cost has been estimated using parametric equations. As mentioned in the literature review, these models are sensitive to the range of use and they hide relationships between cost drivers and cost. A more generative approach, simulation in this case, would be beneficial as it would be applicable to different design scenarios.

3.5.1. Tool Selection

A special-purpose simulation software, ExtendSim® [109], was selected for the maintenance model. The reasons for the use of ExtendSim® are as follows [110];

1. It has an intuitive and comprehensive user interface.
2. It has the flexibility to enable the tool to be modified for experimentation in an academic research environment.
3. Simulations are sufficiently fast and efficient and will allow large and complex models.
4. It is affordable and well-supported by the tool's supplier.
5. There is a very large international user base which stimulates frequent enhancements and releases.

Additionally, a review of five discrete event software packages by Tewoldeberhan [111] identified ExtendSim® to be a good cost effective tool for developing DES models. The graphical interface in ExtendSim® allows complex processes to be displayed and animated graphically. This visual representation of the modelled process enhances model understanding and validation.

Figure 3-19 illustrates this by showing how a process diagram can be replicated. The extend model can be made to be consistent with the original flow diagram for clarity. In ExtendSim®, models are constructed using the software library building blocks which eliminates or dramatically reduces the amount of programming required. Furthermore, ExtendSim® has the capability to create hierarchical blocks. Hierarchical blocks help manage the complexity of a model by grouping sections of a model into a single block. Hierarchical blocks can also be saved into a library so that they can be easily reused. In the maintenance cost model developed, 94 standard library blocks were used and 22 hierarchical blocks were created.

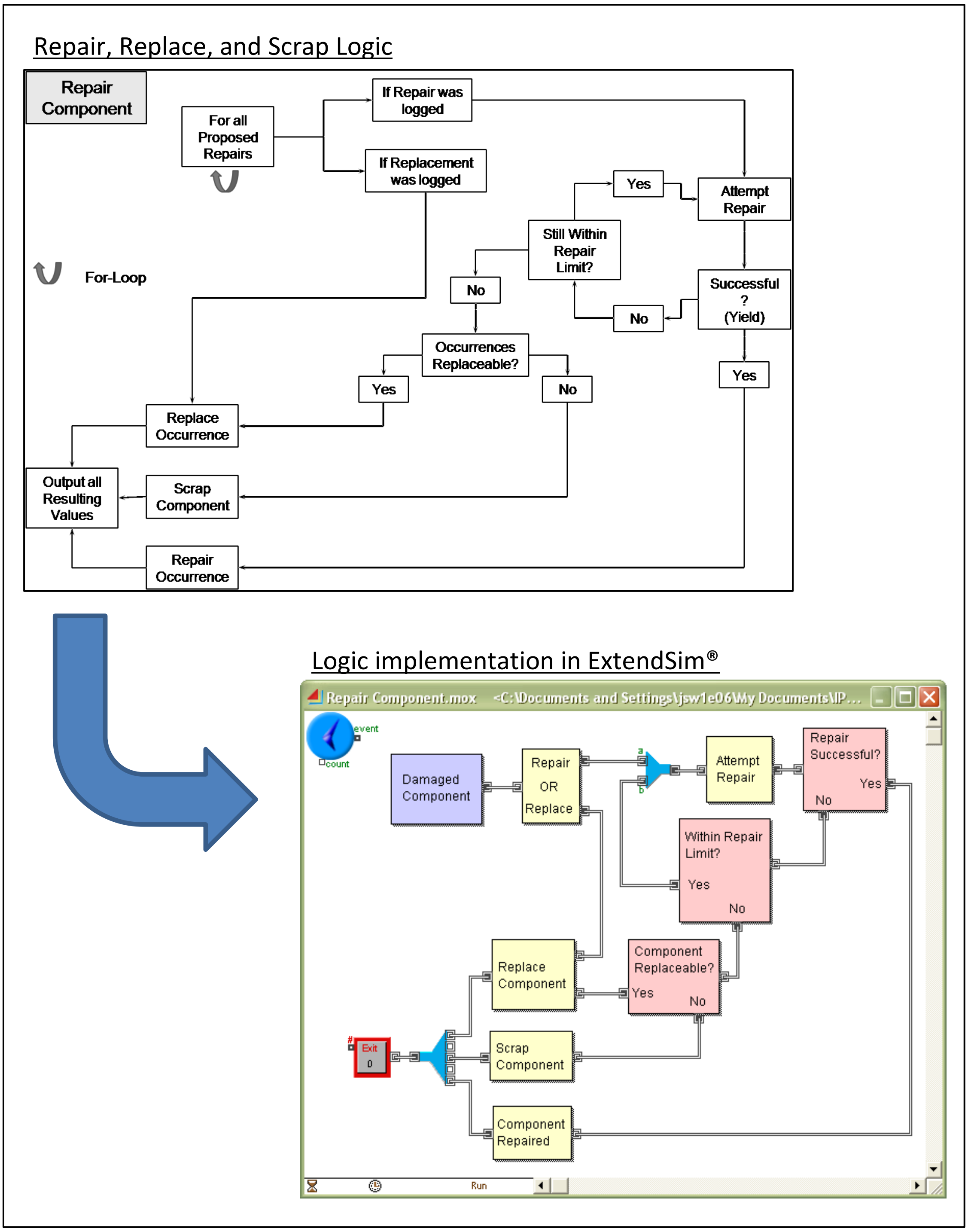


Figure 3-19: Process implementation in ExtendSim®.

One of the drawbacks with previous tools for this kind of application is the representation of the underlying logic behind the cost calculations. In systems such as aero-engine maintenance, the logical rules associated with maintenance policies contribute significantly to the behaviour of the model. Critical logic statements and decision points should be made visible so that the user can study how rule changes affect the final cost calculations. In scripting environments and spread sheet based programs, this logic is hidden within lines of code. The result is that the program becomes difficult to understand, maintain, debug and modify. These programs lack transparency and it is practically impossible to trace if there are any errors in the logic or if the model actually does what it is supposed to do.

3.5.2. Maintenance Process

Figure 3-20 shows the top most process loop in the ExtendSim® model. The basic operation of the simulation is as follows; Firstly, in the “Setup Engine” block, an “Engine” object is created from the engine definition data which is input into the model from the Vanguard Studio hierarchical models. The “Engine” object then progresses to the “Find Shop Visit Cause” block where the event which brings the engine into the overhaul base is found. The simulated model time advances to the point in time when the shop event occurs. The “Engine” object then moves into “Perform Maintenance” block where the necessary operations are performed on the engine, depending on the “state” of the engine and its components. The “state” of the engine is directly related to the expiration of deterioration mechanisms lives associated with each engine component. When the engine object has been returned to an operational “state”, the engine object moves to the “Check for End of Simulation block” where the model checks if the simulation should continue. The simulation ends when the desired simulated time is reached. The following sections describe the blocks in more detail.

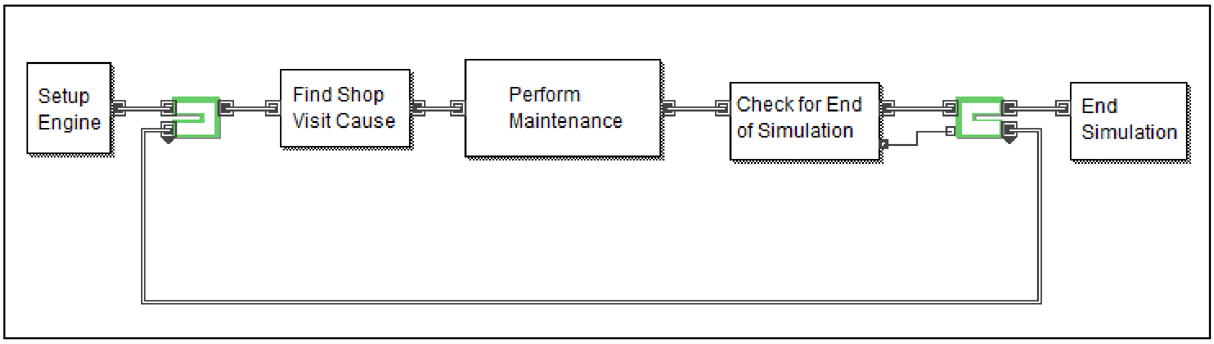


Figure 3-20: Top level process loop implemented in ExtendSim®.

3.5.2.1. "Setup Engine" Block

In this block, objects representing the components in the engine are created according to product definition inputs. The "Setup Engine" block is what is known in the ExtendSim® software as a hierarchical block. This is a way of grouping a number of blocks together to neaten the lay-out of the model. Figure 3-21 shows the contents of the "Setup Engine Block".

The "Setup Engine" block consists of a number of building blocks from the libraries of the ExtendSim® software. Block 1 in Figure 3-21 creates the simulation objects which represent each engine component. The number and type of components generated are controlled by the user inputs; this data link is shown between blocks 1 and 2. Block 3 creates and records unique identification numbers for each simulation object. Block 4 creates incidents associated with each "Component" object. These incidents include deterioration mechanisms, rest of engine damage and foreign object damage.

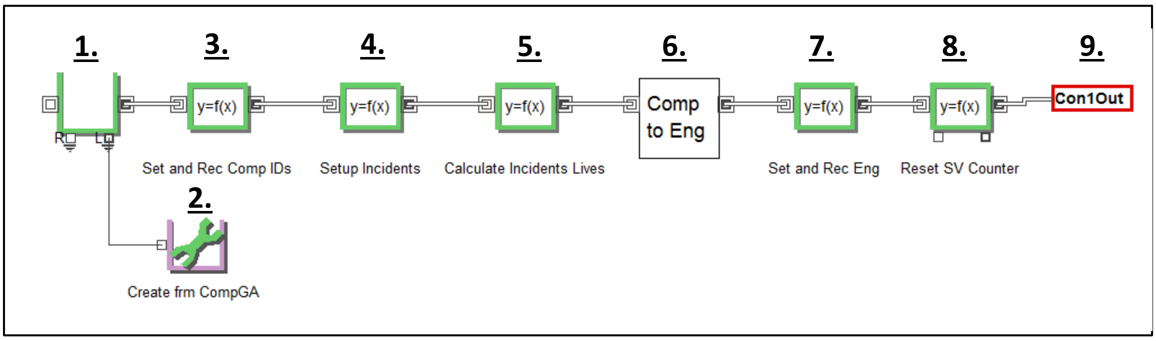


Figure 3-21: Contents of "Setup Engine" block.

Block 5 uses the probability distribution parameters for the relevant incident to stochastically generate an incident life. The incident with the lowest life is what causes the component or engine to require maintenance. Block 6 aggregates the "Component" objects into "Module" objects, which are similarly aggregated into "Engine" objects. Block 7 sets and records the unique engine identification number and block 8 sets the shop visit counter for the simulation. Block 9 is the exit block which allows the simulation objects to leave the "Setup Engine" block and enter the top level process shown in Figure 3-20.

3.5.2.2. "Find Shop Visit Cause" Block

When the "Engine" object enters this block, the incident with the lowest life associated with the engine is identified. The incidents that can cause a shop visit are routine shop visits or deterioration mechanisms failing prematurely causing a disruption. When the incident with lowest life is found, the simulated time of the model progresses to that of the incident. The "Engine" object then exits this block back into the top level process.

3.5.2.3. "Perform Maintenance" Block

The maintenance procedure of an engine is typically decided by the engine manufacturer and the airline operator. Figure 3-22 shows the maintenance plan implemented in this model. An engine can be sent to an overhaul base for any of the following reasons: 1) scheduled maintenance, 2) disruptions due to on-wing inspection or engine health monitoring, or 3) sudden engine failure. On arrival at the overhaul base, it is decided what engine level work-plan is required. The engine could be sent for either a check and rectify, refurbishment or an overhaul. It is from the engine level work-plans, that the maintenance work-plan for each individual module is derived.

In an engine level check and rectify work-plan, only the module where a problem was reported is opened. This module will be subjected to a module level check and rectify work-plan. The remaining modules will be assigned to serviceability work-plans where surface checks are performed. It is possible that at module level serviceability and check and repair stages, defects may be found. This will result in the module being sent for an overhaul. In an engine level refurbishment work-plan, different modules are assigned different module level work-plans depending on engine operating guidelines. In an engine level overhaul work-plan, all modules are assigned to module level overhaul work-plans.

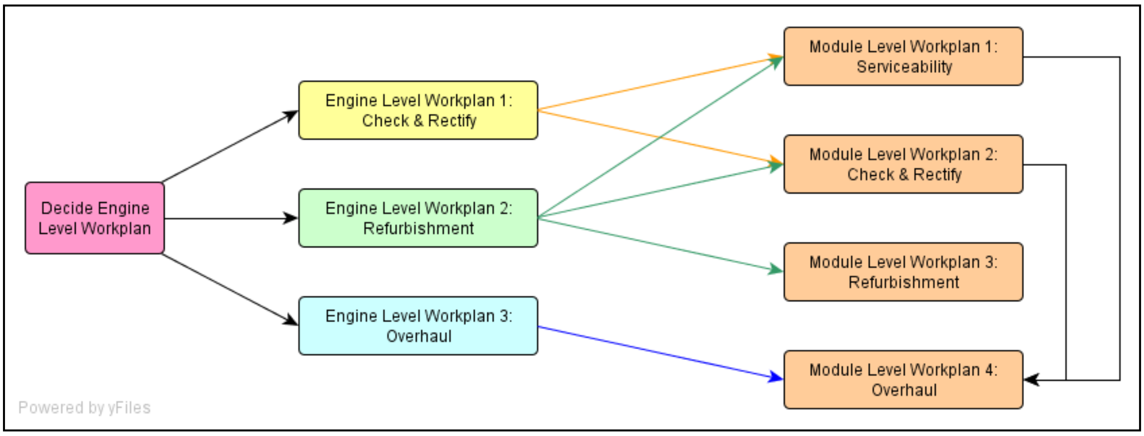


Figure 3-22: Engine maintenance logic.

Figure 3-23 shows the implementation of the above-mentioned logic in ExtendSim®. As before, this is a hierarchical block which is represented as a single block in the top level process; Figure 3-20. Block 1 in Figure 3-23 is where the “Engine” object enters the “Perform Maintenance” block. When the “Engine” object enters block 2, it is assigned an engine level work plan depending on the cause of the shop visit. If it is an unplanned shop visit, the “Engine” object is assigned a level 1: check and rectify work plan. Planned shop visits can be a level 2: refurbishment, or a level 3: overhaul, depending on the number of flight cycles that the engine has completed. After which the “Engine” object proceeds into block 3 and is sent into the block 4 which corresponds to the assigned engine level work plan.

Blocks 5, 6, and 7 first disassembles the “Engine” object into “Module” objects, then assigns the modules a module level work plan. Module level work plans can be either serviceability, check and rectify, refurbishment or overhaul. The blocks bounded by the box labelled 8 help to queue and direct the “Module” objects to the block 9 which corresponds to the appropriate module level work plan. In block 10, module level 1: serviceability block, the module undergoes an inspection. Whether or not the “Module” object requires further work is determined by whether the component has any incident lives that are less than that of the simulated model time. If the “Module” object requires further work, it is sent to block 13; the module level 4: overhaul block.

In block 11, module level 2: check and rectify, the damaged component is replaced or repaired and the rest of the module is inspected. As with block 10, if there are further problems found, the module is sent to block 13.

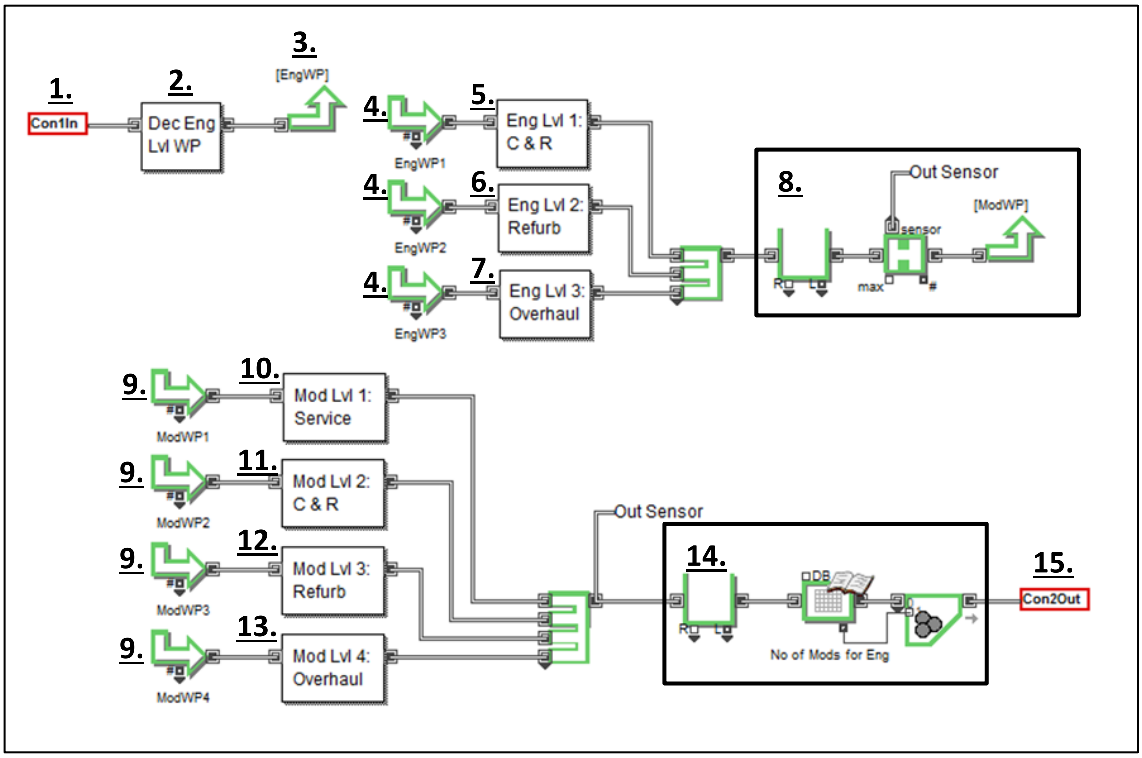


Figure 3-23: Contents of "Perform Maintenance" block.

The process and logic in block 12, module level 3: refurbishment, and block 13, module level 4: overhaul, are the same. The only difference between the two blocks is that for the overhaul case, the module is completely disassembled. In the refurbishment case, some sub-assemblies are left assembled. To account for this, the time taken for refurbishment is shorter than an overhaul. The following section details the process in both blocks 12 and 13. The blocks bounded by the box labelled 14 reassemble the "Module" objects into an "Engine" object. Finally at block 15, the "Engine" object exits the process and returns to the top level process loop (Figure 3-20).

3.5.2.4. "Module Level 3: Refurbishment" and "Module Level 4: Overhaul" Blocks

Figure 3-24 shows the contents of blocks 12 and 13 in the "Perform Maintenance" block. The "Module" object enters from the "Perform Maintenance" block through block 1 in Figure 3-24. When the “Module” object reaches block 2, it is disassembled into its constituent “Component” objects. The “Component” objects then move to block 3 to determine if its group A life has been exceeded. If the life has been exceeded, the part is scrapped. This applies to group A critical parts such as discs and shafts.

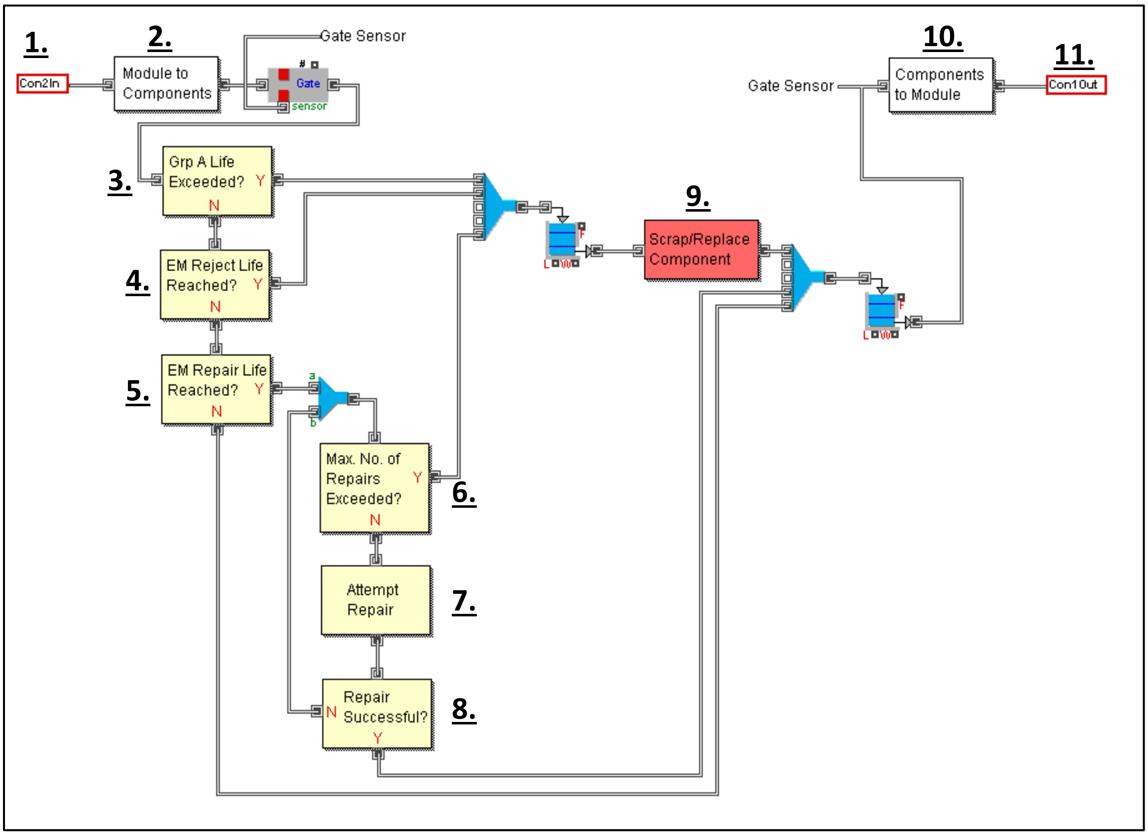


Figure 3-24: Contents of Module Level 3: Refurbishment and Module Level 4: Overhaul Blocks.

At block 4, the “Component” object is assessed against its Engine Maintenance Rejection life. This is the point when the component can still be repaired for a given deterioration mechanism. If this is exceeded, the component is scrapped by sending the “Component” object to block 9. At block 5, it is determined whether the “Component” object requires repair. If its Engine Maintenance Repair life is exceeded, the component is sent for repair. Otherwise, no work is performed on the component.

At block 6, the “Component” object is checked whether it has exceeded the maximum number of repairs. The component is scrapped if it has. At block 7, attempts are made to repair the component. There is a probability that the repair may not be successful. If this is the case, additional repairs may be attempted until the maximum number of repairs is reached. At block 8, the “Component” object is checked whether is repair was successful. If it was, the component is returned to the engine. At block 10, the “Component” objects are reassembled into a “Module” object. The “Module” object exits this process through block 11 where it is returned to the “Perform Maintenance” block.

3.5.2.5. “Check for End of Simulation” and “End Simulation” Block

These are the last two blocks in top level process loop; Figure 3-20. The “Check for End of Simulation” block assesses whether the simulated time has exceeded the time set by the model inputs. If input simulation time required is not exceeded, the “Engine” object is directed back to the “Find Shop Visit Cause” block and goes through another cycle of maintenance. If input simulation time has been exceeded, the “Engine” object is directed to the “End Simulation” block and the simulation terminates.

3.5.3. Maintenance Cost Model Results

ExtendSim® has multiple run functionality and this allows for Monte Carlo simulations which accounts for uncertainty in the input parameters. With the statistical analysis tools available in Vanguard Studio, the simulation results can be displayed comprehensively. Figure 3-25 and Figure 3-26 show samples of the results.

3.5.4. Validation

The nature of the modelled scenario makes it extremely difficult to validate. Aero-engines can have perpetual lives, resulting in long maintenance histories. Thus there is little or no data detailing full life cycle maintenance [105]. To give an indication of the validity of the maintenance model, it was compared against a component level spread sheet simulation developed and used by the Life Cycle Analysis team in Rolls-Royce [95].

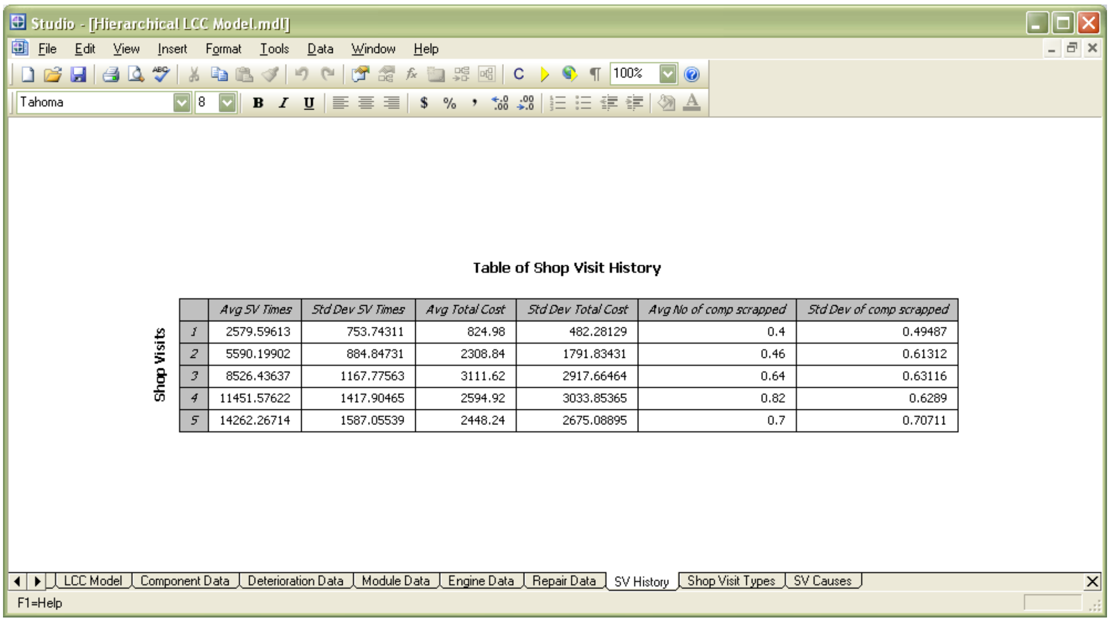


Table of Shop Visit History

Shop Visits	Avg SV Times	Std Dev SV Times	Avg Total Cost	Std Dev Total Cost	Avg No of comp scrapped	Std Dev of comp scrapped
1	2579.59613	753.74311	824.98	482.28129	0.4	0.49487
2	5590.19902	884.84731	2308.84	1791.83431	0.46	0.61312
3	8526.43637	1167.77563	3111.62	2917.66464	0.64	0.63116
4	11451.57622	1417.90465	2594.92	3033.85365	0.82	0.6289
5	14262.26714	1587.05539	2448.24	2675.08895	0.7	0.70711

Figure 3-25: Table of shop visit history.

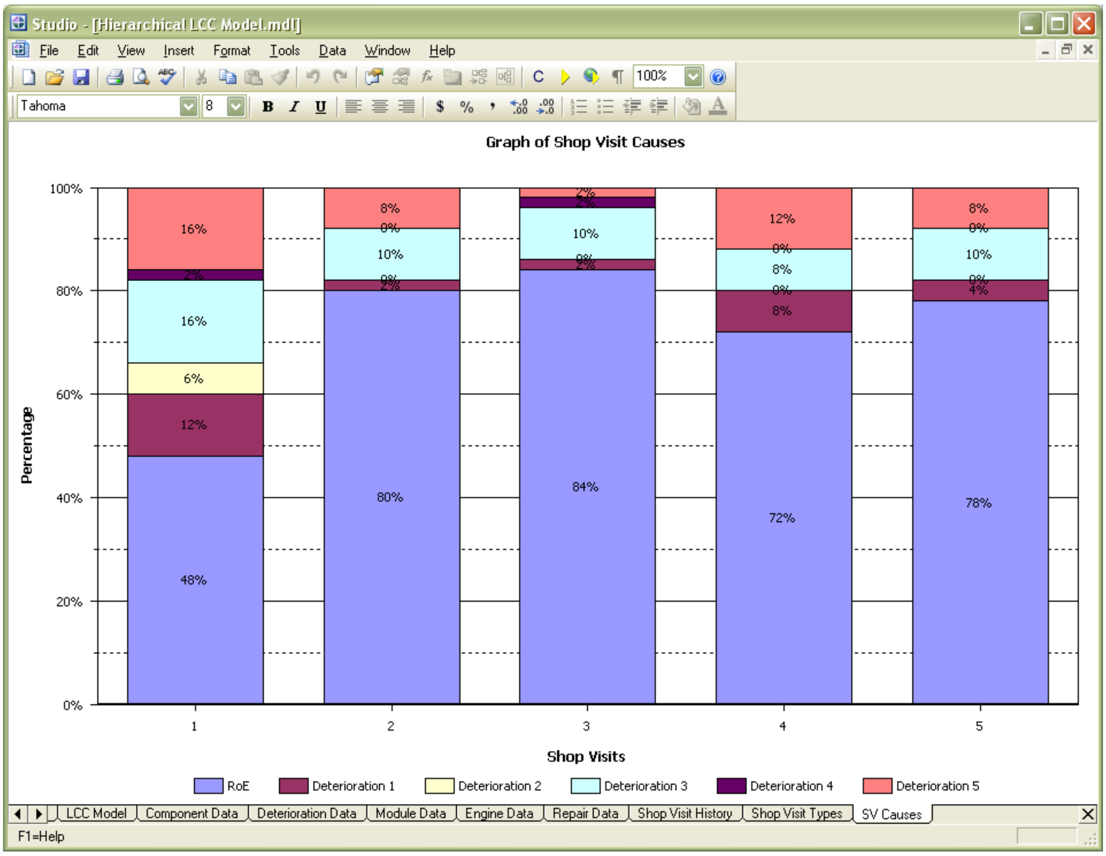


Figure 3-26: Break down of shop visit costs.

The fundamental operation of the spread sheet simulation is similar to that of the DES maintenance model. First it loops through a list of shop visit times. At each shop visit statistical definitions of deterioration mechanisms are used to determine if a component fails. The various repair, scrap and fixed shop visit costs are then aggregated to give the total shop visit costs. The crucial difference between the spread sheet and the DES models is that the spread sheet model does not model the aero-engine components as distinct entities. Instead it uses probabilities obtained from statistical distributions to calculate average number of repaired and failed components at each shop visit. For comparison, both the models were run with the same statistical inputs for a varying number of components and deterioration mechanisms. The coefficient of determination, R^2, was found to be 0.938.

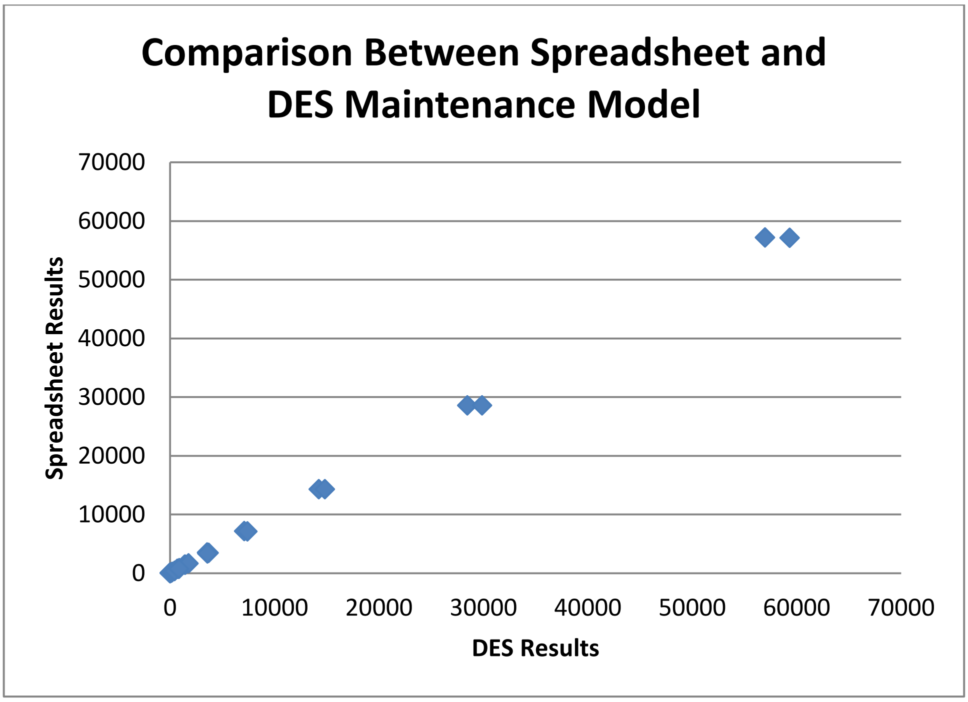


Figure 3-27: Comparison between spread sheet and discrete event simulation maintenance model.

In terms of performance, the run times of the DES model grew significantly with the number of components modelled. Both the Excel and DES models were run for cases for between 1 to 400 components. Figure 3-28 shows the difference in model runtimes between the DES and spread sheet models for 500 Monte Carlo simulation runs. Only one sample per point was taken as the purpose of the graph is to show the significant disparity between the runtimes. The runtime for the DES model with a single component and four deterioration mechanisms was 2.5 minutes. For 400 components, the runtime for the DES model grew to almost 6 hours. It should be noted that each component in the DES model consisted of 17 entities representing the different shop visit causes (deterioration mechanism life limits, design life limits, scheduled shop visits). The Excel spread sheet model consistently had runtimes of less than one minute. In a DES, the simulation time is dependent to the length of its event list. As noted by Yu [110], a longer event list, resultant from having more modelled components, increases the amount of time for an event to be inserted within the event list. This non-linear trend between runtimes and modelled components in a DES was also observed by Yu [110]. In his study, the DES model was used to simulate up to 1,000,000 components.

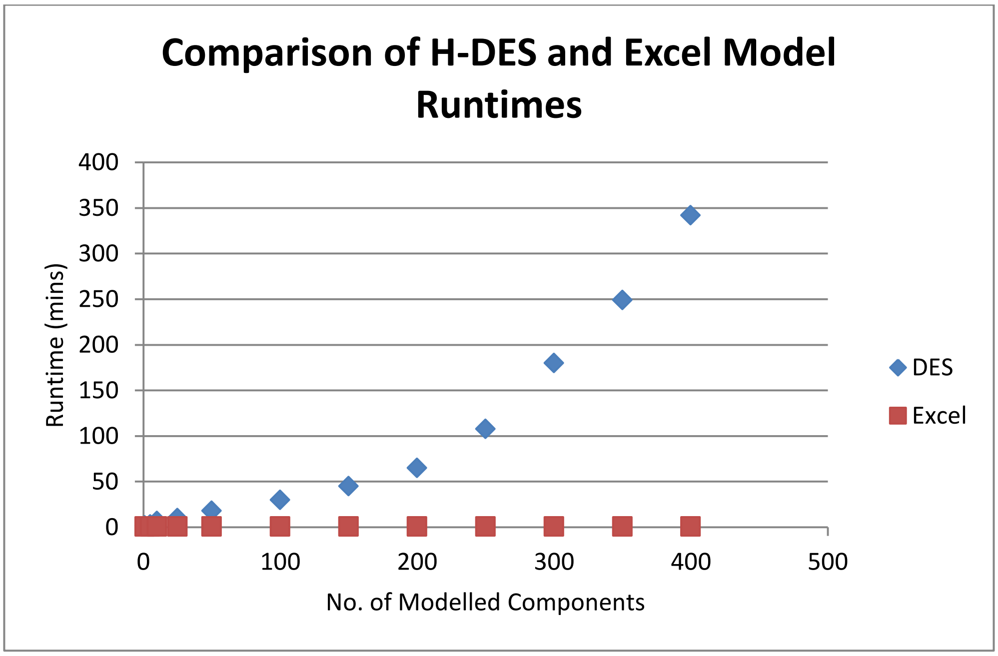


Figure 3-28: Graph of model runtimes vs. number of modelled components.

The event list is a potential bottle-neck for a DES. Despite many efforts to develop techniques to distribute the computing of a DES, its nature demands that events will still have to be processed sequentially in some form; this is discussed in depth by Yu [110]. This ultimately prevents a DES from using parallel processing in order to reduce runtimes. As mentioned earlier, the spread sheet model does not consider individual components. It uses average values of the number of component repairs and failures. This explains the large differences in runtime.

The spread sheet model does however have several limitations. Firstly they are limited in terms of logic representation, data structures and data storage [112]. They are hence a poor tool for the storage and organisation of knowledge. This also makes them difficult for a third party to comprehend [9]. An example of this is given in Harrison [1] where it was observed that an existing Rolls-Royce maintenance spread sheet simulation tool, DMTrade [95], used extensively in another department of the company had mixed success when it was deployed to the designers of another project. This tool's complexity was such that only a minority of individuals were successful in mastering it and using it in the design process. The lack of transparency in spread sheets also makes them highly susceptible to errors. Panko [113] reviewed seven field audits of real organisational spread sheets and found that at least 86% of spread sheets contained errors. Spread sheet models are notoriously difficult to debug, maintain and modify [114]. As a direct consequence, it is hard to expand the capabilities of an existing spread sheet model. This restriction on future development is a major shortcoming of spread sheet models since it does not foster building models on existing knowledge and expertise.

The DES model developed above, as compared to the spread sheet model, is able to show processes and logic in a more coherent manner. It is also more functionally more adaptable and expandable to suit different scenarios. The LCC model can be extended to consider more than just failure rates and Weibull characteristics; such as inventory levels, environment of use, variation in inventory costs and maintenance plans.

3.6. Results and Discussion

The integrated LCC model was run for ±20% change in two parameters of the turbine blade geometry; the aerofoil span mid-chord and the blade root axial length. Figure 3-29 shows the relationship between LCC and the geometric design parameters. It is unsurprising that the point of minimum LCC is found when both the geometric parameters are at -20%. At this point, the manufacturing cost of the blade is at its lowest. In general, the relationship of LCC with the geometric parameters was linear. This is due to the linear relationships between the geometric parameters and unit cost. The gradient of the contour plot shows that LCC is affected more by a change in blade root axial length than the aerofoil span mid chord.

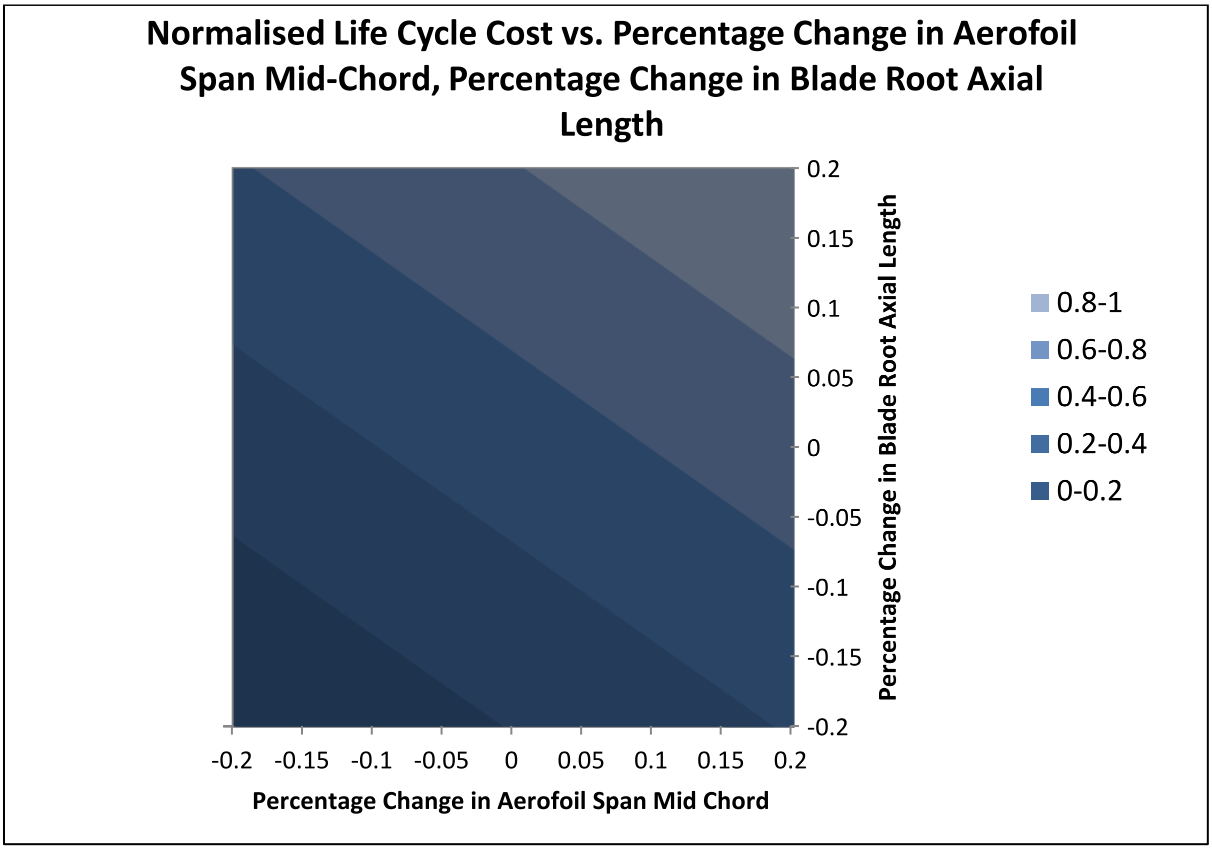


Figure 3-29: Contour chart of normalised life cycle cost vs. percentage change in aerofoil span mid-chord, percentage change in blade root axial length.

A limitation of this model is that it is unable to predict changes in component lives with the changes in geometric parameters. One possible method to generate this relationship is to create a parametric model based on deterioration mechanism life data of similar components of different sizes. This will be dependent on the existence and availability of such data. Another method would be to use finite element analysis. Approaches have been developed which use computer simulations to populate RSMs [115-118]. Essentially these models are surrogate models which allow the prediction of component fatigue and creep lives based on input design parameters such as component geometry, materials, temperatures and stress levels.

This model was constructed in consideration of the issues made in the literature review about LCC models. With the use of an integration work package, the integrated LCC model is modular. Component models can be added or removed to suit the objectives of the study. There is no limitation on the modelling approach used provided that the component models can be run and modified programmatically. The integration package also helps to keep the interfaces and the interactions between the different component models transparent. This is important in managing the complexity of the model and ensuring that the model can be easily understood.

3.7. Summary

An aero-engine LCC model was developed which is able to predict how turbine blade geometric design parameters change LCC. It demonstrates an approach which is modular, customisable, and transparent. It uses commercial software integration package to link three different modelling analyses, hierarchical tree unit cost models, response surface deterioration mechanism life models, and a DES maintenance cost model. With maintenance cost being an important factor in designing for LCC, a generative model is required to provide a flexibility to cost modelling which is not available in more traditional cost modelling approaches. The development of the DES maintenance cost model is described. With the graphical interface provided by the software, the maintenance model is able to capture complex process flows, which enhances model understanding. This integrated model forms the basis of more complex models which will be presented in the later chapters.

4. Using MBSE in Life Cycle Costing

In this chapter, a LCC model was developed to study how the Turbine Entry Temperature (TET) and Cooling Flow Fraction (CFF) of the aero-engine affect LCC. The case study was used to demonstrate how MBSE can support the construction of complex integrated models. This work also serves to demonstrate how complexity can be added to the model developed in chapter 3 to address different objectives.

4.1. Background

Two articles in Flight International [119, 120] highlight how aero-engine manufacturers constantly look to reduce fuel consumption of their engines. This trend is motivated by the rise in fuel prices and the competition between the aero-engine manufacturers to provide the most competitive engine. The ability of gas turbines to operate at higher temperatures is critical to improving their performance. This has driven gas turbine manufacturers to continually pursue higher Turbine Entry Temperatures (TET). Figure 4-1 shows how TET in Roll-Royce engines has risen over a 60 year period [121].

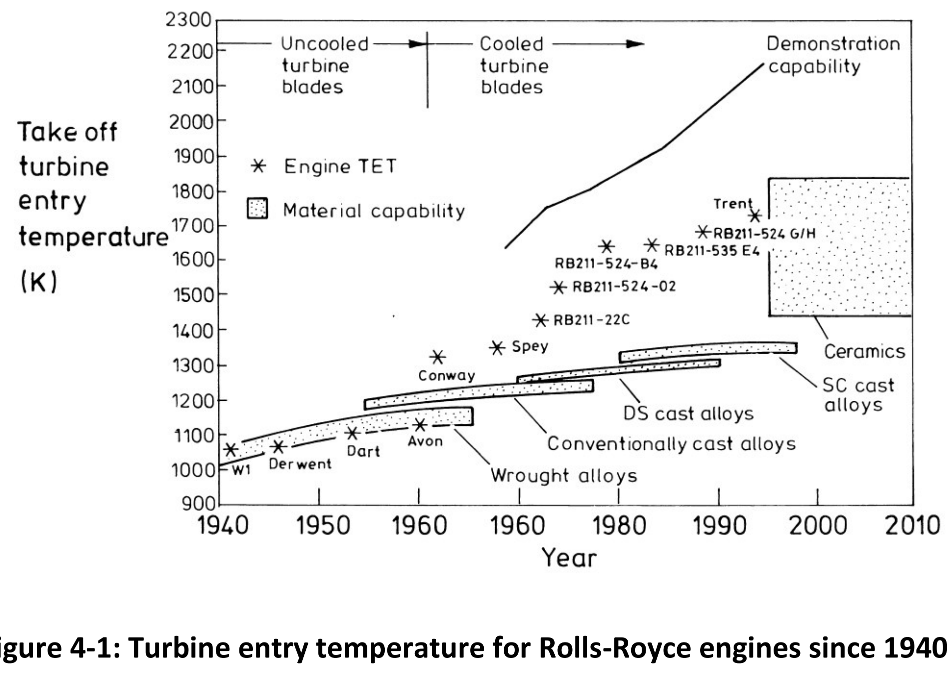


Figure 4-1: Turbine entry temperature for Rolls-Royce engines since 1940 [121].

TET impacts a variety of engine attributes which include thermal efficiency, overall pressure ratio, cooling requirement, engine deterioration and engine core weight. This presents a challenge in assessing what value of TET would result in an optimal engine design. A higher TET would be able to improve thermal efficiency and subsequently specific fuel consumption (SFC). This however would also result in the acceleration of engine deterioration which leads to higher maintenance costs. To address this, cooling flows can be increased to reduce the metal temperatures of the components in the hot section of the turbine; but this would reduce the thermal efficiency of the engine. This scenario provides a useful setting to demonstrate how LCC can be used to compare designs when faced with various competing factors.

For this case study, the main objective was to analyse how TET and cooling flow fraction affect LCC for a single aircraft. Data used for the models was based on the IAE V2500-A1 engine for the Airbus A320 aircraft. To keep the scope of the study manageable, only the effects of the change to the High Pressure (HP) turbine stages were modelled. Nozzle guide vanes and turbine discs were also not included. It was also assumed that the design changes would only have an effect on fuel costs and maintenance costs related to the turbine blades. In reality, TET affects the design of the entire engine so a whole engine analysis would be required to measure the impact of a TET change. As only the relative changes in LCC are of interest, rather than the absolute values, the other cost elements were ignored.

4.2. MBSE Enabled LCC Process

Having identified the issues concerning LCC model development and how MBSE can be used to support system model development, a MBSE enabled LCC process was conceived, shown in Figure 4-2. It links the LCC process in section 2.5.1 and the OOSEM methodology in section 2.6.2. The MBSE enabled LCC process defines the OOSEM activities for each stage of the LCC process. The OOSEM activities “Optimise and Evaluate Alternatives” and “Validate and Verify System” are classed as sub-activities and they occur throughout the OOSEM activities as required. The “Optimise and Evaluate Alternatives” applies to the optimisation the most suitable candidate system architecture.

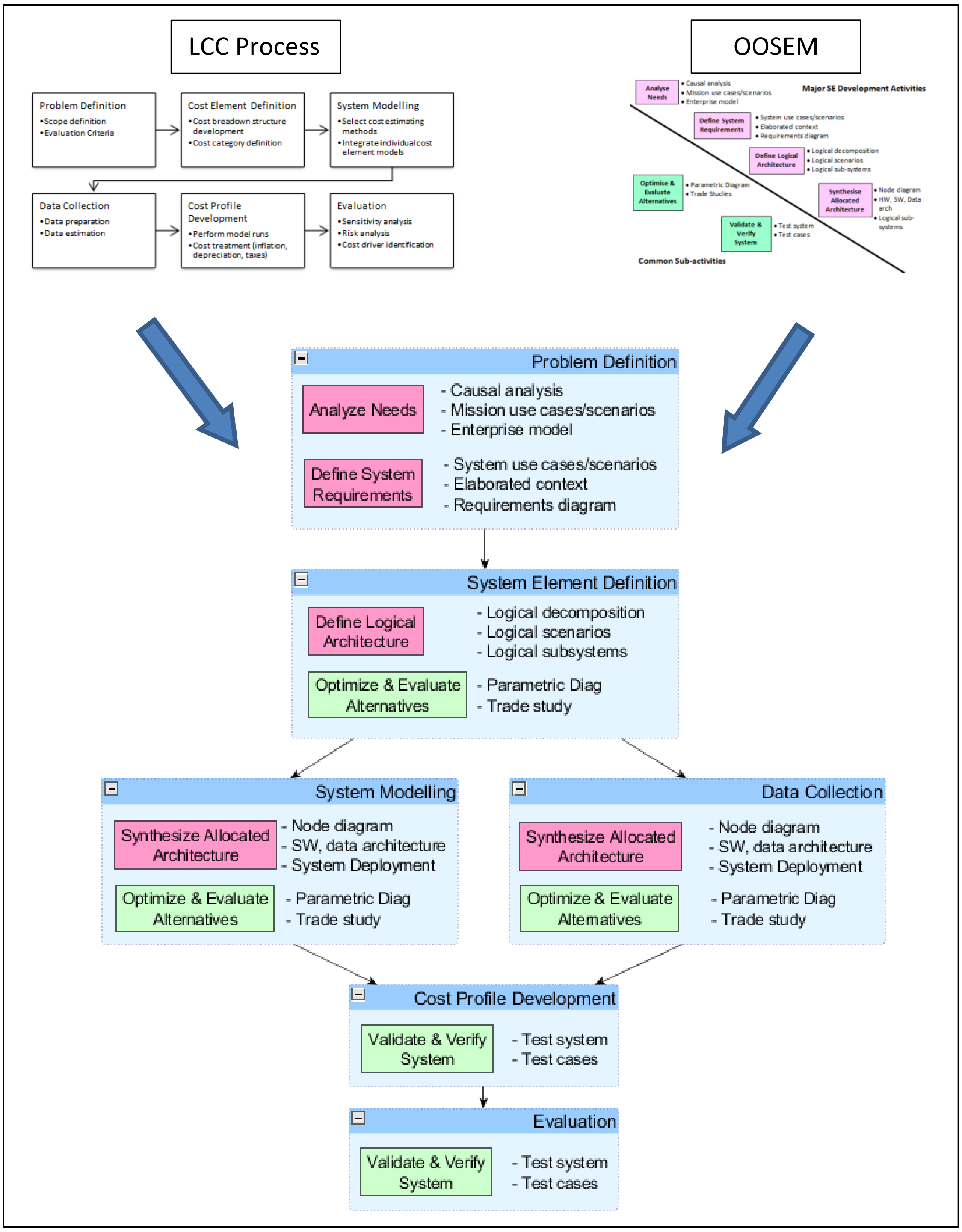


Figure 4-2: MBSE enabled LCC process.

As this study was limited in terms of options, time and resources, only one system architecture was developed. The second step in the LCC process, “Cost Element Definition”, was also renamed to “System Element Definition” to reflect that elements other than cost will also need defining when conceptualising the logical architecture of the integrated LCC model. The “System Modelling” and “Data

Collection" steps were reorganised to run concurrently as these two steps are inter-dependent. Sections 4.3 to 4.7 describe the individual steps of the process in greater detail.

4.3. Problem definition

At the problem definition stage, the aim, scope, as well as evaluation criteria of the LCC model was laid out. The "Analyze Needs" and "Define System Requirements" stages of OOSEM were applicable at this point. Figure 4-3 shows a use case diagram depicting the stakeholders involved in the LCC analysis. The use case diagram shows the role each stakeholder played in the execution of the LCC analysis, as well as the interdependencies between the roles. As mentioned in the OOSEM discussion, analysing the needs of the LCC problem helped in the formulation of requirements. Figure 4-4 shows the requirements diagram that was constructed for the LCC model. The top level requirement for the LCC model was decomposed into categories which represent the various stakeholders: LCC, Engine Design, Manufacturing, Lifing, and Unit Cost. More detailed requirements were derived for each of these categories. Figure 4-4 shows the requirements for modelling fuel cost and maintenance cost which were derived from the LCC requirement.

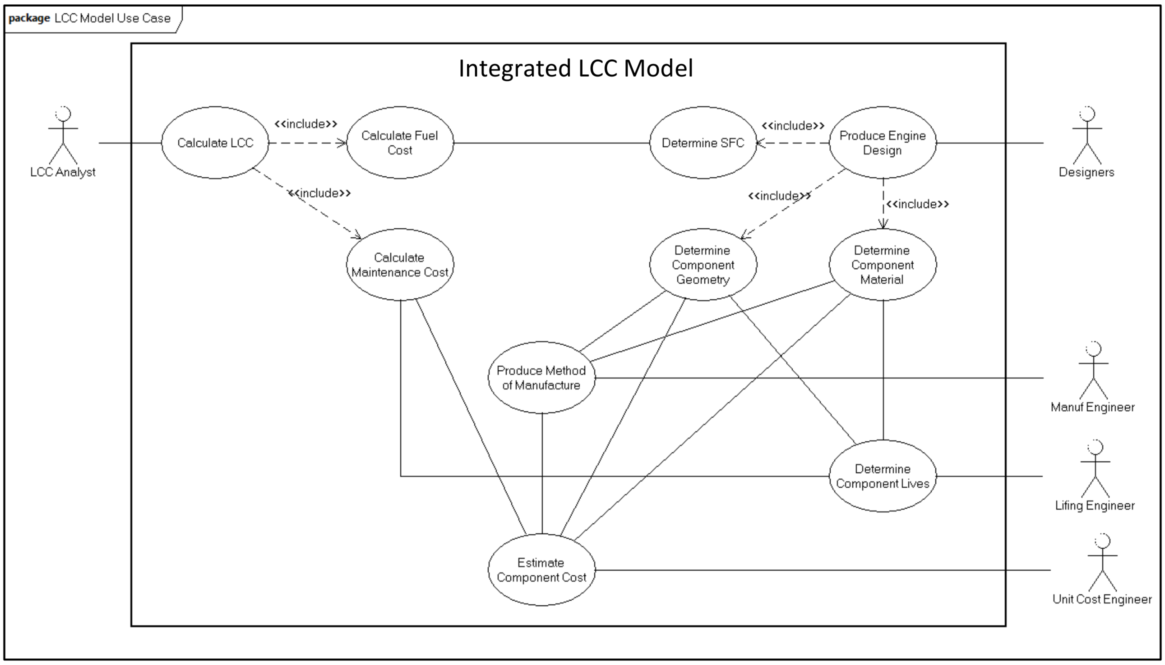


Figure 4-3: Use case diagram for LCC model.

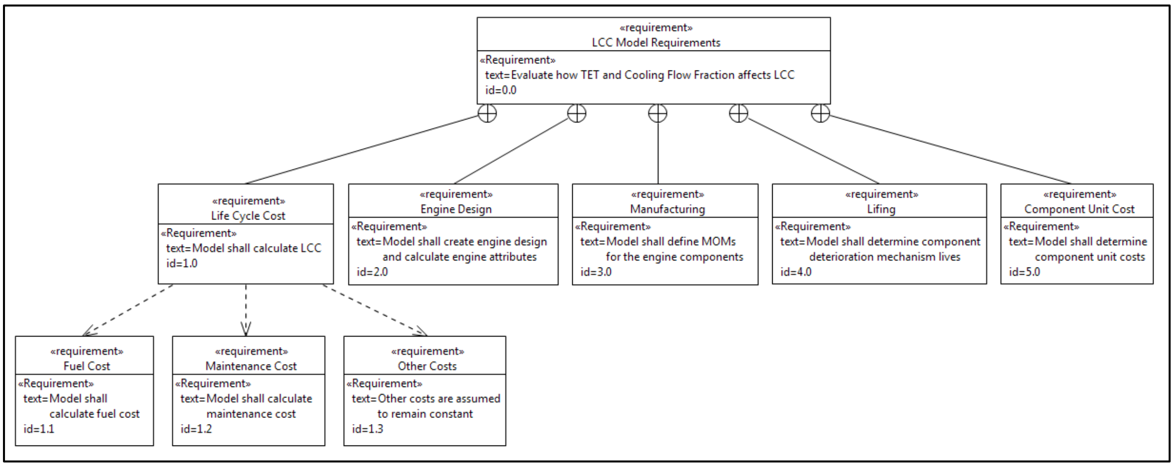


Figure 4-4: Requirements diagram for LCC model.

4.4. System Element Definition

At this stage, the elements of the LCC study were defined and the logical architecture of the model was conceptualised. This involved decomposing the system into logical components which satisfied the system requirements. The logical architecture serves as an intermediate level of abstraction between the system requirements and the software architecture of the LCC model. Figure 4-5 shows the logical decomposition of the LCC model. The logical architecture of the model follows the decomposition of the requirements: LCC, Engine Design, Manufacturing, Lifing, and Unit Cost. Each of these elements of the LCC model was then decomposed further to address specific details of the requirements.

4.5. System Modelling

At this stage, a model was required to quantify the cost elements in the LCC analysis. It served to simulate the relationships between the study inputs and outputs. To build the model, the physical architecture of the system had to be established. The logical components of the LCC model, determined in the previous stage, were assigned to physical components. Figure 4-6 shows the allocated architecture of the LCC model. The allocated architecture groups the logical components identified in the previous stage into physical components which are represented by the swim-lanes in Figure 4-6. Data flows between the physical components are also shown. For example, TET is input into the Engine Design Tool which produces an estimate of component geometries (blade lengths, disc diameters, fan size). The component geometry is then forwarded to the Component Unit Cost model which estimates forging and machining costs.

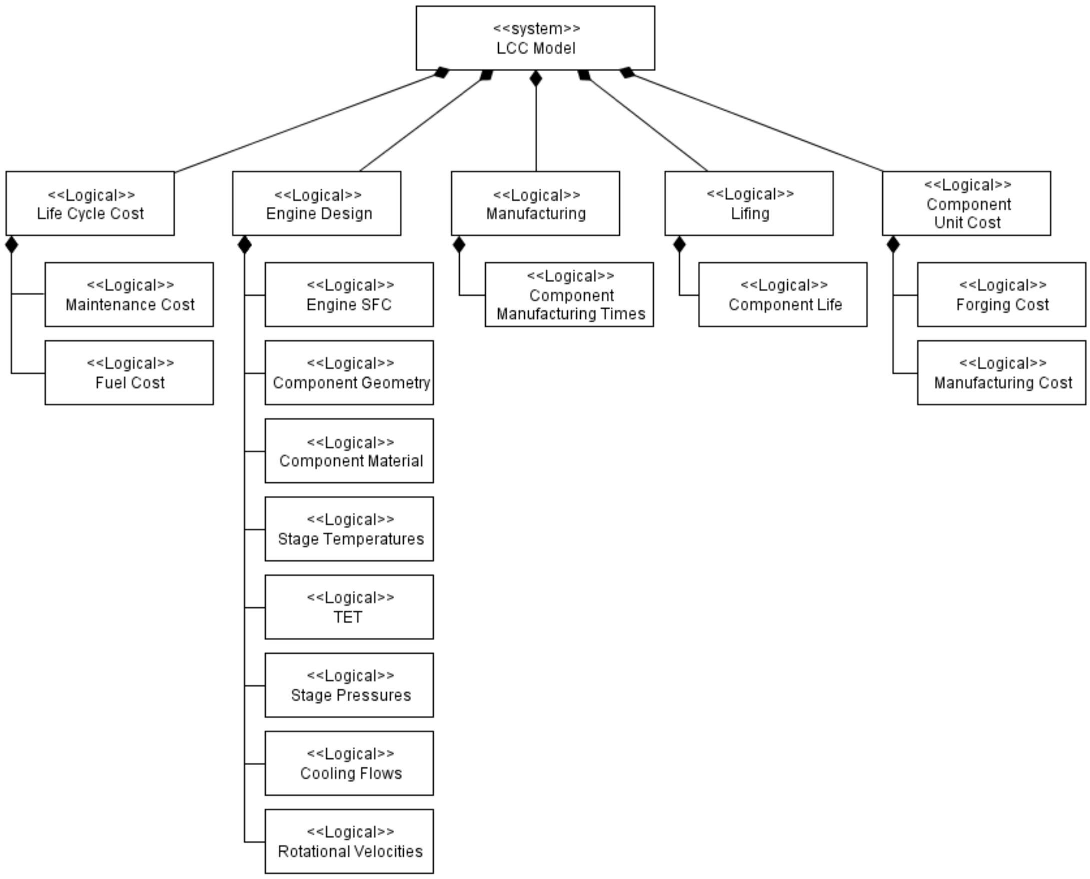


Figure 4-5: Logical decomposition of the LCC model.

The implementation of the integrated LCC model was done in Isight, which is a tool used for integrating a variety of applications, in order to automate the exploration of design spaces and identify of optimal performance parameters. The allocated architecture diagram was used in identifying the required analysis models and the data flows between them.

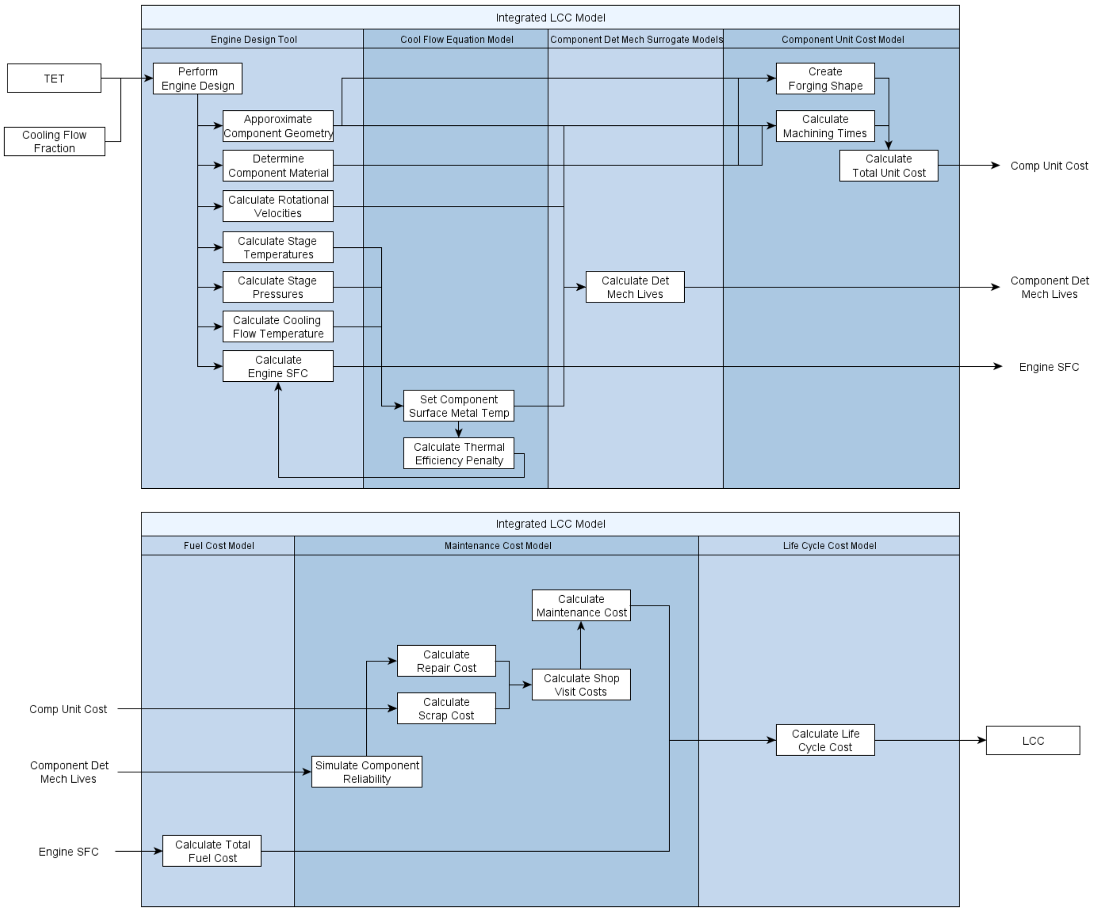


Figure 4-6: Allocated architecture of LCC model.

Figure 4-7 shows the integrated model in Isight, where the swim-lanes in Figure 4-6 correspond to the Isight components in the main process flow. These Isight components run the various analyses in the sequence that they are positioned in the process flow. Within Isight, with the exception of the Engine Design spread-sheet and the fuel cost equation, the analysis models were executed by Isight via the operating system command line interface. Data input and output was done with comma-separated value (CSV) files transferred between the individual analysis models and Isight. For the Engine Design spread-sheet, the engine design had to be optimised to get the required maximum thrust (2500N). The bypass ratio (BPR) and the input mass flow rate were adjusted until the maximum thrust target was reached. In the sections below, further details of the analysis models are described. In Figure 4-7 the boxes around the engine design spread-sheet and the deterioration mechanism models indicate that these tools were not developed as part of this research.

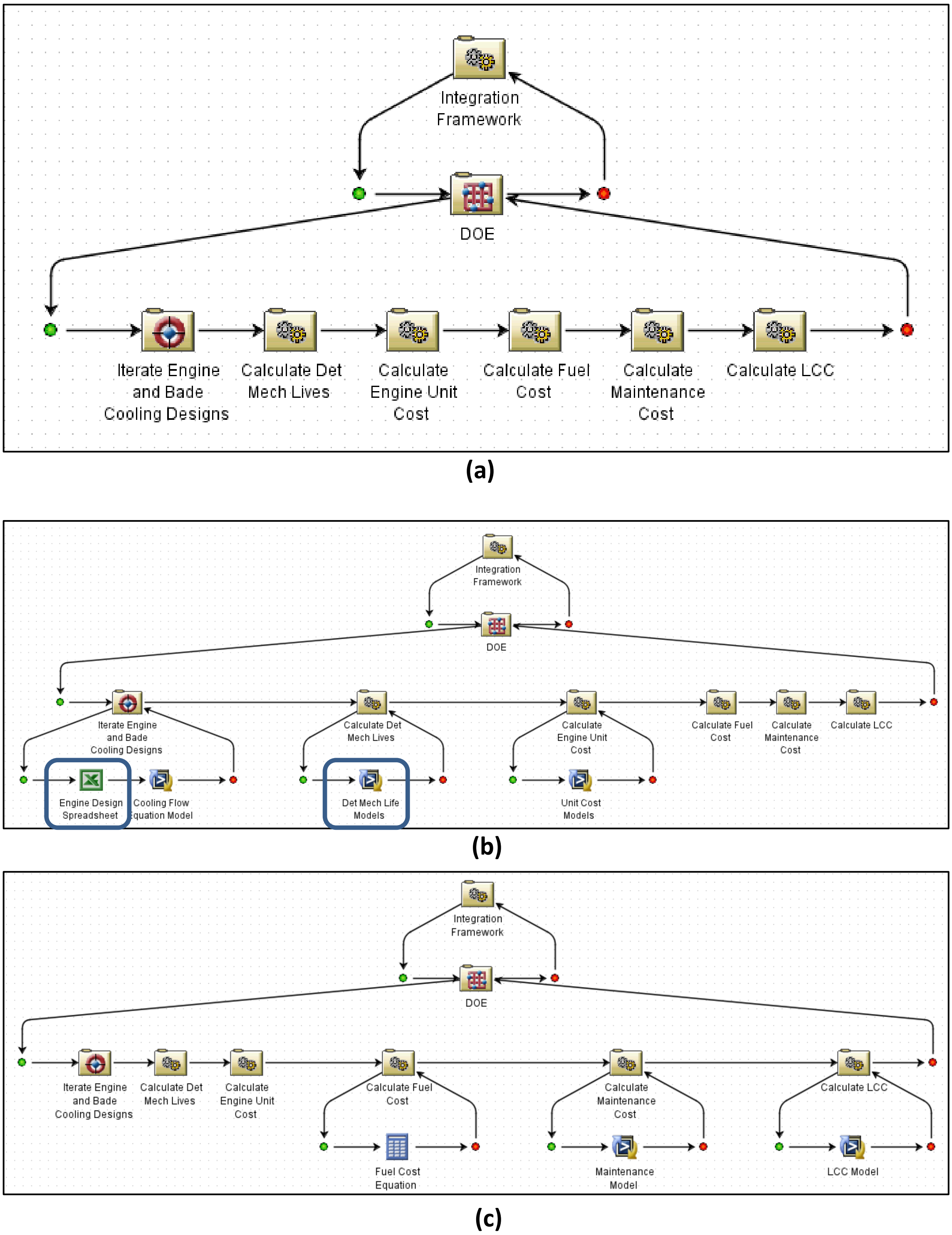


Figure 4-7: LCC case study implemented in Isight. (a) Workflow nodes collapsed. (b) and (c) Workflow nodes expanded.

4.5.1. Engine Design

The study begins from a change in the preliminary engine design. Figure 4-8 shows how the characteristics of the engine affect each other. Initially, thrust and flight speed will be set by the aircraft manufacturer. Based on this, the engine designer selects a fan face axial Mach number and a fan hub/tip ratio which will provide a good balance of intake size against drag and fan efficiency. The choice of total engine mass flow rate controls the fan outer diameter. This controls other parameters such as fans size, fan pressure ratio, and jet velocities. Finally, an appropriate TET is selected which influences characteristics such as core size, thermal efficiency, overall pressure ratio, thrust and SFC. To produce an improvement in SFC, the thrust requirement must be held constant for the varying levels of TET.

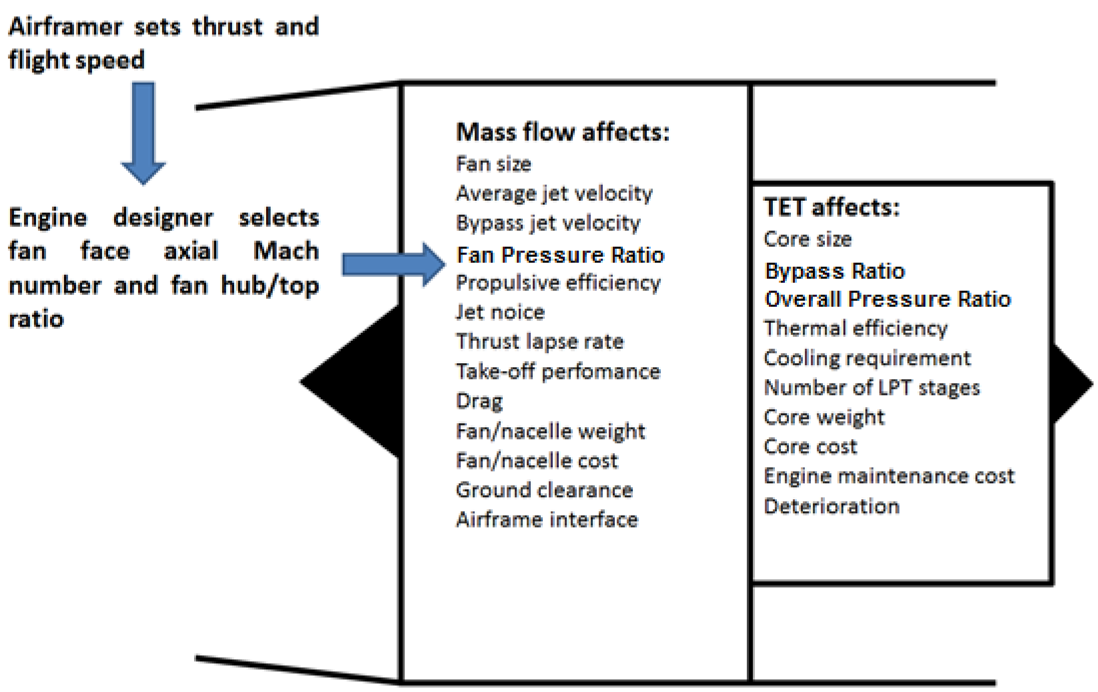


Figure 4-8: Selecting turbofan size and TET [122].

Figure 4-9 shows an Excel spread sheet model of the open-gas turbine cycle used to predict the changes in engine design with changes in TET and cooling flow fractions used in Rolls-Royce for teaching gas-turbine design. The model is able to generate the diameters, angular velocities, temperatures and pressures for the various stages in the designed turbine. Figure 4-12 is a plot generated from the spread sheet. It shows how SFC changes with TET for an engine producing 2500N of thrust.

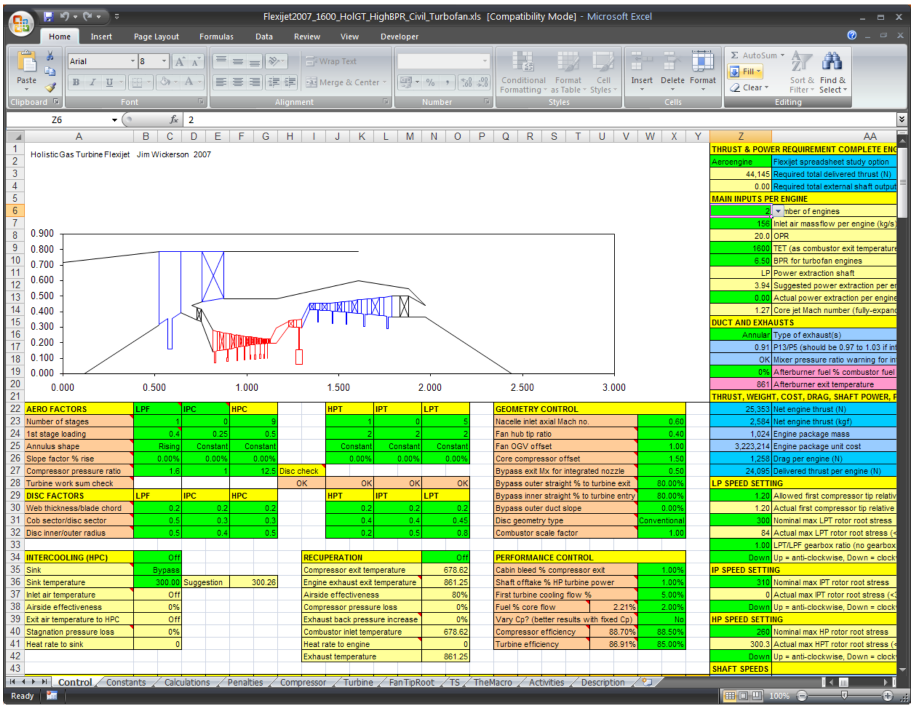


Figure 4-9: Engine design spread sheet model.

4.5.2. Cooling Flow

Turbine blade cooling has played an important role in the rise of TET in gas turbine engines. It achieves this by extracting air from the compressor section of the turbine and releasing it through cooling holes in the turbine blades. However, extracting cooling air from the compressor stage affects the thermal efficiency of the turbine as the amount of air available to do work is reduced. Aerodynamic losses induced from mixing the cooling and mainstream air flows together impede the thermal efficiency even further. Cooling flow design is an optimization between a myriad of factors such as stage temperatures and pressures, allowable material temperatures, life required and the technology level of cooling and materials [123]. In this study, an equation from Wilcock [124] is used to approximate the thermal efficiency penalty due to cooling.

This is given as:

$$\Delta\eta_{th} = \frac{1}{(\theta - \xi_{HP})}\left[\psi_{HP}\left(\frac{\xi_{HP}}{\zeta_{HP}} - 1\right) + \psi_{LP}\left(\frac{\xi_{LP}}{\zeta_{LP}} - 1\right)\right], \qquad (4.1)$$

where $\theta = T_{\text{combuster outlet}}/T_{\text{ambient}}$. The high pressure cooling flow fraction is ψ_{HP} and the low pressure cooling flow fraction is ψ_{LP}. The HP or LP compression temperature ratio is $\xi = r^m$, where $m = (\gamma - 1)/(\gamma\eta_{poly})$ and r is the pressure ratio over which the coolant stream is compressed (from ambient to supply). The HP or LP expansion temperature ratio is $\zeta = r^n$, where $n = \eta_{poly}(\gamma - 1)/\gamma$ and r is the pressure ratio over which the coolant stream is expanded (to ambient) after mixing in the turbine. The main assumptions to be considered when using the above equation is that it ignores combustor pressure losses, entropy production due to friction in the internal blade passages, and the kinetic energy dissipation during mixing [124].

From the same paper, an expression for ψ is given as:

$$\psi = \frac{K_{cool}}{(1+B)}\left\{\frac{\varepsilon_0 - \varepsilon_f[1 - \eta_{int}(1 - \varepsilon_0)]}{\eta_{int}(1 - \varepsilon_0)}\right\}, \qquad (4.2)$$

where K_{cool} is the cooling flow factor, η_{int} is the internal cooling efficiency.

Here, B is defined as:

$$B = Bi_{tbc} - \left(\frac{\varepsilon_0 - \varepsilon_f}{1 - \varepsilon_0}\right) Bi_{met}, \qquad (4.3)$$

where Bi_{tbc} is the Biot number for the thermal barrier coating.

In equations (7) and (8), ε_f is the film cooling effectiveness and the blade cooling effectiveness ε_0 is defined as:

$$\varepsilon_0 = \frac{T_{0g} - T_{met,ext}}{T_{0g} - T_{0c,in}}, \qquad (4.4)$$

where T_{0g} is the relative total temperature of the mainstream gas, $T_{0c,in}$ is the relative total temperature of the coolant entering the blade passages and $T_{met,ext}$ is the allowable external surface metal temperature (assumed constant over the blade).

By relating the four equations above, the thermal efficiency penalty due to cooling can be made a function of the allowable external surface metal temperature. These equations were used a model within the Vanguard software, show in Figure 4-10. Figure 4-11 shows a plot of the thermal efficiency penalty due to cooling against the allowable surface metal temperature of a HP turbine blade. In the case study, the cooling model retrieved data of the necessary stage temperatures and pressures from the engine cycle design model. The cooling efficiency penalty is then calculated for each iteration of TET and CFF.

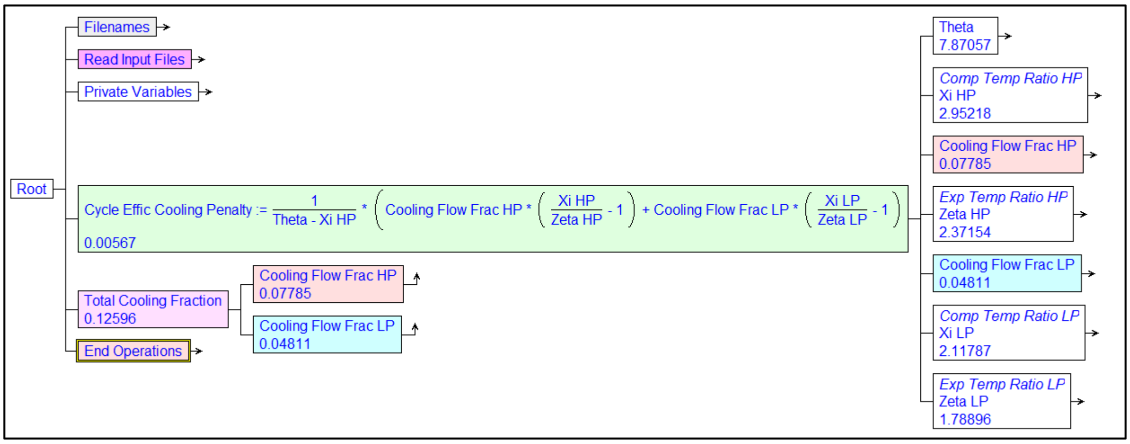


Figure 4-10: Screenshot of cooling flow model.

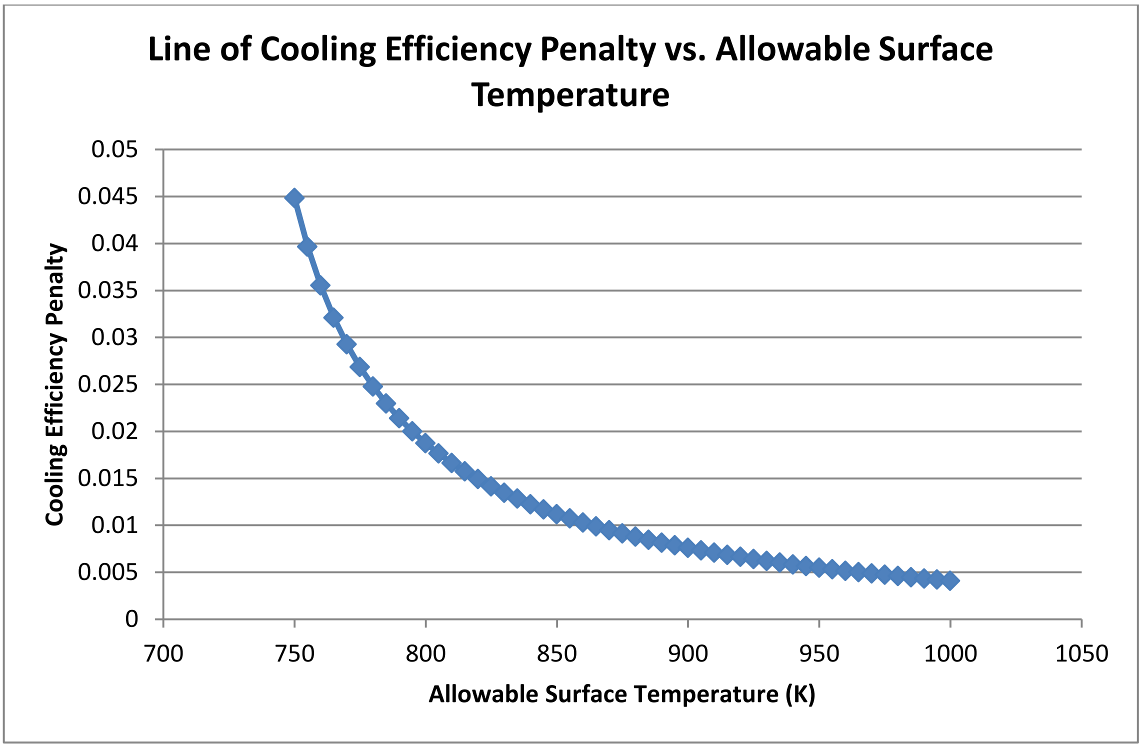


Figure 4-11: Relationship between thermal efficiency penalty due to cooling and the allowable metal surface temperature of a high pressure turbine blade.

4.5.3. Fuel Cost

This model uses the fuel mass flow rate from the engine cycle design model and defined mission profile to calculate the average fuel costs per year.

$$\begin{aligned} &Average\ Fuel\ Costs\ per\ year\ = \\ &Average\ Fuel\ Mass\ Flow\ Rate\ (kg/s)\ \times \\ &Fuel\ Cost\ (£/kg)\ \times\ Annual\ Utilisation\ , \end{aligned} \qquad (4.5)$$

Annual utilisation refers to the percentage of a year that the engine will be operational.

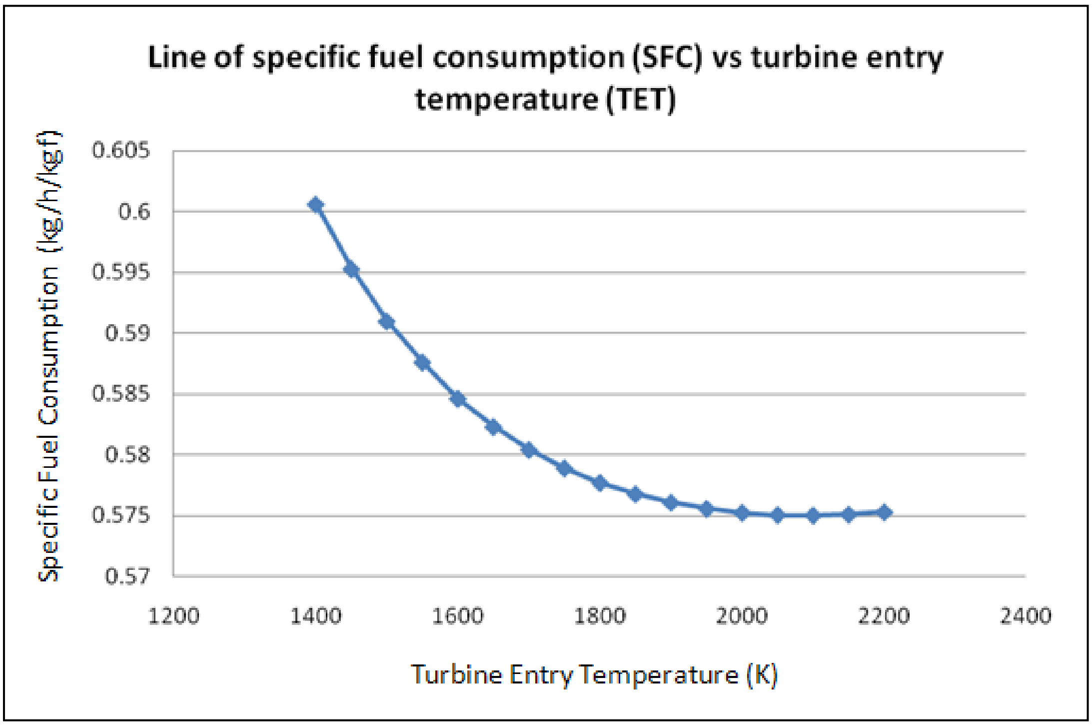


Figure 4-12: Relationship between SFC and TET.

4.5.4. Deterioration Mechanisms

As in chapter 3, the deterioration mechanisms modelled were creep and fatigue. These deterioration mechanisms were used for this case study because they are highly influenced by temperature. Using Rolls-Royce's proprietary finite element analysis code, SC03 [125], Response Surface Models (RSM) were populated which predicted how deterioration mechanism lives change with stress and temperature. Running a FE analysis can be computationally expensive. The use of RSMs is thus necessary to ensure that the integrated system model does not have impractical runtimes.

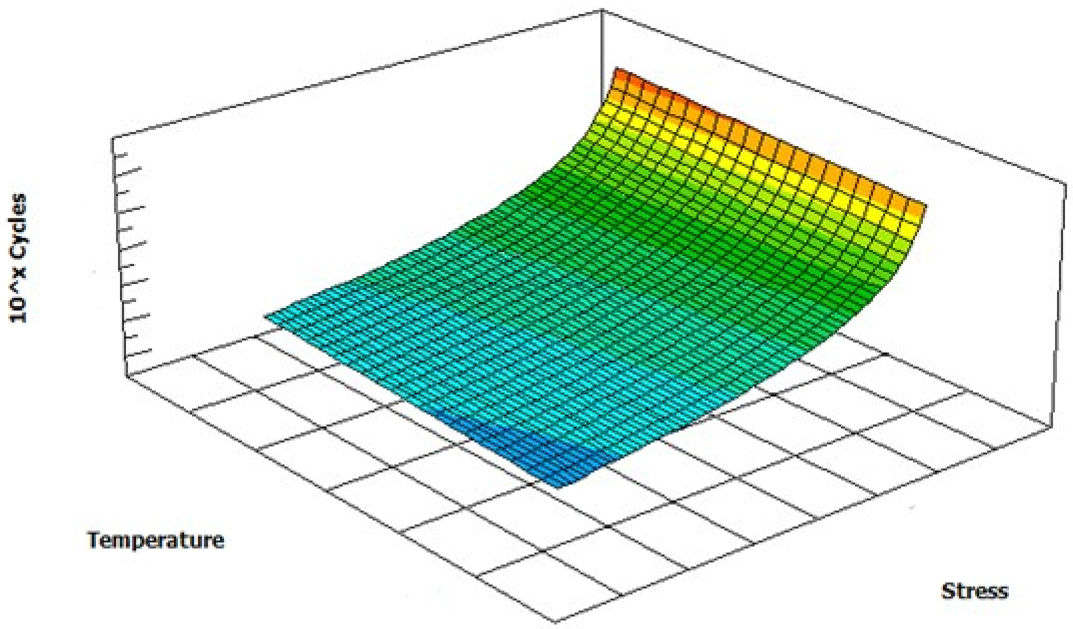


Figure 4-13: Response surface model for deterioration mechanism life.

4.5.5. Maintenance Cost

The DES maintenance model developed in chapter 3 was used to relate component deterioration mechanism lives and manufacturing costs to maintenance costs. The costs considered in the model include fixed shop visit costs, labour costs, repair costs, and component replacement costs.

4.5.6. Unit Cost

The unit cost models, as described in section 3.3.5, predict manufacturing costs. In this case study, the models were used to link component dimensions from the engine cycle design model to engine component unit costs.

4.5.7. Life Cycle Cost Model

The LCC model used in the integrated model follows the assumption made in the study in Chapter 3, which was that LCC should only include cost elements where changes in cost will be incurred. In this study, it was assumed that only the initial engine investment cost, maintenance and fuel cost elements will be affected by the design change. It was the relative change in LCC that was of interest rather than the absolute value. LCC is expressed as equation 4.6.

$$Life\ Cycle\ Cost = \\ Engine\ Investment + Maintenance\ Cost + Fuel\ Cost, \quad (4.6)$$

4.6. Data Collection

Accuracy of input data is crucial to improve the certainty of the LCC prediction. As for data collection, it is required to identify the requirements of input data and to access reliable data sources related to the LCC analysis. Figure 4-14 shows an example of the data requirements of the unit cost model. It illustrates and captures the different sources of data that was used to construct the unit cost model. Where actual data was not available, the elements relevant to the non-available data were estimated based on expert judgements.

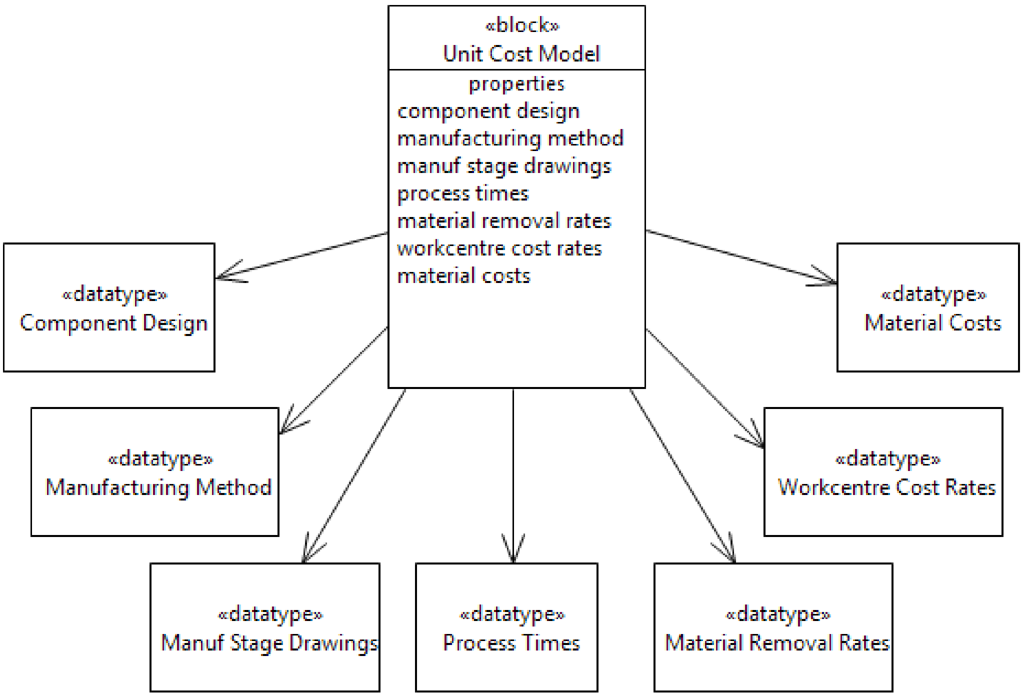


Figure 4-14: Unit cost model data requirements.

4.7. Cost Profile Development & Evaluation

The range of input values for the LCC model was 1700 to 2000K for TET and 0.1 to 0.3 for the cooling flow fraction. A 10 level full factorial design was used to generate the response surfaces. Figures 4-15 and 4-16 show the response surfaces of maintenance and fuel costs respectively. Fuel cost is lowest at maximum TET, 2000K, and minimum cooling flow fraction, 0.1. This behaviour can be attributed to the improved SFC at higher TETs and reduced thermal efficiency penalties due to cooling. Maintenance cost is lowest at minimum TET, 1700K, and maximum cooling flow fraction, 0.3, because this combination of factors produces the lowest turbine blade metal surface temperatures. As a result, the creep and fatigue lives of the components were extended. It is immediately apparent from Figures 4-15 and 4-16 that the trends of fuel cost and maintenance cost are increasing in opposite directions. Figure 4-17 shows the relationship of LCC against TET and total cooling fraction. Performing a LCC analysis gives the cumulative effect of these two competing factors.

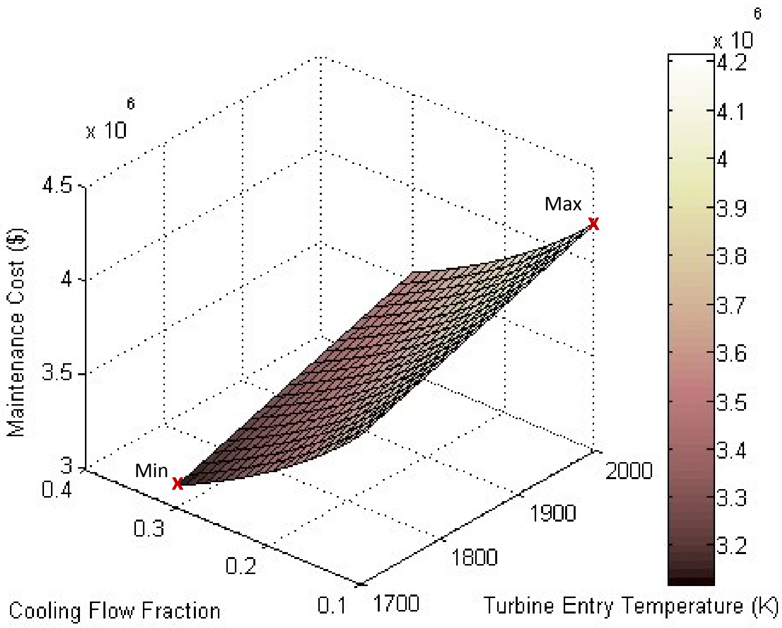


Figure 4-15: Contour plot of maintenance costs against TET and total cooling flow fraction.

It was clear that fuel cost is the dominant factor in the LCC study. The minimum LCC, US$1.0859 billion, occurs when TET is at 2000K and cooling flow fraction is 0.191; however this simplified study only takes into account the impact of HP turbine blades. A more conclusive result can only be arrived at if the other components in the hot section of the turbine are included to give a more accurate estimate of maintenance cost.

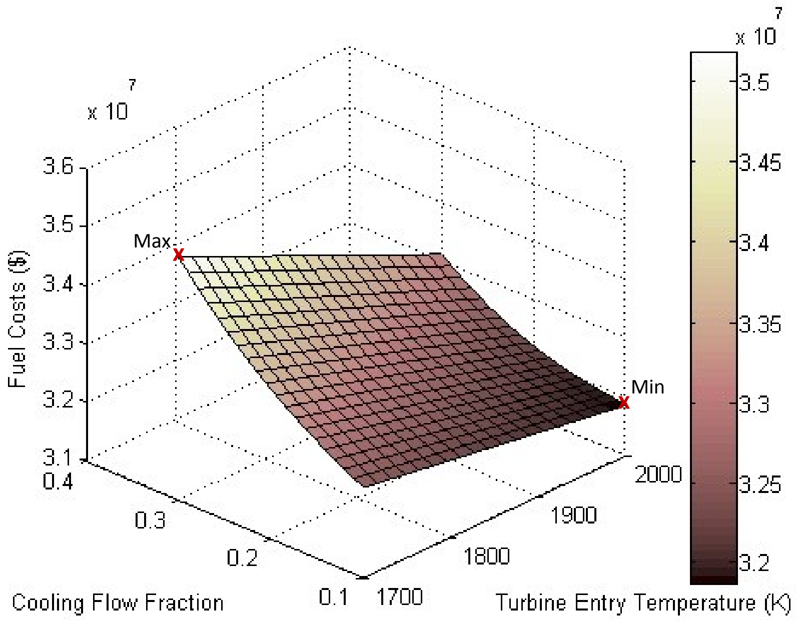


Figure 4-16: Contour plot of fuel costs against TET and total cooling flow fraction.

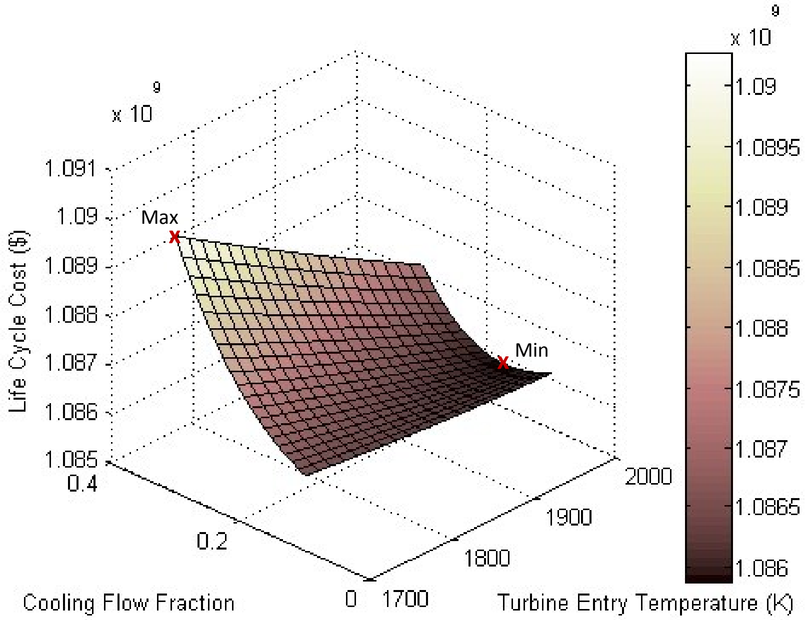


Figure 4-17: Contour plot of LCC against TET and total cooling flow fraction.

It is difficult, if not impossible, to validate an integrated LCC model such as this. Actual data illustrating the relationship between LCC and the inputs, TET and CFF, for a specific engine does not exist and would be unfeasible to generate. To mitigate this, the individual component models will need to be validated and the results of the integrated model will have to be verified by the appropriate subject matter experts. In this case, it was the designers at Rolls-Royce who have endorsed the results. This is illustrated in Figure 4-18 which shows the stakeholders and their involvement in the verification process.

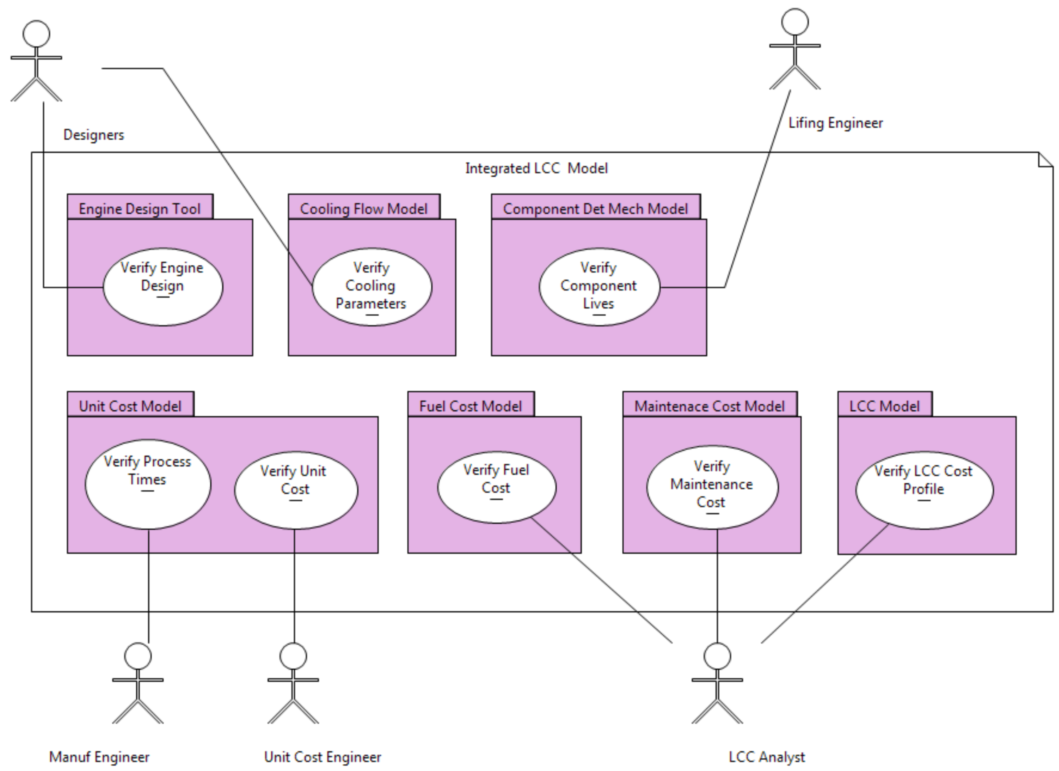


Figure 4-18: Use case diagram illustrating the verification of the integrated LCC model.

4.8. Discussion

The integrated LCC model was developed with consideration to the three characteristics identified in section 2.5.6. These characteristics were customisability, transparency, and modularity. The selection of Isight as the integration tool was influenced by these factors. The user interface in Isight, as seen in Figure 4-7, allows analysis modules to be added or removed without the need for complex scripts to be written. This made the integrated model easy to customise. This was demonstrated by adapting the LCC model used in Chapter 3 for the study in this chapter. The unit cost, maintenance cost and deterioration mechanism models were the same in both cases. The integrated LCC model was also kept modular with each analysis as an individual Isight component. This allows analyses models to be changed according to what is required, such as if the engine design model needed to be replaced with a tool which has greater functionality. Finally the user interface in Isight has functions which let users track and manage the data flows between the modules. This and the modular structure help to make the integrated model transparent to the end users.

The case study presented above is a simplified one. However, even with this simplification, the integrated system model developed for the study already possesses a certain degree of complexity. This complexity will increase when more components are added. This serves to highlight the need for complexity management when developing models for complex products such as an aero-engine. Capturing the different aspects of complexity in SysML allows for a more complete description of the model as compared to a document-centric approach. Doing this maintains transparency of the model which not only aids understanding and communication, but is also useful in developing an understanding of the results produced.

The integrated systems model developed for this study was tailored to study TET and cooling flow effects on LCC. If the objectives of the study change, the make-up of the integrated systems model will need to be modified as well. The knowledge recorded in the SysML models will help identify what changes will need to be made. This may seem a trivial task for the models presented in this thesis; however a more realistic analysis will include more components with more complex interactions. Additionally, the construction of the system model is not be the responsibility of an individual but will require the involvement of different stakeholders which necessitates the need for a standardised representation.

4.9. Summary

An integrated LCC model which is able to predict how LCC changes with TET and CFF was presented and discussed. It was also shown how MBSE was used in the model development. The results of the study showed that minimum LCC occurs when TET is high and cooling flow fraction is low. Although the case study was significantly simplified through the consideration of just the HP turbine blades, it exemplified the issues identified before in the literature review. In a more complete study, these issues will be magnified due to the increased size and complexity of the integrated system model. Ultimately this work demonstrates how MBSE can be applied to the LCC analysis process to provide greater structure and understanding in order to reduce model building time and effort.

5. Using MBSE in Value Driven Design

VDD is "an improved design process that uses requirements flexibility, formal optimization and a mathematical value model to balance performance, cost, schedule, and other measures important to the stakeholders to produce the best outcome possible" [126]. By this definition, there exists an overlap in the scopes that VDD and LCC cover. In the previous chapters, the use of LCC in design was discussed and demonstrated. This chapter compares and contrasts the VDD and LCC processes in aero-engine design. It also discusses how MBSE can be of benefit to VDD.

5.1. Introduction

In 2005, the initial AIAA VDD program committee was formed, motivated by cost and schedule overruns often seen in the design and development of large, complex systems [127]. The VDD program committee was officially launched in 2006 to promote the application of Value-Driven Design. It brought together three experts who each chaired the AIAA Economics, Systems Engineering and Multidisciplinary Optimisation Technical Committees [128]. Prior to the formation of the AIAA VDD program committee, various members from industry and academia had contributed to the ideas of VDD; these include Keeney [129], Hazelrigg [130], Brathwaite and Saleh [131]. An account of the history, ideologies and current methodologies of VDD is presented in Collopy and Hollingsworth [128].

The application of VDD is illustrated in Figure 5-1 which shows a cyclical view of the design process. At the *Design Variables* stage, designers parameterise the design of the system and pick a point in the design space. The set of design variables are then used to define the configuration of the system using the appropriate CAD tools. From this, the attributes of the system is estimated and modelled; often using physics based modelling tools. Where VDD differs from the usual systems engineering design process is in the *Evaluate* arc of the design loop. Systems engineering processes at this point would assess whether the attributes of the system has met requirements.

Once it has, the design cycle may be stopped. VDD instead uses the attributes of the system in an objective function or a value model to calculate a score. Designers may choose to change the configuration and continue on another iteration to determine if a higher score can be achieved.

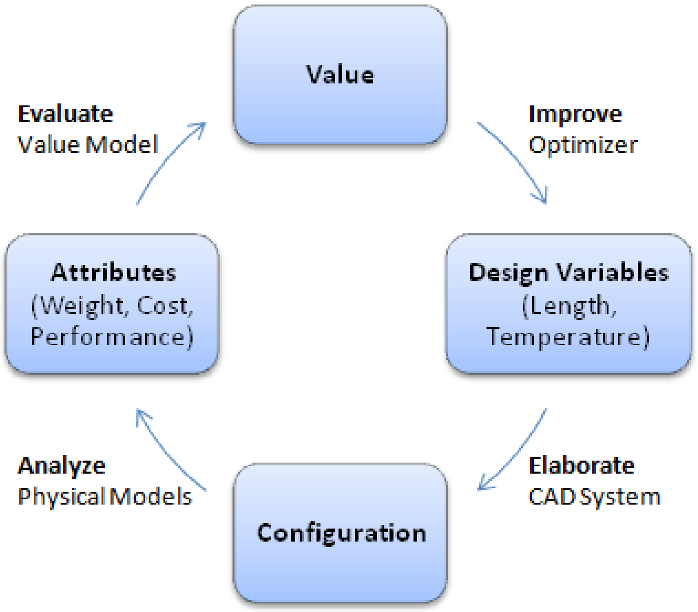


Figure 5-1: The Value Driven Design process; adapted from Collopy [128].

VDD also aims to change the way in which designers deal with extensive attributes. These are attributes of a complex system where the value of the attribute at the system level is a function of the values at the component level; some examples are weight and all aspects of cost. VDD considers requirements on extensive attributes to be a hindrance in the design of complex systems. In a VDD framework, no requirements are placed on extensive attributes at the system, sub-system or component level [128]. Hence, there is no flow-down of requirements, as practiced in traditional SE methodologies. Instead, engineering teams are assigned a scalar objective function, which converts design attributes (weight, performance, reliability) into a score. The ultimate goal is to maximise this score for the best design. Therefore, instead of flowing down requirements the aim is to flow down objective functions to each sub-system and components of the system, for which design teams need to maximise.

5.2. Representing Value

The concept of value, in the broadest sense of the word, has many definitions. In economics, value is the worth of goods or service as determined by the market. In the field of value engineering, value is defined by SAVE International[*] to be function divided by cost.

In the context of VDD, value conveys an order of preference. In other words, system A, S_A, is preferred to system B, S_B, if the value of system A, $v(S_A)$, is greater than that of B. Through establishing a consistent manner of assigning value to a system, designs can be objectively ranked according to which is better.

In several aerospace value models, value has been represented as:

- monetized summary attributes (pollutants, noise, fuel consumption, macroeconomic effect);
- performance divided by cost;
- an ordered pair of utility and cost (Keeney-Raiffa approach);
- surplus value: A profit calculation independent of pricing and product feature actions of competitors [14];

There is no restriction on what the final output of the value model is, however Collopy suggests that value should be represented as an ordinal number; which is capable of being ranked. As a result, the Keeney-Raiffa approach should be deemed unsuitable as ordered pairs cannot be ranked. The emphasis on the ability to rank the output values stems from VDD's aim to provide an objective function through which a design can be optimized.

As this thesis is focussed on linking cost with design, value models of an economic nature are the ones of interest. Expressing value in monetary terms is practical and convenient as there are usually cost elements in a value model, and it is a scale with which everyone is familiar and thus more easily understood.

[*]"What is Value Engineering" Webpage - http://www.value-eng.org/value_engineering.php

5.3. Value Driven Design Modelling Process for Aero-Engine Systems

To perform VDD, a system value model, which represents value as a function of the system's extensive attributes, needs to be constructed. This section gives a brief overview of the process. For a more detailed description, the reader is advised to consult Cheung et al [132]. Figure 5-2 outlines the steps in the VDD modelling process.

Process step 1, "Identify stakeholders of the system", begins with establishing a point of view of the model. The model used in this thesis uses the surplus value equation, which measures profitability, as the system value measure. The model hence needs to be developed from the view of key stakeholders who would affect the profitability of the system.

Step 2, "Build system value model" involves the selection of a measure of value, as discussed in section 5.2, and the system value equation parameters.

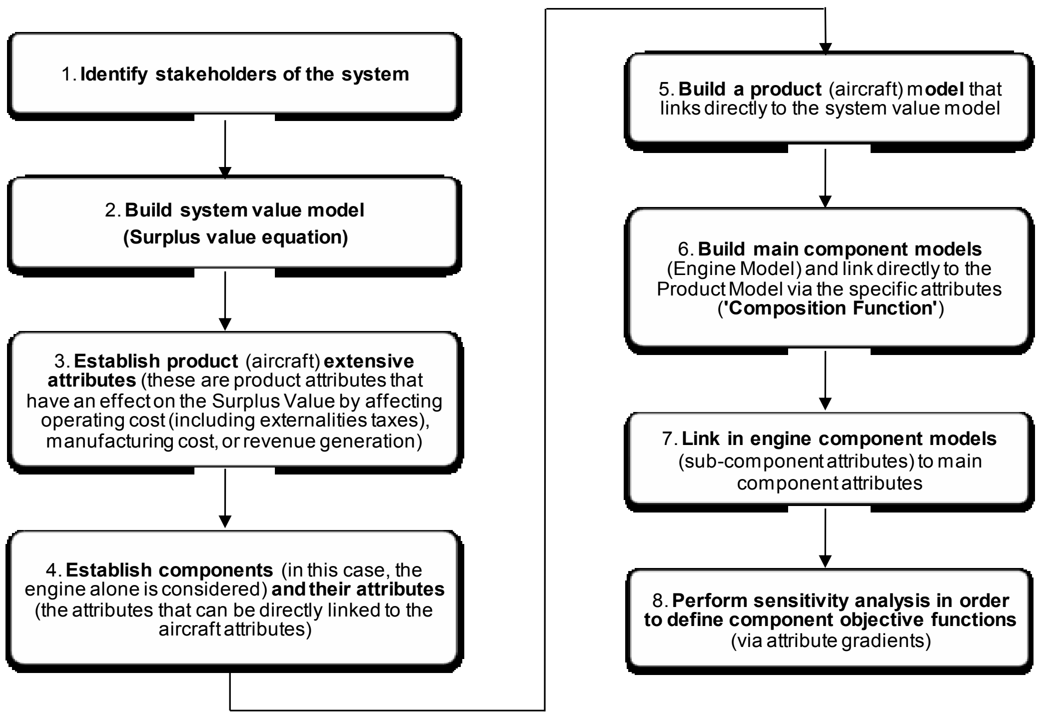


Figure 5-2: VDD modelling process for aero-engine systems [132].

Step 3, "Establish product extensive attributes", the relevant input product attributes are identified. Since surplus value is a function of revenue, operating costs, and manufacturing costs, the product attributes which need to be identified must be related to these three factors. Next, the attributes which have a natural economic relationship are grouped together. For example, reliability and maintainability can be combined into a maintenance cost component. Higher level components can also be formed; maintenance costs and fuel costs can be merged into operating cost component. Eventually, a hierarchy is formed with the system value at the root, as shown in Figure 5-3.

In step 4, this process of identifying and linking attributes is repeated again for the component, "Establish component extensive attributes". The end result from steps three and four will be a hierarchy which links system value, product attributes, and component attributes. In general, the system value model will be easier to work with if there are fewer top level components. The consolidation of top level components can be one of the challenging aspects in VDD modelling. This is because it is not always clear how to convert all the components into a single value of measure.

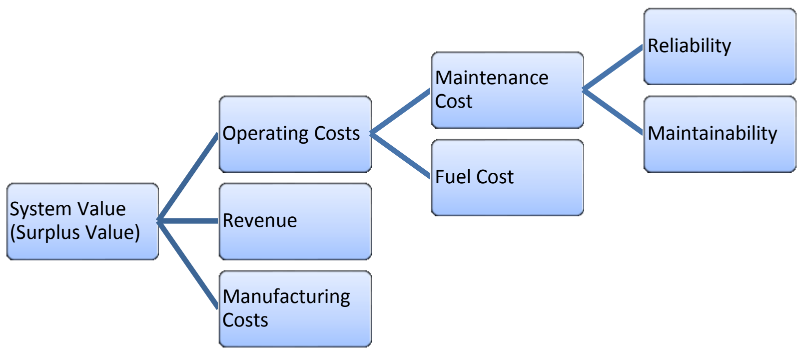


Figure 5-3: Identifying system attributes.

Steps 5 to 7 involve the integration and development of models which will model the relationships, identified in the previous process steps, between system value and the component attributes of interest.

In step 8, sensitivity analyses are performed to obtain an understanding of the relationships between input and output variables and to define the objective function for the component of interest. When the baseline design of the system has been fixed, the desired characteristics of the sub-systems and components can be established. In a normal systems engineering process, this would involve the flowing down of requirements and the division of extensive attributes such as weight and cost. Under the VDD framework, objective functions specific to the sub-system or component are flowed down instead.

5.4. Examples of VDD Models

VDD as a concept has only started to mature in the last decade. It was in 2006 that the AIAA VDD program committee was launched to promote the application of VDD. As a result, there is limited literature on the application of VDD in its current form. A review of some value models can be found in Collopy [14]. In this section, the focus is on the type of models used and how various sub-system models are integrated.

The DARPA System F6 program [133] involved the parallel development of four different value models for the design of a fractionated satellite. The four companies involved were Boeing, Lockheed Martin, Northrop Grumman, and Orbital Sciences Corporation which brought together a variety of tools and models which include component-level manufacturing cost models, development cost models, launch cost estimation, and post-launch system operation cost models. Table 2 gives an overview of the models developed.

Table 2: DARPA System F6 Model Overview

Team	Tool	Front-end	Back-end	Simulation Method
Lockheed Martin	SVM Tool	Matlab .m files	Matlab .m files	Markov chain, Monte Carlo
Northrop Grumman	SVM Tool	Excel	Matlab .m files	Discrete event simulation, Monte Carlo
Orbital Sciences Corporation	PIVOT	Matlab .m files	Matlab .m files	Discrete event simulation, Monte Carlo
Boeing	RAFTIMATE	Matlab .m files	Matlab .m files	Discrete event simulation, Monte Carlo

O'Neil et al [134] reviewed the capabilities of the tools and provided some insight into their characteristics. Firstly, it was found that two of the tools were only capable of running architectures and missions which were hard-coded in the tool and were consequently not able to perform broad or detailed design trades. Secondly, the tools had a steep learning curve with regards to their usage even for a knowledgeable spacecraft designer; which was further exacerbated by a lack of documentation. Thirdly, the four tools had different quantifications of value due to the vastly differing perceptions of value by the respective developers of the tools. Lastly, it was found that there existed a significant disparity amongst the four tools; included: modelling fidelity, technology inclusion/exclusion, mission considerations and input/output structures.

Castagne [135] developed a methodology which applies VDD to optimise the structural configuration of aircraft fuselage panels. The model consists of a variety of manufacturing cost equations, structural equations and objective functions implemented in Microsoft Excel. This approach is compared to past approaches which optimised the fuselage panels to minimum weight and minimum manufacturing cost. Significant changes to the fuselage configuration were observed when different objective functions were used. The authors concluded that it is crucial to consider what needs to be optimised in the design process and for whom.

Peoples and Willcox [136] developed a methodology that integrates an aircraft performance model with a program valuation technique based on real options theory to address uncertain market demand and managerial flexibility. The models used in their study included Wing-MOD, which is a multi-disciplinary optimisation code used to estimate aircraft sizing and performance characteristics. The outputs were then used by cost and revenue models to estimate cost, price, and baseline demand figures for the design. The valuation module uses a stochastic dynamic programming algorithm, which accounts for market growth and uncertainty, to determine a set of optimal design decisions and the objective; expected Net Present Value.

5.5. VDD Modelling Issues

The previous chapter discussed some of the modelling issues associated with the development of LCC models. System Value, like LCC, is a system attribute and is thus prone to the same modelling challenges. The following sections consider whether VDD modelling involve similar issues.

5.5.1. Application Specific

The value models reviewed above were found to be unique in their application. A value model has to be constructed from a certain point of view. For example, the value models applied in military or emergency services would be significantly different to those applied in a commercial context as the former would not have sources of revenue. Therefore different value models have to be used to suit the system being modelled.

As mentioned above, composition functions are created to extend the value model to component design. Each of these composition functions is unique as they relate the specific attributes of the component under study to the system value. With a complex product or system, a number of different composition functions will be required.

5.5.2. Modelling Approach

Modelling approaches seen in VDD models varied from using analytical equations to integrated simulation environments. This is largely dependent on the tools and system information available. In Collopy [128], the use of surrogate models was put forward as a promising method in system model development. The advantages highlighted were that surrogate models would decrease the computational effort needed to evaluate the computational effort required in calculating the system value and that the models would be "more or less" continuously differentiable. This last point is desirable as it would allow gradient based optimisers to operate on the value model and its sub-models. It should be noted however that there are limitations to the use of surrogate models. First of which is that a large number of training points may be needed to construct an accurate surrogate, especially for a large number of input values and highly non-linear input-output relationships. It is recommended that current surrogate modelling techniques can be used for between 10 to 20 design variables. Secondly, this approach is a black-box approach which means that they are constructed for specific cases and that they obscure relationships between inputs and outputs. These hidden relationships are of detriment to the discovery of cause and effect factors which is necessary in design.

5.5.3. Model Complexity

The review of VDD models revealed that the range of tools and models used was just as varied as those used in LCC. In actuality VDD modelling will require an even more diverse set than that of LCC because it involves more aspects of the system than just cost. As will be discussed later, the optimum design for an aerospace engine is the one which optimises profit for the airline. This results in the need for airline operation and aircraft models which will be used to relate engine attributes to those of the system value model.

5.5.4. Model Transparency

If LCC models face transparency issues, the issues with VDD models will be even more severe. A larger scope will mean a larger integration of models from various disciplines. VDD aims to be applied throughout the different design stages (from concept to detailed design) and from system to component design. Transparency must be maintained so as to be able to allow the rationalization of results.

5.6. MBSE Enabled VDD Aero-Engine Modelling Process

From the sections above, it becomes clear that VDD modelling face similar issues to that of LCC. Just as it was with the LCC case study in the previous chapter, VDD can benefit from the use of MBSE. A process was devised which merges VDD with the OOSEM methodology. Figure 5-4 illustrates the points in the VDD modelling process where the OOSEM MBSE activities are performed. The following case study shows this process in more detail.

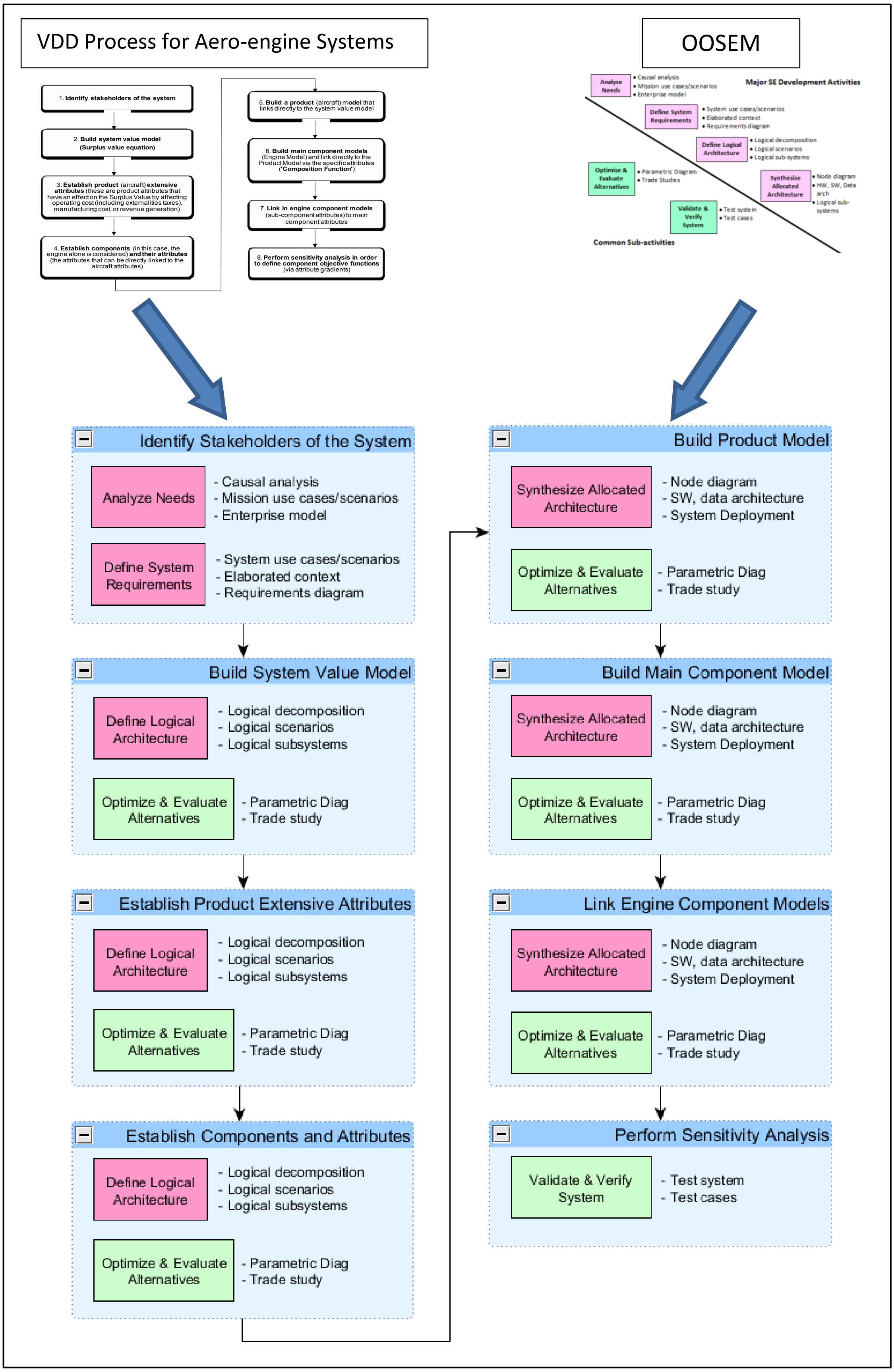


Figure 5-4: Using MBSE in the VDD modelling process.

5.7. Case study: Impact of Turbine Entry Temperature and cooling flow on Surplus Value

To enable a comparison between designing for optimum system value and designing for minimum LCC, the same case study in chapter 4 was performed. The main objective was to analyse how TET and cooling flow fraction affected the system value for a single aircraft. Data used for the models was based on the IAE V2500-A1 engine for the Airbus A320 aircraft. To keep the scope of the study manageable, only the effects of the change to the High Pressure (HP) turbine stages were modelled.

The main difference in the VDD case study was that the calculation of the system value measure, surplus value, considered the aspects of the airline and aircraft. To reiterate what was mentioned in section 5.5.3, the best engine design is the one which generates the most profit for the airline. As illustrated in Figure 5-5, increasing the profit generated by the airline is of benefit to all involved parties; airline, aircraft and engine manufacturers. A larger profit pie means that there is more to be distributed. It is ultimately the market which decides how the profit is divided between the three companies.

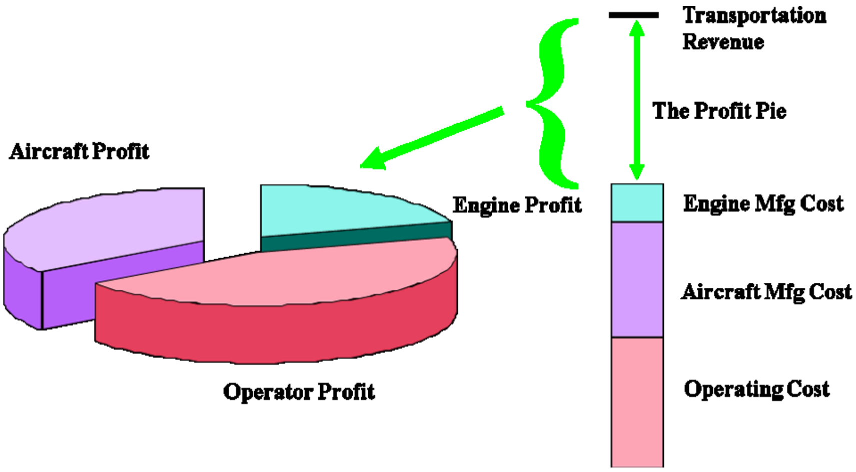


Figure 5-5: The profit pie; taken from [137].

Airline surplus value was used as the measure of value for the system value model. Surplus value was used instead of profit as the former does not depend on the pricing and product feature actions of competitors.

5.7.1. Identify Stakeholders of the System

The key stakeholders are the ones who can influence the system value. In this case, they were identified to be the airline, aircraft manufacturer and engine manufacturer. The other stakeholders are the ones who will have direct input into the VDD model, such as design, manufacturing and cost engineers. Figure 5-6 shows the context in which the VDD model was to be used and what roles the stakeholders had. Due the information and data available, most of the detailed analyses were performed on the engine. The models which link the engine attributes to the aircraft and airline attributes used crude “rule of thumb” approximations developed by several subject matter experts [132]. The purpose of the use case diagram was to help identify what functions the VDD model needed to perform. This helped to compose the requirements for the VDD model, shown in Figure 5-7. The top level requirement for the VDD model was decomposed into categories (System Value, Airline, Aircraft, and Engine) to represent the major stakeholders involved. Figure 5-7 shows the decomposition of the engine requirements to describe the required engine modelling activities.

5.7.2. Build System Value Model

The system value model was constructed using Surplus Value theory; which is based on the profit calculation for industrial products. In general, Surplus Value is the difference between total revenue (incoming cash flow) and total costs incurred (outgoing cash flow) discounted to the number of years of the programme. To keep model development simple, this equation effectively views the different companies involved as a single entity. This ignores the realities of competition between the different airlines, aircraft manufacturers and engine manufacturers.

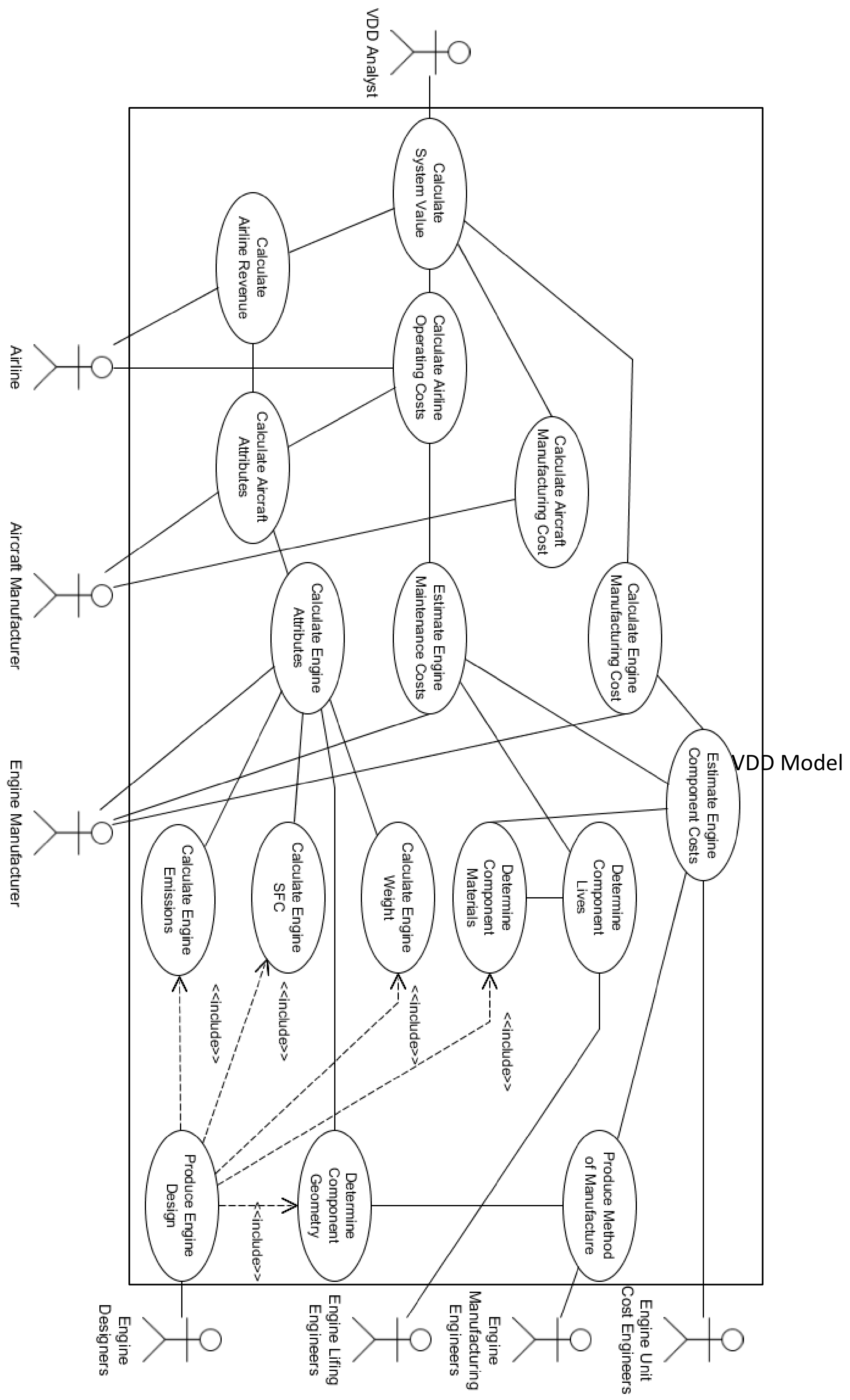


Figure 5-6: VDD model use case diagram.

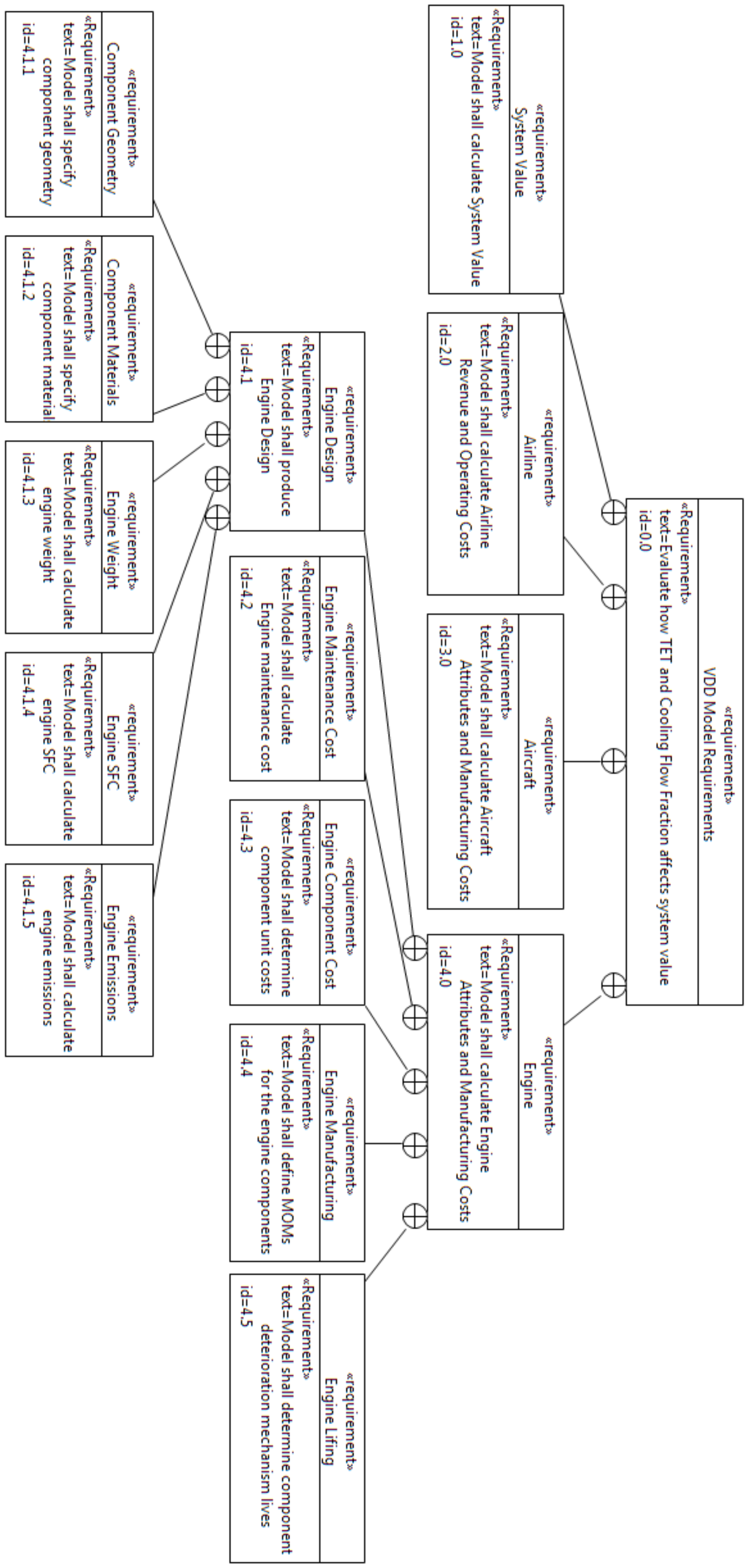


Figure 5-7: Requirements for VDD model.

As shown by Collopy [137], the surplus value for an aircraft in operation is the total profit of the airline, aircraft manufacturer and engine manufacturer. This is calculated by:

$$Surplus\ value = Discount\ Factor * (Airline\ revenue - Operating\ costs - Aircraft\ and\ engine\ manufacturing\ costs), \quad (5.1)$$

The final surplus value equation eventually became more complex and is shown in Appendix A implemented within the Vanguard Studio software. This derivation was formulated during VDD workshop conducted by RR [138]. How the surplus value equation evolves depends on the system being modelled. Airline revenue for instance was split into *Passenger Revenue* and *Cargo Revenue*. *Passenger Revenue* was then linked to the *Number of Seats* of the aircraft and the *Stage Length* of the particular flight. *Cargo Revenue* was related to *Stage Length* and to the *Aircraft Cargo Capacity*. The surplus value equation in the case study was used to relate the profitability of the airline to the attributes of the aircraft, and finally to the engine attributes as well. The subsequent steps in the modelling process focused on linking the design of the engine to this equation and maximising surplus value.

5.7.3. Establish Product and Component Attributes

As it is assumed that the best engine design is one which optimises profit for the airline, the VDD model needs to incorporate relationships between aircraft operations, aircraft, engine and the engine component as illustrated in Figure 5-8. The determination of which relationships and attributes to be modelled requires the input of the relevant stakeholders and domain experts. In a VDD workshop [132], a brainstorming session was performed to identify the various system attributes. Figure 5-9 shows the tables created during the VDD workshop. Starting with the surplus value equation above, the terms of the equation were then decomposed into attributes which were relevant to those of the aircraft. Examples of these attributes are payload, range, fuel burn, maintenance cost, and manufacturing cost.

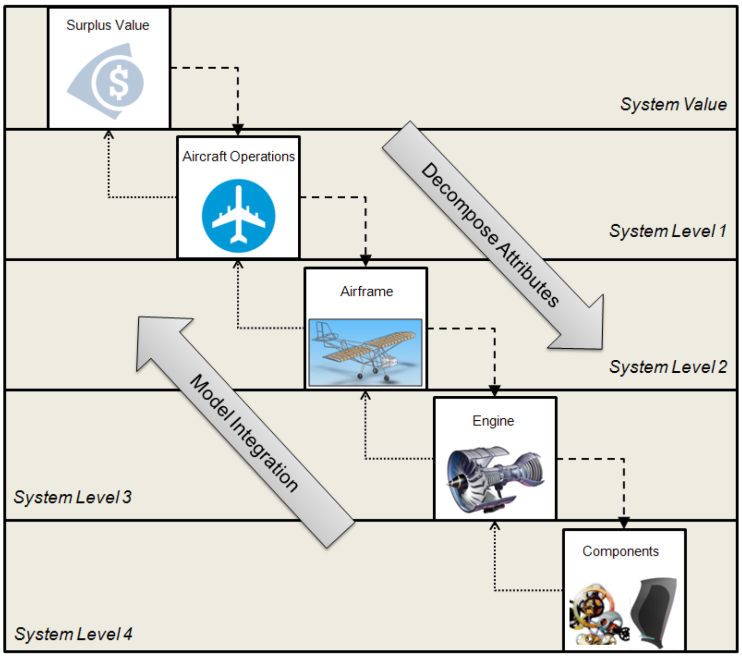


Figure 5-8: System hierarchy [132].

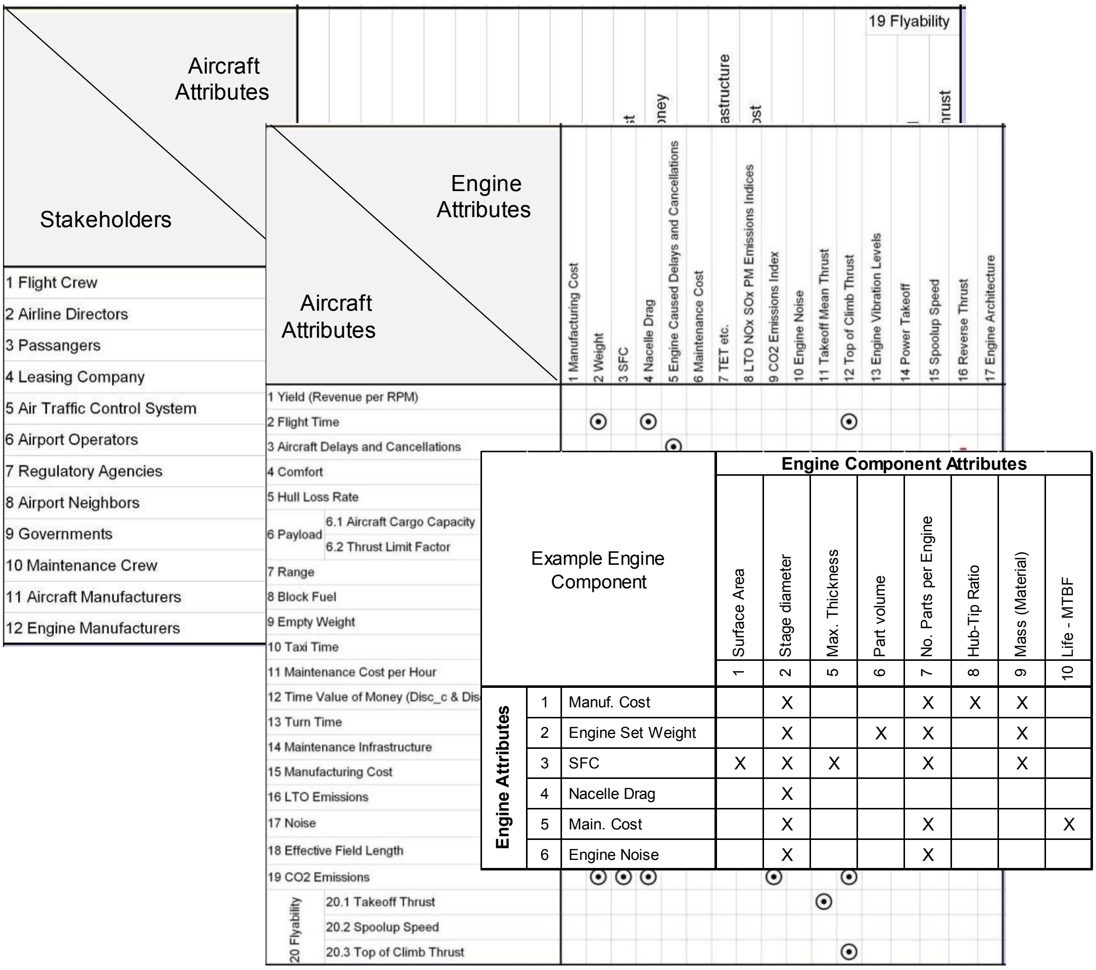


Figure 5-9: Tables of system attributes [132].

Following that, the attributes of the aircraft which are related to those of the engine were identified and similarly, the engine attributes were linked to those of the component of interest. A table was created for each system level, stakeholders to aircraft, aircraft to engine, and finally engine to engine component. The system attributes identified in the brainstorming session were then collated and used to form the logical architecture of the VDD model, shown in Figure 5-10. This diagram captures the functions that the VDD model must perform to satisfy the requirements. It shows that the VDD model has four main categories; System Value, Airline, Aircraft, and Engine. The functions of the VDD model are then sorted according to the relevant categories.

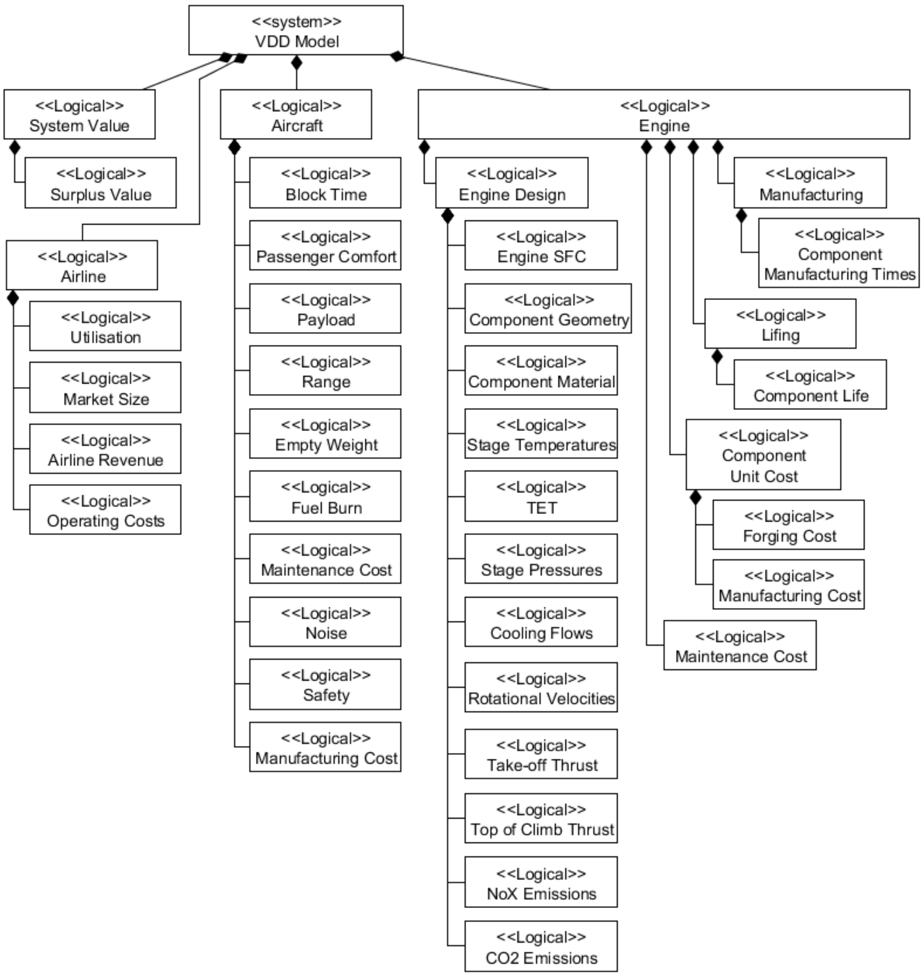


Figure 5-10: Logical architecture of VDD model.

5.7.4. Build and Integrate Models

This section covers steps 5 to 7 of the VDD aero-engine modelling process; which are "Build product model", "Build main component models", and "Link in engine component models". These process steps were grouped together because the development of models needed to consider how the individual models were to be integrated. In the previous process step, the functions required for VDD modelling were identified. These functions were then allocated to analyses models that were to be produced. Figure 5-11 shows the allocated architecture diagram of the system value, airline, and aircraft models, indicated by the shaded boxes. On the left of the diagram, are inputs from the engine model. These inputs connect to the aircraft, airline, and system value models to calculate surplus value, shown on the right of the diagram.

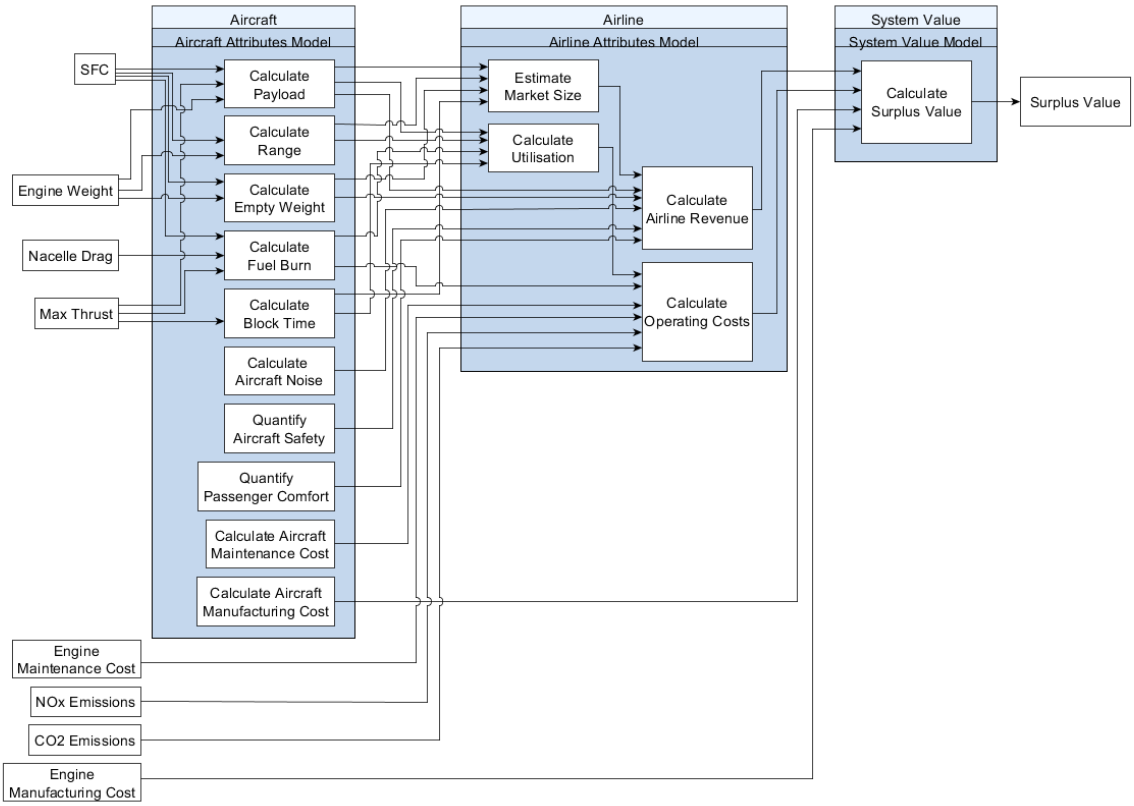


Figure 5-11: Allocated architecture diagram for system value, airline, and aircraft models.

Figure 5-12 shows the allocated architecture diagram for the engine model, with the analyses models indicated by the swim-lanes. It should be noted that the model outputs from the top half of the diagram feed into the models at the bottom half. At the top left of the diagram are the main inputs of interest for the study, TET and cooling flow fraction. These connect to the engine design model and subsequently to the other analyses models to provide the inputs, at the bottom right of the diagram, for the aircraft model in Figure 5-11. These allocated architecture diagrams also help to show the data flows and relationships between the models. With the VDD model architecture and data flows established, the analyses models could then be constructed. These included a variety of modelling tools and techniques such as Excel spread sheets, hierarchical tree models, surrogate models and discrete event simulations. Sections 5.7.4.1 to 5.7.4.5 describe the models in more detail.

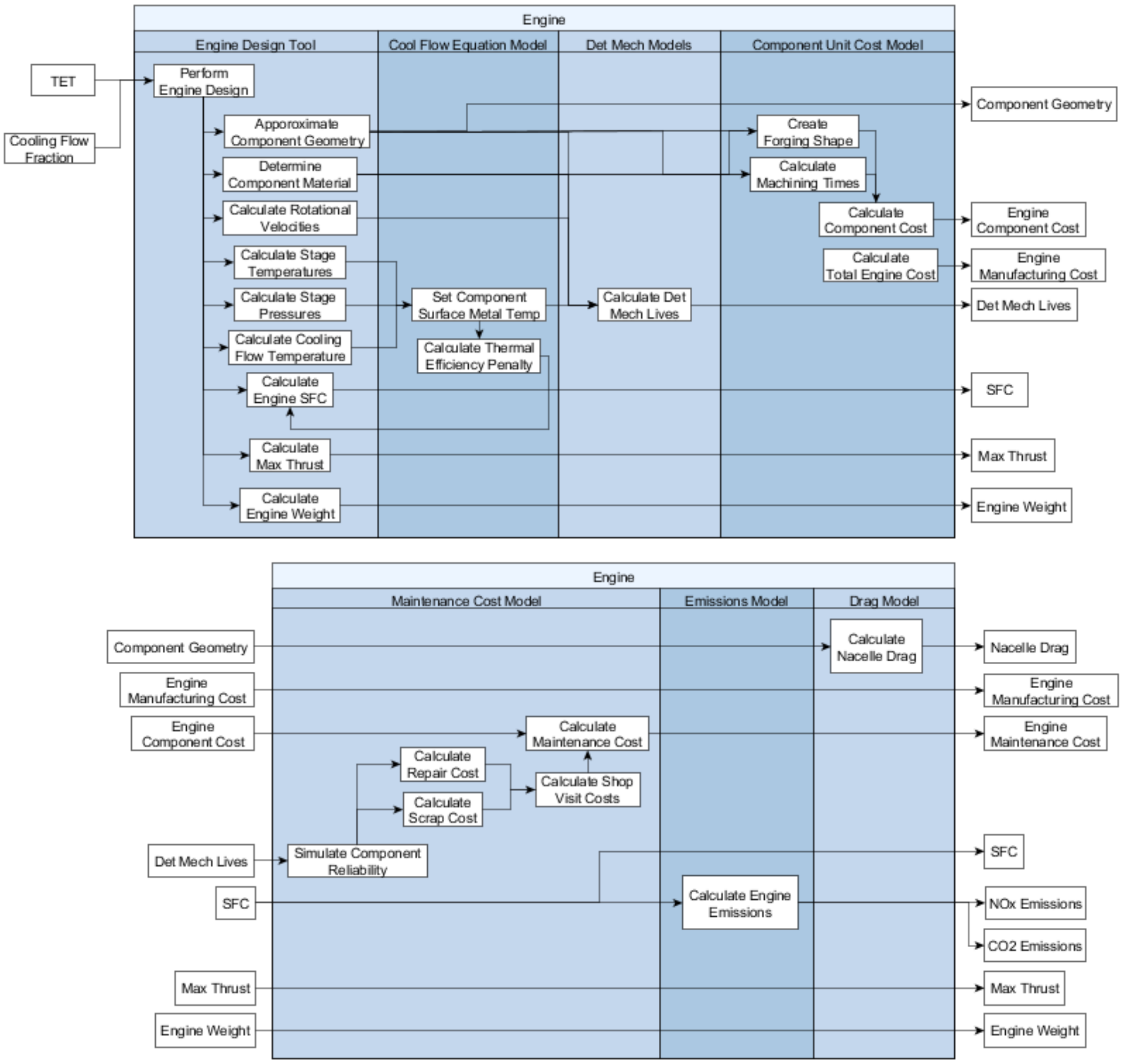


Figure 5-12: Allocated architecture diagram for engine model.

The implementation of the integrated VDD model was done in Isight, just as it was done in the LCC study (see section 4.5).

Figure 5-13 shows the integrated model in Isight, where the swim-lanes in Figure 5-12 correspond to the Isight components in the main process flow. To reiterate, the system value, airline, and aircraft models were developed in a single Vanguard Studio model. The Isight process flow ran the various analyses in the sequence that they were positioned in the allocated architecture diagrams. Many of the models in this study were used in the LCC study and thus the execution of the models was as described in 4.5. The analyses models that were added were the Emissions, Engine Drag, and the combined System Value, Airline and Aircraft models. In Figure 5-13, the boxes around the engine design spread-sheet, deterioration mechanism life models, FLOPS, and the System Value, Airline, and Aircraft model indicate that these tools were not developed as part of this research.

5.7.4.1. System Value, Airline and Aircraft Model

As mentioned earlier, due to the amount of data and resources available, the system value, airline and aircraft models were created from equations which were based on experience and information from open sources and implemented in a single Vanguard Studio model. Figure 5-14 shows a screenshot of this model with the system value (measured in surplus value) node and the related inputs shown. This model was developed using data available on the Airbus A320 aircraft and equations used in preliminary aircraft design. The rest of the model is described in Appendix A.

5.7.4.2. Engine Models Reused from LCC Study

As was mentioned earlier, some of the models required coincided with those used in the LCC case study. The models which were reused were the:

- Engine Design Model (Section 4.5.1)
- Cooling Flow Model (Section 4.5.2)
- Fuel Cost Model (Section 4.5.3)
- Deterioration Mechanism Models (Section 4.5.4)
- Maintenance Cost Model (Section 4.5.5)

- Unit Cost Model (Section 4.5.6)

Additional models required for the VDD study are described in the following sections.

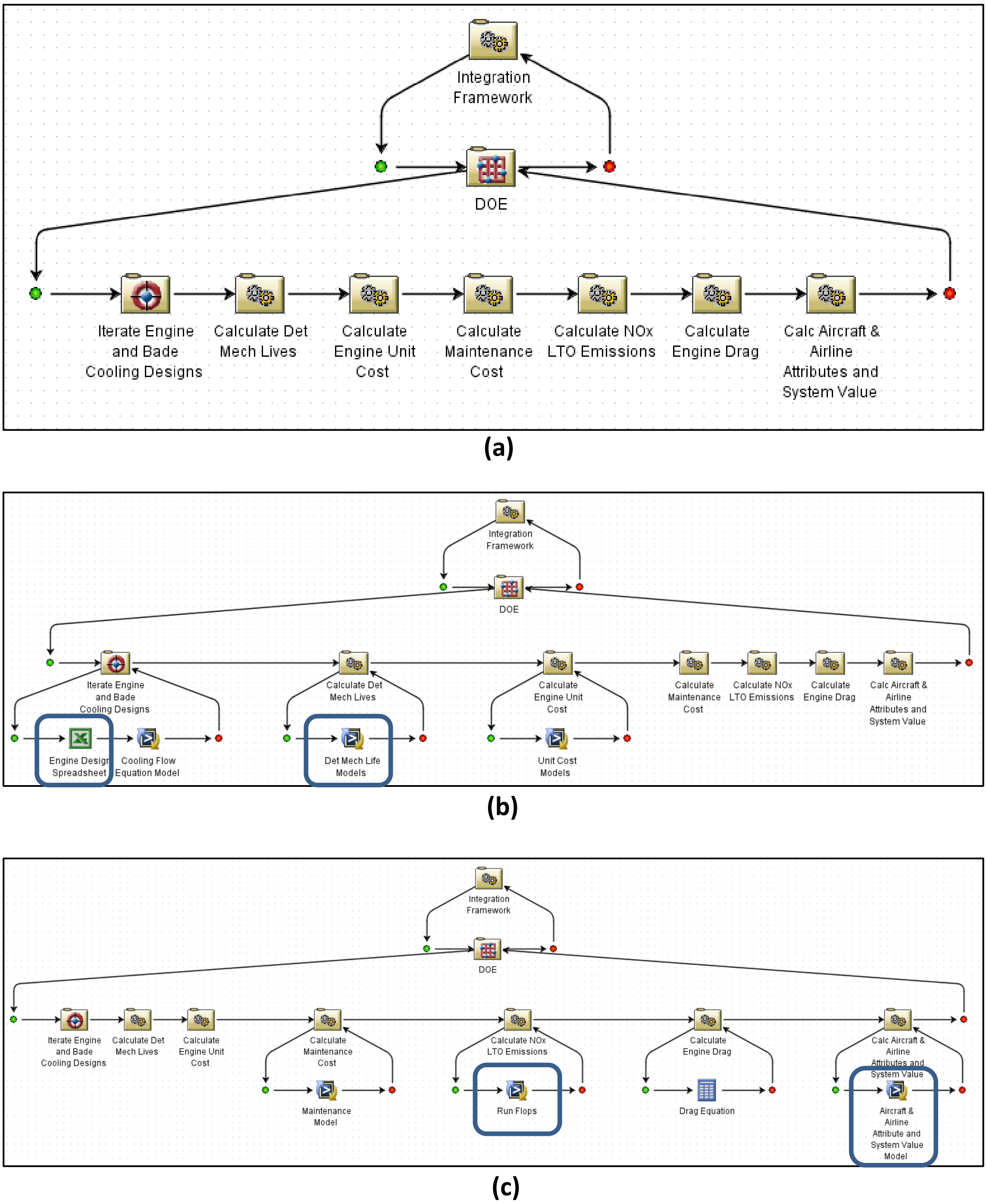


Figure 5-13: Integration of models in Isight-FD. (a) Workflow nodes collapsed. (b) and (c) Workflow nodes expanded.

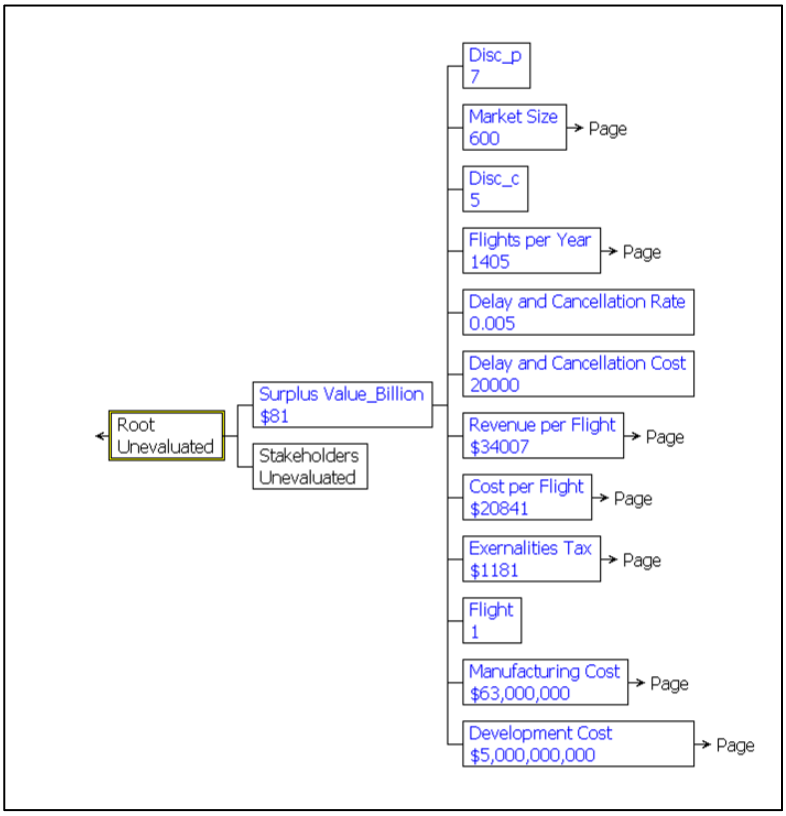


Figure 5-14: Screenshot of System Value, Airline and Aircraft model.

5.7.4.3. Emissions

The TET of the engine is linked to NO_x and CO_2 emissions as TET controls thermal efficiency and SFC. The calculation of total emissions produced is also dependent on the specific flight profile. The relationship of emissions with TET was modelled using the Flight Optimization System (FLOPS) [139] developed by NASA. FLOPS is a multidisciplinary system of computer programs for conceptual and preliminary design and evaluation of advanced aircraft concepts. It consists of nine primary modules: weights, aerodynamics, engine cycle analysis, propulsion data scaling and interpolation, mission performance, take-off and landing, noise footprint, cost analysis, and program control.

5.7.4.4. Nacelle Drag

For turbo-fan engines, it is typical to assume that only the outermost surfaces of the nacelle contribute to pure aerodynamic drag. Any drag induced within the engine gas path is regarded as thrust loss. Nacelle drag can hence be taken to be directly proportional to the fan stage diameter. An empirical formula taken from Jenkinson [140] was used to estimate nacelle drag. This is given as:

$$D_n = \frac{1}{2}\rho V^2 C_d S, \qquad (5.2)$$

where ρ is the air density, V is the flight velocity, C_d is the coefficient of drag and S is the wetted area of the nacelle.

5.7.4.5. Engine Weight

Engine weight was calculated using the same spread sheet model for the engine cycle design seen in section 4.5.1. As the spread sheet was able to produce estimates of the engine geometry, such as fan diameter and turbine blade size, an approximated weight could be arrived at given the densities of the material used for the various components.

5.7.5. Results and Discussions

As in the LCC study, the range of input values for the LCC and VDD models were 1700 to 2000 for TET and 0.1 to 0.3 for the cooling flow fraction. A 10 level full factorial design was used to generate the response surfaces. Figure 5-15 shows the relationship of surplus value against TET and total cooling fraction. The optimum system value was found to be US$357.05 billion at a TET of 2000K and cooling flow fraction of 0.155. This is close to the optimum point seen in the LCC study which was at a TET of 2000K and cooling flow fraction of 0.191. A sensitivity study, shown in Table 3, was performed using Vanguard Studio to analyse how the engine attributes affect system value. Partial derivatives of surplus value with respect to each engine attribute were taken. The partial derivatives were calculated for ± 20% change in engine attribute. From the results it was determined that the dominant factor is the SFC of the engine.

Further examination of the equations defined in the system value model revealed that the influence of SFC was greater in the VDD study than it was for the LCC study. The effect of SFC on aircraft attributes was analysed and the results showed that SFC had a significant effect on three aircraft attributes. These were: fuel cost per flight, payload capability and emissions tax. As such, a reduced SFC not only provided the benefit of lower fuel burn, it also gave the aircraft a higher payload which increased the amount of revenue. Additionally, lower fuel burn reduced emissions which had a direct influence on the amount of taxes paid.

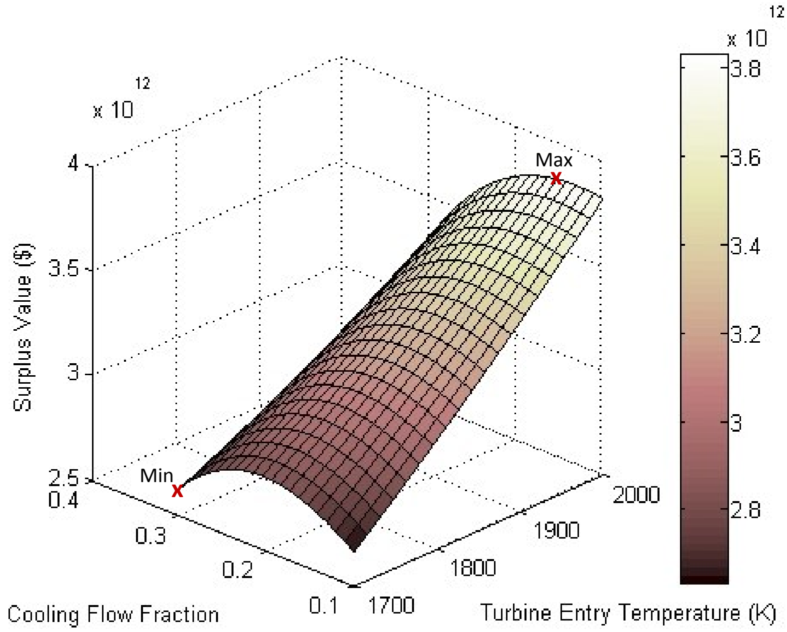


Figure 5-15: Surplus value vs. turbine entry temperature and cooling flow fraction.

These results, like the LCC results, suggested that it would be beneficial to design the engine for higher TET values. The next logical step might have been to explore how the surface behaves beyond 2000K however the engine cycle design program was unable to arrive at a solution in that region for the desired thrust and flight speed. The FLOPS program, used for the calculation of emissions, was also not stable for high TET values. Furthermore, the low cycle fatigue life analysis tool, SC03, could not generate sensible results for the required conditions. The pursuit of accuracy for this case study would not be sensible as the scope it covers, despite the significant simplification, is still large and the knowledge required is more than a single person can cope with. As with the LCC case study results, discussed above, it would be practically impossible to validate the surplus value results of the VDD study. Especially in this case which linked the profitability of the airline, aircraft manufacturer and, engine manufacturer. The mitigating action was to have the results of the individual models verified and endorsed by subject matter experts at Rolls-Royce. Figure 5-16 shows the subject matter experts and the corresponding models which were verified.

Table 3: Sensitivity Study of Engine Attributes on Surplus Value.

Engine Attributes	Relative Gradient	Absolute Gradient
SFC (lb/lbf·h)	-0.91921	-465.30822
Maintenance Cost (US$)	-0.28487	-0.00006
Engine Weight (lb)	-0.24206	-0.01264
Engine Manufacturing Cost (US$)	-0.18283	-0.00001
Maximum Thrust (lbf)	-0.00821	-0.00009
TET (K)	-0.00438	-0.00077
Emissions (g/kN)	-0.00324	-0.01768
Engine OPR	0.00023	0.00211

The VDD model was developed in consideration of the desired characteristics discussed in section 2.5.6 (i.e., customisability, transparency, and modularity). Just like the LCC model developed in the previous chapters, the use of the Isight integration tool provided transparency and facilitated customisation. The studies performed in this thesis demonstrated the ability to increase the complexity of the integrated models. Each study added new modular analyses models to adapt to the changing study objectives. The use of MBSE also gave structure to the development of the integrated model by helping to link the model requirements to the final model architecture. What this case study ultimately demonstrated was a methodology for performing VDD for a complex product like an aerospace engine and how MBSE could be used to manage the complexities involved in the process.

5.8. Comparison of VDD and LCC approaches

LCC and VDD approaches are fundamentally different. LCC is concerned with the cost that the system of interest has incurred over its life time. VDD on the other hand seeks a singular metric which encompass all important aspects of the system design; which is to say VDD concerns much more than just cost. However when VDD utilises a monetary measure such as surplus value, it includes all the elements of LCC. This means much of the effort expended in modelling LCC can be used in a VDD analysis. Both LCC and VDD can require complex system models depending on the system of interest, but as the scope of VDD is larger than that of LCC, the complexities regarding cost model development are more severe for VDD than they are for LCC.

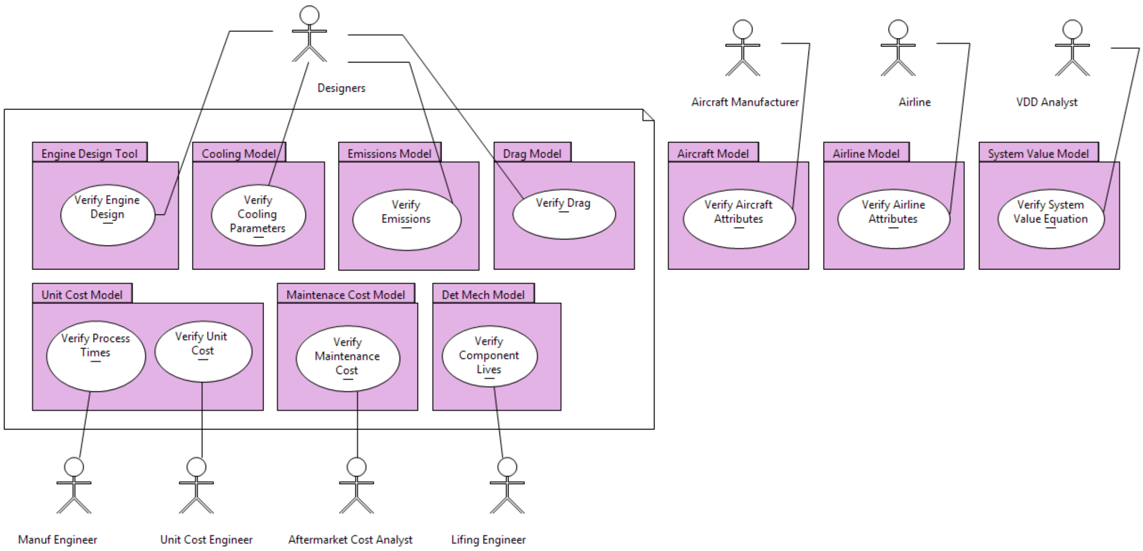


Figure 5-16: Use case diagram illustrating the verification of the VDD model.

In the VDD case study of an aero-engine, its extensive attributes are linked to those of the airframe to study its impact on the operations of an airline. Such a holistic approach is desirable as an aero-engine needs to be designed in conjunction with the airframe and aircraft operations. An anecdote by Cumpsty [121] illustrates this point. When the Boeing 777 was designed, three engine manufacturers were in contention; General Electric (GE), Rolls-Royce (RR) and Pratt & Whitney (PW). GE chose a bypass ratio (BPR) of 9 for their engine, the GE90, while RR and PW selected a BPR of 6. In service data for these three engines showed that the GE 90 had the lowest specific fuel consumption (SFC), but only by 1% lower than the other two. The GE90 also turned out to be the heaviest, about 3 tonnes heavier than the lightest, which is the RR Trent engine. As a result, aircraft using the lightest engines had 6% extra payload or could operate with a greater maximum range. In spite of this, the GE90 did prove successful in giving significant lower nose levels since the higher BPR allows for lower fan tips speeds and jet velocities. This led to later engines designed with a higher BPR to meet tightening noise restrictions.

The account above demonstrates the need for an approach such as VDD to enable designers to consider the various aspects of aero-engine design. Furthermore, Cumpsty [121] states that the goal of commercial aero-engine design is to produce an engine which will generate the most profit; which surplus value theory allows us to analyse as part of the objective function. However, there are some limitations to this approach. In value modelling, it can be a challenge to convert the various elements of the model into a singular measure of worth. In using surplus value for the model, all the components must be converted into monetary units. This is not easy to do for intangible factors such as reputation, safety or charm. Model transparency can also become an issue when a model is complex, as was discussed for LCC models. The scope of a VDD model is much larger than a LCC model thus making it even more complex than the latter. This will make it difficult to understand and justify why the model behaves the way it does consequently affecting confidence in the results produced. Additionally, the system value metric cannot be validated or calibrated against any actual historical data. The only validation that can be done is to verify the makeup of the model to ensure that it is logical. Lastly, it must also be acknowledged that VDD as compared to LCC is a relatively new concept. Therefore VDD standards and best practices have yet to be established.

There are clear benefits in the use of the VDD approach shown above in the design of complex systems as opposed to using LCC. In addition to providing a framework for the design optimization of the entire system, the use of surplus value aligns the design focus to the main objective of the company; which is to be profitable. This however does not negate the use of LCC. The case study presented above treats the airline, aircraft manufacturer and engine manufacturer as a single entity based on the premise that larger profit for the airline is only of benefit to all involved parties. With this arrangement, it is important for the individual companies to monitor how the system value optimal design affects their respective LCCs to ensure that the design is not beyond budgetary constraints.

5.9. Summary

This chapter discusses the VDD process to compare and contrast it with the design for LCC approach used in Chapter 4. It was found that VDD provides a better measure of design for commercial aero-engines as it considers all aspects of the engine as well as knock-on effects that engine design has on aircraft and airline attributes. VDD also aligns the engine design process to the objectives of the company, which is to produce a profitable engine. It was also concluded that the VDD modelling process suffers from issues similar to those of LCC modelling and can benefit from the use of MBSE. A case study which shows how system value changes with TET and cooling flow fraction was performed to demonstrate the MBSE methodology. The optimum point was found to be similar to that of the LCC study as SFC was found to have a significant effect on aircraft and airline attributes.

6. Conclusions & Future Work

6.1. Research Summary and Contributions

With the growth of leasing arrangements, OEMs need to consider the costs of their product over its lifecycle in order to remain profitable. Tools and processes are hence needed for designers to assess the impact of their design choices on the life of their products. From a review of whole life cost tools for aero-engine design, ideal characteristics for the tools were proposed. These were customisability, modularity, and transparency. Two different approaches of relating design to cost were also considered. The first is the more established route of designing for low LCC. The second is the novel and more holistic approach of VDD. The points below highlight the main contributions of the thesis:

- A methodology was developed with the proposed characteristics which linked component design to LCC.
- A more complex LCC study was performed which studied how TET and CFF affected LCC. This study demonstrated the customisability of the integrated model described in the previous contribution by being able to adapt it for new study objectives.
- A VDD study was performed for changes in TET and CFF so that it could be compared against the LCC approach. Merits and faults were discussed for the LCC and VDD approaches.
- The OOSEM methodology was used within the LCC and VDD processes to demonstrate how MBSE could be used to manage the development of the integrated models.

The sections below describe the work done in greater detail.

6.1.1. Determine desirable characteristics for aero-engine LCC design tools.

In Chapter 2, Life Cycle Cost was described to be the total cost associated with the acquisition and ownership of a product or system over its full life. The concept of LCC has since been the subject of much research, as shown in the reviewed literature. There were several limitations and shortcomings with the models reviewed. Firstly, the LCC models were often found to be application specific and developed for specific phases of a product life cycle. Secondly, LCC models varied in modelling approaches dependent on objectives, scope, level-of-detail, results desired and available resources. Thirdly, model complexity can become an issue as causal relationships between costs drivers and life cycle cost can be complex. System properties emerge based on the relationships between the involved sub-systems. Modelling these system properties may require multi-disciplinary analyses with complex interfaces, interactions and dependencies. Finally, model transparency needs to be maintained as much as possible to ensure that LCC models can be more easily understood, debugged and maintained. The recognition of these issues with current LCC models was used to develop desirable characteristics for a LCC design tool. These characteristics are customisability, modularity and transparency.

6.1.2. Develop a methodology to link aero-engine design to LCC

A LCC model was developed in consideration of the desirable characteristics identified. The model was used to assess the impact of changes of geometry of an engine component on LCC. A commercial software integration package to link three different modelling paradigms, hierarchical tree unit cost models, response surface deterioration mechanism life models, and a DES maintenance cost model. This modular approach taken to building the model ensures that LCC model can be customised to the specific study objectives. The use of the software integration package also keeps the model transparent through its representations of the integrated modules and the management of the connections between the modules.

6.1.3. Incorporate MBSE into the LCC process

Model Based Systems Engineering (MBSE) is an approach which uses modelling in the specification, design, integration, validation, and operation of a system. It aims to improve complexity management, integration and communication among system/project stakeholders. A study was performed to discover how MBSE could be of benefit to the LCC process. It was found that MBSE can be used to capture knowledge about the design of a LCC model. MBSE was used in capturing requirements for the models and then linking them to the logical and functional architecture of the integrated model. This helped to identify the different analyses modules required and map the data flows between them. MBSE also provides a consistent language that can be used across different system domains. This was important as LCC models can involve complex multi-disciplinary models.

A case study was performed which looked at how LCC changes with Turbine Entry Temperature (TET) and Cooling Flow Fraction (CFF). It was used to illustrate the use of MBSE in the development of a LCC model. This study also demonstrated the customisability of the integrated model as it extended the capability of the model described in section 6.1.2. The results of the study showed that minimum LCC, US$1.0859 billion, occurs when TET is at 2000K and cooling flow fraction is 0.191.

Validation of the integrated LCC model was difficult as actual data illustrating the relationship between LCC and the inputs, TET and CFF, for a specific engine does not exist and would be logistically unfeasible to generate. This is especially so for LCC as the life of an aero-engine typically spans decades, one shop visit interval is typically around 20,000 hours. To mitigate this, the individual component models and the results of the integrated model were verified and validated by subject matter experts at Rolls-Royce. Ultimately, relative LCC results are sufficient as an indicator for design selection.

6.1.4. Incorporate MBSE into the VDD process

Value Driven Design (VDD) is a design philosophy which uses requirements flexibility, formal optimization and a mathematical value model to balance performance, cost, schedule, and other measures important to the stakeholders to produce the best outcome possible. System value, which is the main output of a VDD analysis, is an emergent system attribute. This means that system value is dependent on the behaviour of the various sub-systems it encompasses. It was concluded that the modelling issues identified in LCC modelling were similar to that of VDD modelling. The case study used to illustrate its application links the design of an aero-engine component, to engine performance, to aircraft performance, and finally to the profit of the airline. With so many factors to consider, the scope of the VDD model must be tailored to the objectives of the study to prevent the VDD model from becoming too large and complex. VDD models will thus tend to be application specific. VDD models were also seen to be varied in modelling approaches; from analytical equations to discrete event simulations. The model is inherently complex due to the number of relationships and interfaces between the sub-system attributes.

As VDD and LCC face similar modelling issues, a MBSE approach to VDD modelling can have the same benefits that MBSE has for LCC modelling. As in the LCC study, MBSE was used to capture requirements for the models, and generate the logical and functional architecture of the integrated model. A parallel case study to that of the LCC study was performed. In this case, it looked at how changes in TET and CFF affect system value. The design for the optimum system value was found to be US$357.05 billion at a TET of 2000K and CFF of 0.155. This was similar to that in the LCC study as the Specific Fuel Consumption (SFC) had a significant effect on aircraft and airline attributes. This meant that the effect of SFC on the VDD model was magnified in comparison to the LCC model, which only considers the effects on the engine. As before, the difficulty in validating the results meant that they needed to be verified and endorsed by the relevant subject matter experts.

In this case, both the LCC and VDD models have the same conclusion. A case study where the VDD and LCC models might have differed is an open rotor engine compared to a turbo-fan. Open rotor engines are more fuel efficient, but are noisier and slower than turbo-fans. As a result, the LCC of the open rotor engine would be improved but, depending on how the airline values engine noise and flight times, the system value for the airline might worsen.

6.1.5. Comparison of LCC and VDD Approaches

LCC and VDD approach system design in different ways. LCC will form part of the requirements that a system must satisfy; as in a traditional design process. VDD on the other hand seeks requirement flexibility by incorporating all the measures important to the stakeholder into a singular value of measure. A VDD analysis thus incorporates all elements of LCC. VDD's holistic approach to system/product design is more beneficial as it can determine how profitable the design is rather than just how much cost the system/product incurs. However VDD does have several issues. Firstly, it can be challenging to convert all the attributes of a system into a single measure of value. Secondly, the complexity of a VDD model can cloud understanding of the model leading to difficulties in rationalising results. This is significant especially as the system value cannot be validated and depends on expert judgement to ensure that the estimates are sound. Lastly, VDD is a relatively new concept, therefore implementations are rare and standards have yet to be established.

Ultimately, if VDD can be implemented correctly, it will prove to be an invaluable design approach. It provides a framework for different design attributes to be converted into a singular indicator of design merit and can be used as the objective function to optimise the system as a whole. Regardless of this, LCC remains a useful tool. A system can be designed for lowest system value, but there must still be an awareness of the cost of ownership for cash flow management. In the case study presented in section 5.7, it is assumed that airline, airframe manufacturer and engine manufacturer are one single company. In practice, there will be negotiations about how the final profit will be distributed. The cost of ownership will undoubtedly affect the outcome of how the profit will be split.

6.2. Future Work

6.2.1. Value Driven Design

The benefits of a VDD approach have been highlighted and discussed in this thesis. It has also been mentioned that VDD is a relatively new concept and thus there are a number of avenues for future research. This has been comprehensively discussed by Soban, Hollingsworth and Price [141]. In their paper, five research areas for VDD were identified.

These are:

- System;
- Stakeholders;
- Value Function;
- Finding the Best Value;
- Identifying the Enablers.

In the first area, the System, Soban et al, ponder the various ways of defining the system to be assessed. Correctly defining the system is of critical importance as VDD hinges on optimising the value of the system as a whole. There can be various sub-system boundaries that need to be defined in a study; as was seen in the case study in chapter 5. System boundaries can also be political, organisational, or physical. Additionally, there is the question of how the system can be decomposed (physical, functional, or otherwise) and whether all system elements need to be at the same level of fidelity. Finally there is also a need to find ways of identifying interrelationships between critical sub-systems.

The second research area centres around issues regarding the stakeholders involved in the VDD analysis. Stakeholders are considered to be the people who define what the systems objectives, constraints, design drivers and success criteria are. There are uncertainties regarding who the stakeholders in the VDD system should be. This is important as the involved stakeholders directly influence the definition of value.

The third research area regards the definition of the value function. In this area, the first issue to be resolved is how value should be defined. In section 5.2, the various ways of representing value was discussed. While the merits for a monetary measure was put forward, there are disadvantages it to using it as well. The next issue concerns who the people involved in defining it should be. The stakeholders alone may not have all the knowledge about the system thus other experts, such as designers and manufacturing engineers, could be considered as well. It is also challenging to correctly define sub-system value functions. Some sub-system attributes can be difficult to connect to the system value function; such as reputation.

The forth research area asks questions about how the best system value can be found. This includes issues regarding modelling strategy; whether component models should be mathematical models, physics based models or empirical models. Robust design philosophies could also be considered to ensure stable solutions. It was also noted that as VDD promotes flexible requirements to expand the design space, a completely open design space may be detrimental too. Thus methods for initial scoping of the design space are required.

The final area of proposed research is in the identification of enablers in VDD to realise the philosophy into practical applications. In this respect, there are numerous engineering methodologies and processes that VDD can leverage upon. Systems engineering (SE) is a straight forward field to explore as it shares many aspects with VDD. Practices in Product Lifecycle Management (PLM) and Concurrent Engineering should also be considered. Concurrent engineering aims to consider all aspects of a system lifecycle (functionality, producibility, assembly, testability, maintenance issues, and disposal and recycling) at the design stage; this coincides with the VDD imperative to develop an objective function which inputs all the important attributes of the system being designed. PLM, on the other hand, is a process which manages the entire lifecycle of a product. It helps to integrate people, data, processes and business systems to provide a product information backbone for companies and their extended enterprise. This centralisation of data, processes and people will benefit VDD in terms of cross functional integration.

6.2.2. Using MBSE in LCC and VDD

The MBSE approach to modelling LCC and VDD was proposed as a way to manage the development of the integrated analyses models. The work that has been done in this thesis regarding MBSE however has been experimental. The LCC and VDD case studies presented in this thesis have been created with simplified and approximated models. The next phase of the research should thus be to apply MBSE to a project which more accurately represents the difficulties and complexities of designing an aero-engine. The project should involve cross-functional teams which Include members from various disciplines involved in the process; such as stress, life prediction, thermal, geometry and manufacturing. Ideally, the project should be trialled in the early concept design phase of the aero-engine to limit the amount of design detail that has to be considered.

There are other developments in MBSE which will facilitate the development of VDD and LCC models. The work by Peak et al [13], mentioned earlier in section 2.6.4, has spawned a spin-off company called Intercax [142] which provides standard and custom software applications for SysML-driven modelling. This includes integration of SysML with CAD/E/M and PLM software. Further work with SysML should utilise the integration tools as an enabler for large-scale, multi-disciplinary and complex system-design projects.

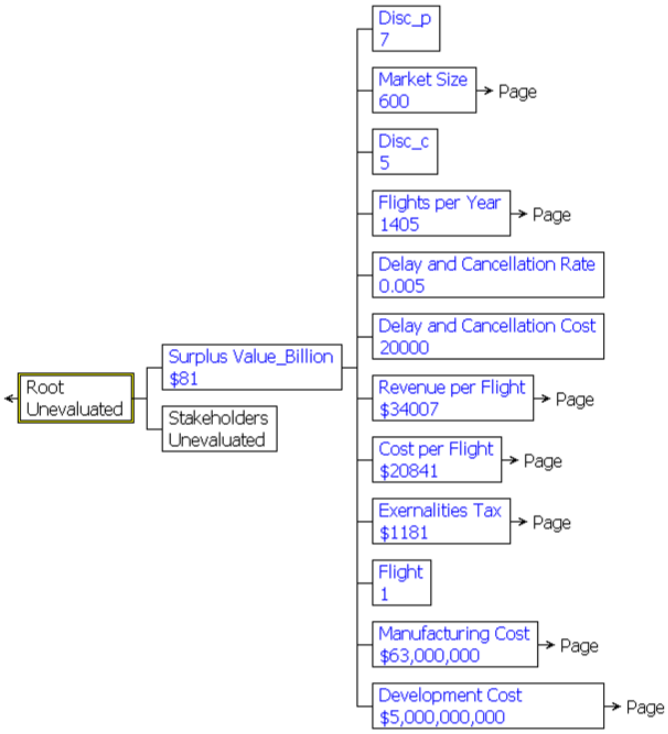
As mentioned in section 2.6.3.2, domain-specific-languages (DSL) is an approach whereby experts perform MBSE in a modelling language tailored to their particular domain with the facility to link the domain specific models into a coherent system model. The first method for DSL consolidation is ontology-based mapping of the relevant DSL meta-models and the second is via the use of a language workbench. Language workbenches are integrated development environments which are able to handle multiple domain-specific languages. DSLs will benefit multi-disciplinary VDD and LCC analyses as they will allow domain experts in the integrated product team to work in modelling languages that they are accustomed to rather than using a common MBSE modelling language with predefined semantics and syntax.

Appendix A

Taken from "Advanced Cost Modelling Methodologies Value Driven Design – Scoping and Feasibility Study" by Collopy et. al. [138]

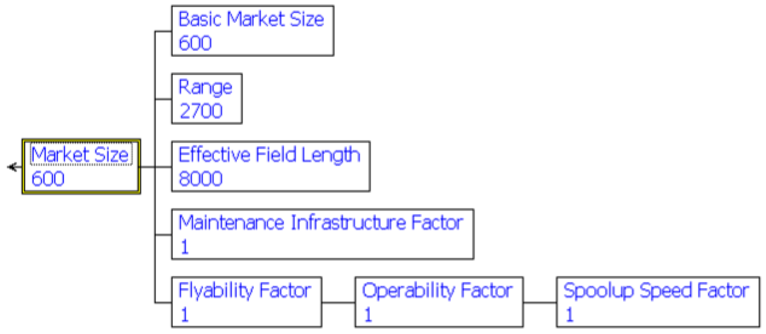
Vanguard Surplus Value and Product (Aircraft) Models

Aircraft Model Page 1

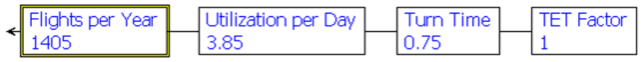


Note that delay and cancellation rate and cost should strictly be included as a contributor to cost per flight and the model will be updated such that this is the case.

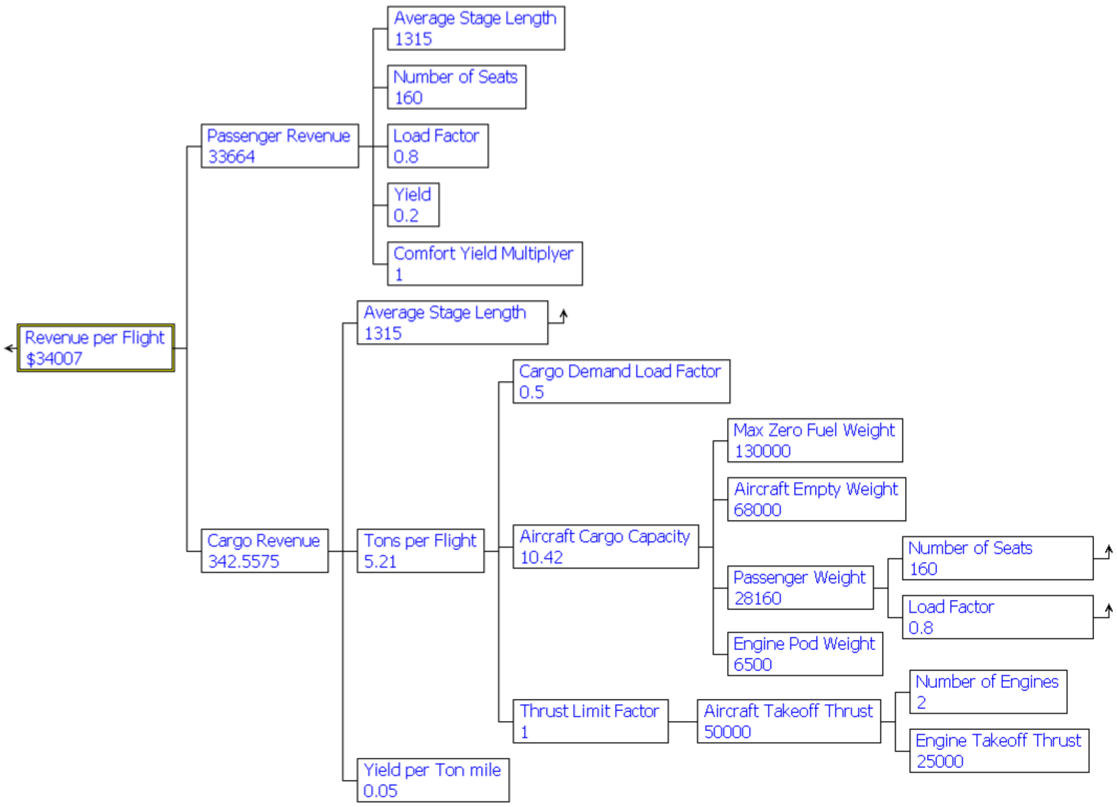
Aircraft Model Page 2



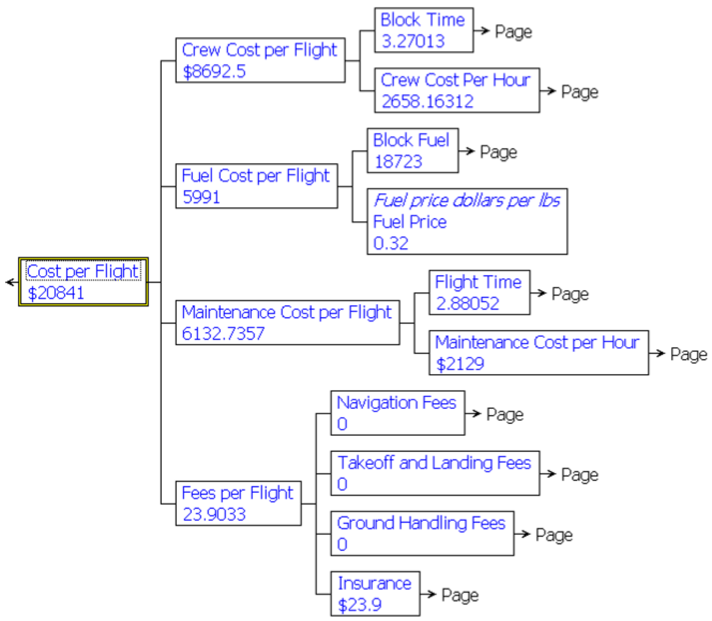
Aircraft Model Page 3



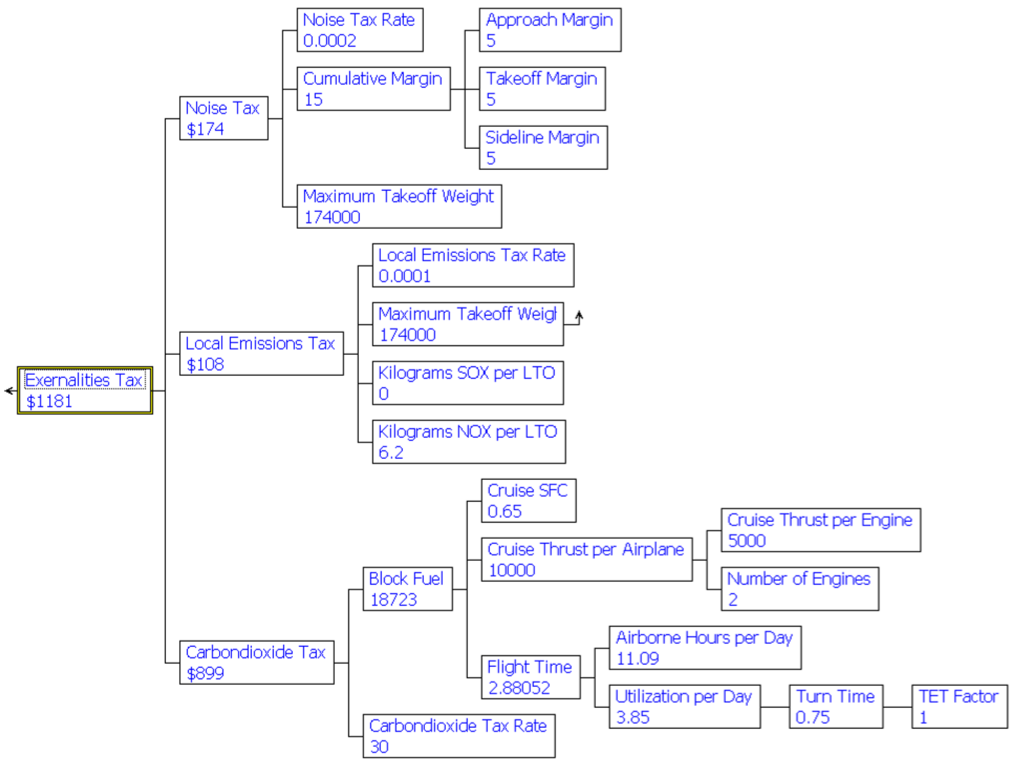
Aircraft Model Page 4



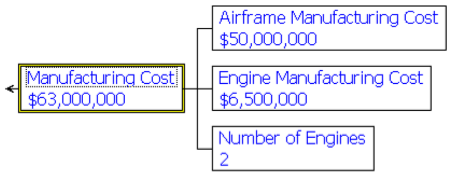
Aircraft Model Page 5



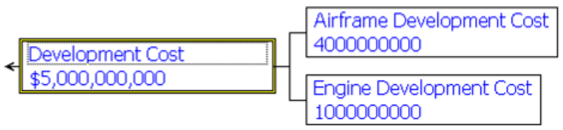
Aircraft Model Page 6



Aircraft Model Page 7



Aircraft Model Page 8



Product (Aircraft) Model Inputs and Equations

Revenue Generation

The principal components of revenue are passenger and cargo revenue. The model currently assumes that passenger revenue is preferred. This assumption more generally holds true for the 100-200 mid-range single aisle aircraft. As such, revenue generation focuses on maximising passenger lift before adding cargo lift. The key components of revenue equations are:

- demand,
- yield,
- passenger mile/kilometre ($/RPM or $/RPK),
- cargo ton mile/kilometre ($/RTM or $/RTK),
- lift capacity, and
- route.

In the current model demand, yield, and route structure are assumed independent of aircraft and engine attributes, i.e. they are exogenous values. These assumptions only partially hold in reality, but in the first order they are appropriate. They can also be varied to determine surplus value sensitivity.

The lift capacity is modelled as dependent upon both aircraft and engine attributes. The first of these are airframe weight limits. These include:

- Operational Empty Weight (OEW),
- Maximum Zero Fuel Weight (MZFW), and
- Maximum Take-Off Weight (MTOW).

It is assumed that the maximum landing weight (MLW) is higher than the maximum zero fuel weight meaning that the maximum potential payload (MPLW) is limited by the lesser of the difference between MZFW and OEW or the difference between MTOW - the required Fuel Weight (FW) for a mission and OEW. This is given in:

$$W_{MPL} = \min[W_{MZF} - W_{OE} W_{MTO} - (W_F + W_{OE})]$$

where W_{OE} (OEW) is:

$$W_{OE} = W_{airframe} + W_{fittings} W_{engine\,pod}$$

This allows the engine pod weight to vary independently of the remainder of the airframe weight. As such it does not include any pylon or wing weight build-ups that are caused by changing engine pod weights. In the current form of the model the MTOW limit has not yet been implemented.

The effective payload can be further limited by operational limits on available take-off and landing field lengths. This is a function of both the route and performance of the airframe and engine combination. Currently no specific route information is included; instead the airports are treated in aggregate. The result of this is the model does not currently limit payload by landing field length; it does, however, have a limiter placed on the system for take-off filed length. This is in the form of an available thrust factor. This is implemented as a function of a ratio between the nominal thrust, ideally determined by aircraft specific properties, and the actual engine thrust. The current equation is linear, but will be updated to a logistic form:

$$Factor_{Thrust\,Limit} = \frac{1}{1 + e^{\left[-\left(\frac{F_{actual}}{F_{nominal}}\right)z + b\right]}}$$

This produces a sigmoid curve that asymptotically approaches 1 as the thrust ratio is increase, and zero as the thrust ratio decreases. By changing the relationship between *z* and *b* the steepness and location of the thrust factor drop can be controlled.

The logistic form is useful because it does not allow payload to increase beyond that of the MPLW for a mission.

Future developments will explicitly calculate take-off payload limits by incorporating aircraft properties such as:

- wing loading
- C_{Lmax}
- $L/D_{Takeoff\_}$

The specific calculation of revenue is then performed using the form:

$$Rev = \left(Yield_{pax} \cdot Num_{seats} \cdot LF_{passenger} \cdot Length_{avg\ stage} \cdot Factor_{comfort}\right) + \left(Yield_{pax} \cdot LF_{passenger} \cdot Length_{avg\ stage} \cdot Factor_{thrust\ limit} \cdot W_{cargo\ capacity}\right)$$

Cargo capacity is determined by the following equation:

$$W_{CC} = W_{MPL} - \left(W_{per\ passenger} \cdot Num_{seats} \cdot LF_{passenger}\right)$$

Cost Determination

The cost of the operating and aircraft in service is determined by two primary components, the operating costs that are directly related to the aircraft and the externalities taxes associated with the negative effects of operating the aircraft on surrounding communities and society at large. The aircraft cost determination is further split into:

- crew cost,
- fuel cost,
- maintenance cost, and
- fees.

The crew costs are a function of the number of crew, the cost of the crew and the block time for each flight. These in turn are determined by:

- aircraft seats,
- cruise speed, and
- stage length: average distance flown per aircraft departure.

Currently the block time and crew cost per hour are not calculated directly but this will be updated in a future version.

The fuel cost is determined by the engine SFC, thrust required, and time in each mode of the flight and the cost of fuel. Nominally this includes taxi-out, take-off, climb, cruise, descent, landing, and taxi-in plus any hold times. For the pilot version it was assumed that the cruise fuel-burn would dominate the mission, the average stage length is such that cruise fuel-burn is dominant. As such the fuel costs are approximated by the following equation:

$$W_{Fuel} = SFC_{Cruise} \cdot F_{Required\ cruse} \cdot Time_{Flight}$$

where the flight time is determined by aircraft cruise speed/Mach, and the thrust required for cruise is determined by:

- aircraft wing loading,
- aircraft drag polar,
- aircraft cruise speed/Mach no., and
- aircraft cruise altitude.

These are a combination of both aircraft and aircraft-operational attributes. In the current implementation a cruise thrust is external specified and not internally calculated.

The final fuel cost is determined using the following equation:

$$Cost_{Fuel} = W_{Fuel} \cdot Cost_{Fuel\ per\ pound}$$

The maintenance cost is split into both engine and airframe contributions. In the case of airframe maintenance cost, an assumed cost per flight hour is used; the nominal value is based upon the Alaskan Airlines data. The number of flight hours is again based upon aircraft cruise speed/Mach.

The final component of operating cost is the operational fees. In this model noise and emissions surcharges are not included in these fees as they are handled by the externalities taxes. These fees can be broken down into the following components:

- navigation fees,
- take-off and landing fees,
- ground handling fees, and

- insurance.

Navigation and take-off fees are typically charged as functions of aircraft MTOW. Landing fees are either based upon MTOW or MLW. In the case of this model the take-off and landing fees are combined and based upon MTOW. The navigation fee, as implemented, is:

$$Fee_{Navigation} = Rate_{Navigation} \cdot W_{MTO}$$

The take-off and landing fee is calculated:

$$Fee_{Take-off\ landing} = Rate_{Take-off\ landing} \cdot W_{MTO}$$

Ground handling fees are often a component of aircraft size, e.g. MTOW and the number of passengers carried. This is implemented as:

$$Fee_{Ground} = Rate_{Ground} \cdot W_{MTO} + Rate_{Passenger\ Handling} \cdot Num_{Seats} \cdot LF_{Passenger}$$

Cargo handling fees are ignored.

The final component of fees is the insurance cost. This is based upon a hull loss and fatalities rate and the aircraft and engine acquisition cost. The aircraft acquisition cost is considered and external input.

Externalities Taxes

The externalities tax, as implemented in the current value model, is a relatively simple model that approximates the current standards as used by airports and governments. It is based upon the margin to the current ICAO Chapter 4 noise limits, ICAO LTO Emissions, and CO_2 generated.

The current noise tax scheme is based upon cumulative margin, a basic rate, and MTWO as given in:

$$Tax_{Noise} = W_{MTO} \cdot (20 - Margin_{Cummulative}) \cdot Rate_{Noise\ Tax}$$

This provides a rebate for cases where aircraft have a greater than 20 dB margin. This will correct in the future.

The LTO emissions tax is modelled from the tax that is implemented at BAA's London airports, which is based upon regulated LTO emissions and a tax rate.

$$Tax_{Emissions} = \sum (Rate_{Emitant} \cdot W_{Emitant\ LTO})$$

Currently only NO_X rates are typically implemented, but all regulated local emissions can be taxed.

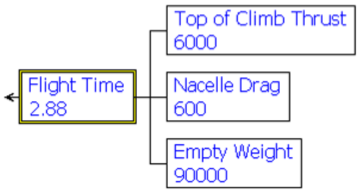
The final tax is based upon CO_2 emissions. This is a function of the block fuel, the CO_2 content of the fuel and a tax rate.

Ideally, the externalities taxes would be based upon actual impact, and not externally imposed rates. Future versions of the model will include an impact base approximation for the taxes; however, its use will be optional as the future regulator environment is uncertain. Additionally, the noise tax calculation will be updated to include the option of using actual certification values in place of margin for calculation of the noise tax.

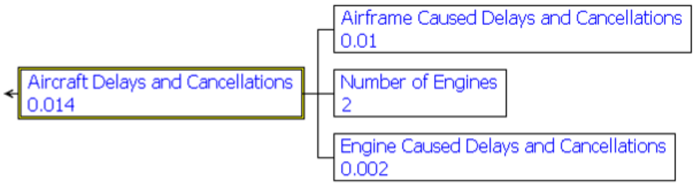
Vanguard Component (Engine) Models

Engine Model Page 1

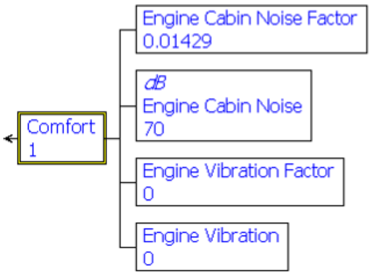
Engine Model Page 2



Engine Model Page 3



Engine Model Page 4



Component Model (Engine) Inputs and Equations

The engine model is currently populated with a limited number of input values and equations for:

- flight time,
- aircraft delays and cancellations (D&C),
- comfort, and

- hull loss rate.

Remaining parameters require additional equations and where we aim to illustrate 'drilling down' into more detailed analysis for comparative studies such detailed models (these need not be Vanguard models) will be incorporated.

The equations used to model the four attributes above are listed:

$$Flight\ Time = 3.36 - 8 \times \frac{Top\ of\ Climb\ Thrust - Nacelle\ Drag}{Empty\ Weight}$$

(formulated with use of the 'fudge factor', 8, to result in the same figure as the flight time derived in the aircraft model)

$$D\&C = AD\&C + E \times ED\&C$$

Where:

$D\&C$ = Aircraft Delays and Cancellations

$AD\&C$ = Airframe Caused Delays and Cancellations

E = Number of Engines

$ED\&C$ = Engine Caused Delays and Cancellations

$$Comfort = 2.0 - CNF \times (CN - VF) \times V$$

Where:

CNF = Engine Cabin Noise Factor

CN = Engine Cabin Noise

VF = Engine Vibration Factor

V = Engine Vibration

Hull Loss Rate - There is not currently an equation for this attribute. The figure is extracted from the aircraft model.

Engine Caused Delays and Cancellations - There is not currently an equation for this attribute.

References

[1] Harrison, A. "Design for Service - Harmonising Product Design with a Services Strategy," *Proceedings of GT2006, ASME Turbo Expo: Power for Land, Sea and Air*. Barcelona, Spain, 2006.

[2] Derber, A., Fendt, M., and Horwitz, M. "Why TotalCare can work for every Rolls-Royce Customer," *The Engine Yearbook 2007*. Aviation Industry Press, 2007, pp. 14-19.

[3] Anders, J. M., Haarmeyer, J., and Heukenkamp, H. "A Parametric Blade Design System." Von Karman Institute for Fluid Dynamics, 2008.

[4] Asiedu, Y., and Gu, P. "Product Life Cycle Cost Analysis: State of the Art Review," *International Journal of Production Research* Vol. 36, No. 4, 883-908, 1998.

[5] Gibbs, R. E. "Life-cycle cost modelling of military aero-engines in Rolls-Royce," *Seminar on Life Cycle Costs*. Professional Engineering Publishing Ltd, Heslington, England, 1998, pp. 31-44.

[6] Barton, J., Love, D., and Taylor, G. "Design determines 70% of cost? A review of implications for design evaluation," *Journal of Engineering Design* Vol. 12, No. 1, 2001, pp. 47-58.

[7] Fabrycky, W. J., and Blanchard, B. S. "Life-Cycle Cost and Economic Analysis." Prentice Hall, 1991.

[8] Keane, A. J., and Nair, P. B. "Computational Approaches for Aerospace Design: The Pursuit of Excellence." Wiley, 2005, p. 602.

[9] Tammineni, S. V.,"Designer Driven Cost Modelling", Ph.D. dissertation, University of Southampton, 2007.

[10] Gupta, Y. P. "Life cycle cost models and associated uncertainties," *Electronic Systems Effectiveness and Life Cycle Costing* Vol. F, 1983, pp. 535-549.

[11] Rolls-Royce plc (2012), "TotalCare", Available: http://www.rolls-royce.com/sustainability/knowledge/totalcare.jsp

[12] Griego, R. (July 2007), "MBSE Vision and Roadmap", Available: http://www.incose.org/enchantment/docs/07Docs/07Jul_4MBSEroadmap.pdf

[13] Peak, R., Paredis, C., McGinnis, L., Fridenthal, S., and Burkhart, R. "Integrating System Design with Simulation and Analysis Using SysML," *INSIGHT*. Vol. 12, INCOSE, 2009.

[14] Collopy, P. D. "Aerospace System Value Models: A Survey and Observations," American Institute of Aeronautics and Astronautics, No. AIAA 2009-6560, Reston, VA, 2009.

[15] Meisl, C. J. "Th Future of Design Integrated Cost Modelling," *AIAA/AHS/ASEE Aerospace Design Conference*. AIAA, California, U.S.A., 1992.
[16] Wierda, L. S.,"Cost Information Tools for Designers", Ph.D. dissertation, University of Delft, Delft, Netherlands, 1990.
[17] Sheldon, D. F., Huang, G. Q., and Perks, K. "Design for Cost: Past Experience and Recent Developments," *Journal of Engineering Design* Vol. 2, No. 2, 1991, pp. 127-139.
[18] Rush, C., and Roy, R. "Analysis of cost estimating processes used within a concurrent engineering environment throughout a product life cycle," *7th ISPE International conference on concurrent engineering: research and applications*. Lyon, France, 2000.
[19] Ulrich, K. T., Pearson, S. A., and Scott, A. "Does product design really determine 80% of manufacturing cost?," Cambridge, Mass. : Alfred P. Sloan School of Management, Massachusetts Institute of Technology, No. 000658398, 1993.
[20] Dean, E. B. "The Design-To-Cost Manifold," *Symposium on Space Systems Cost Methodologies and Applications*. International Academy of Astronautics, San Diego California U.S.A., 1990.
[21] Dean, E. B., and Unal, R. "Designing for Cost," *Transactions of the American Association of Cost Engineers, 35th Annual Meeting*. Seattle WA, 1991.
[22] Curran, R., Raghunathan, S., and Price, M. "Review of aerospace engineering cost modelling: The genetic causal approach," *Progress in Aerospace Sciences* Vol. 40, 487-534, 2004.
[23] Ostwald, P. "Engineering Cost Estimating." Prentice-Hall, 1992, p. 576.
[24] Stewart, R., Wyskidsa, R., and Johannes, J. "Cost Estimator's Reference Manual." 2nd ed., Wiley Interscience, New York, 1995.
[25] Wright, T. P. "*Factors affecting the cost of airplanes*," *Journal of Aeronautical Sciences* Vol. 3, No. 4, 1936, pp. 122-128.
[26] Schiller, B. "Essentials of Economics." 4th ed., McGraw-Hill/Irwin, New York, 2001.
[27] Roy, R., and Palacio, A. "Cost estimating and risk analysis in manufacturing processes," *Proceedings of MATADOR conference 13-14 July, p177-182*. Manchester, 2000.
[28] Roy, R., Kelvesjo, S., Forsberg, S., and Rush, C. "Quantitative and Qualitative Cost Estimating for Engineering Design," *Journal of Engineering Design* Vol. 12, No. 2, 2001, pp. 147-162.
[29] Scanlan, J., Hill, T., Marsh, R., Bru, C., Dunkley, M., and Cleevely, P. "Cost modelling for aircraft design optimization," *Journal of Engineering Design* Vol. 13, No. 3, 2002, pp. 261-269.
[30] Niazi, A., Dai, J. S., Balabani, S., and Senerviratne, L. "Product Cost Estimation: Technique Classification and Methodology Review,"

Journal of Manufacturing Science and Engineering Vol. 128, No. 2, 2006, pp. 563-575.

[31] Shields, M. D., and Young, S. M. "Managing product life cycle costs: An organizational model," *Cost Management*, No. Autumn, 39-52, 1991.

[32] Greves, D., and Schreiber, B. "Engineering costing techniques in ESA," *ESA Bulletin* Vol. 81, pp. 63-68, 1995.

[33] Briand, L., Emam, K., Surmann, D., Maxwell, K., and Wieczorek, I. "An assessment and comparison of common software cost estimation modeling techniques.," International Software Engineering Research Network, 1998.

[34] Briand, L., and Wieczorek, I. "Resource estimation in software engineering," International Software Engineering Research Network, 2000.

[35] Federal Aviation Administration. "Life Cycle Cost Estimating Handbook 2nd Edition." 2002.

[36] U.S. Department of Defence (DoD). "Parametric estimating handbook, 2nd Ed.." 1999.

[37] Rush, C., and Roy, R. "Capturing quantitative & qualitative knowledge for cost modelling within a CE environment," *ISPE international conference on concurrent engineering: research and applications*. Anaheim, Los Angeles, U.S.A., 2001, pp. 209-18.

[38] Rand Corporation. "Military jet acquisition: technology basics & cost-estimating methodology," Rand Corporation, 2002.

[39] Shuford, R. H. "Activity-Based Costing and Traditional Cost Allocation Structures," *Cost Estimator's Reference Manual*. John Wiley & Sons, New York, 1995, pp. 41-94.

[40] Cooper, R., and Kaplan, R. S. "The Design of Cost Management Systems:Text,Cases and Readings.." 1st ed., Prentice Hall, 1991.

[41] Wierda, L. S. "Linking design, process planning and cost information by feature based modelling," *Journal of Engineering Design* Vol. 2(1), p. 3-19, 1991.

[42] Taylor, I. M. "Cost Engineering - a feature based approach," *85th Meeting of the AGARD Structures and Material Panel* Vol. 14, p.1-9, 1997.

[43] Gerla, G. "Fuzzy logic mathematical tools for approximate reasoning." Dordrecht: Kluwer Academic Publishers, 2001.

[44] Cordon, O., Gomicide, F., and Herrera, F. "Ten years of genetic fuzzy systems: current framework and newtrends," *IFSA World Congress and 20th NAFIPS International Conference, Joint 9th* Vol. 3, p.1241-1246, 2001.

[45] Mamdani, E. H., and Gaines, B. R. "Fuzzy reasoning and its applications." New York Academic Press, 1981.

[46] Daschback, J. M., and Apgar, H. "Design analysis through techniques of parametric cost estimation," *Engineering Costs and Production Economics* Vol. 14, 87-93, 1988.
[47] Scanlan, J., Rao, A., Bru, C., Hale, P., and Marsh, R. "The DATUM Project: A Cost Estimating Environment for the Support of Aerospace Design Decision Making," *Journal of Aircraft* Vol. 43, No. 4, p1022-1028, 2006.
[48] Ahmed, N. U. "A Design and Implementation Model for Life Cycle Cost Management-System," *Information & Management* Vol. 28, No. 4, 1995, pp. 261-269.
[49] "Reliability of systems, equipment and components ", British Standards Institution, 5760-23, 1997.
[50] Barringer, H. P. "Life Cycle Cost Tutorial," *Fifth International Conference on Process Plant Reliability*. Houston, Texas, U.S.A, 1996.
[51] Kawauchi, Y., and Rausand, M. "Life Cycle Cost (LCC) analysis in oil and chemical process industries " ROSS Gemini Centre, 1999.
[52] Wilson, R. L. "Operations and Support Cost Model for New Product Concept Development," *Proceedings of the 8th Annual Conference on Components and Industrial Engineering, 128-131*. 1986.
[53] Sandberg, M., Boart, P., and Larsson, T. "Functional product life-cycle simulation model for cost estimation in conceptual design of jet engine components," *Concurrent Engineering-Research and Applications* Vol. 13, No. 4, 2005, pp. 331-342.
[54] La Rocca, G., Krakers, L., and van Tooren, M. J. L. "Development of an ICAD Generative Model for Aircraft Design, Analysis and Optimisation," *13th IIUG Conference*. Boston, 2002.
[55] Marx, W. J., Mavris, D. N., and Schrage, D. P. "A Hierarchical Aircraft Life Cycle Cost Analysis Model," *AIAA, Aircraft Engineering, Technology, and Operations Congress, 1st*. Los Angeles, CA, U.S.A., 1995.
[56] Johnson, V. S. "Minimizing Life Cycle Cost for Subsonic Commercial Aircraft," *Journal of Aircraft* Vol. 27, No. 2, 1990, pp. 139-145.
[57] Tan, X. C., Xu, Y. C., Early, J., Wang, J., Curran, R., and Raghunathan, S. "A Framework for Systematically Estimating Life Cycle Cost for Integrated Wing," *7th Aviation Technology, Integration and Operations Conference (ATIO)*. AIAA, Belfast, U.K., 2007.
[58] Thokala, P., Scanlan, J., and Chipperfield, A. "Life cycle cost modelling as an aircraft design support tool," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* Vol. 223, 2009.
[59] Newnes, L. B., Mileham, A. R., Cheung, W. M., Marsh, R., Lanham, J. D., Saravi, M. E., and Bradbery, R. W. "Predicting the whole-life cost of a product at the conceptual design stage," *Journal of Engineering Design* Vol. 19, No. 2, 2008, pp. 99-112.

[60] Waghmode, L. Y., and Sahasrabudhe, A. D. "Product life cycle cost modelling: a suggested framework," *2008 1st International Conference on Emerging Trends in Engineering and Technology (ICETET)*. IEEE, Nagpur, Maharashtra, India, 2008, pp. 745-748.
[61] Cheung, W. M., Newnes, L. B., Mileham, A. R., Marsh, R., Lanham, J. D., and Asme. "A study of life cycle costing in the perspectives of research and commercial applications in the 21(st) century," *ASME International Design Engineering Technical Conferences/Computers and Information in Engineering Conference*. Amer Soc Mechanical Engineers, Las Vegas, NV, 2007, pp. 721-730.
[62] Dhillon, B. S. "Life Cycle Costing: Techniques, Models, and Applications." Routledge, 1989.
[63] Liebeck, R., Andrastek, D., Chau, J., Girvin, R., Lyon, R., Rawdon, B., Scott, P. W., and Wright, R. "Advanced subsonic airplane design and economic studies," NASA, No. CR-195443, 1995.
[64] Collopy, P. D., and Curran, R. "The Challenge of Modeling Cost: The Problem," *Innovation and Integration in Aerospace Sciences*. CEIAT, Belfast, Northern Ireland, U.K., 2005.
[65] Rao, A. R., Scanlan, J. P., and Keane, A. J. "Applying multiobjective cost and weight optimization to the initial design of turbine disks," *Journal of Mechanical Design* Vol. 129, No. 12, 2007, pp. 1303-1310.
[66] Dictionary.com (2010), Available: http://dictionary.reference.com
[67] Price, M., Raghunathan, S., and Curran, R. "An integrated systems engineering approach to aircraft design," *Progress in Aerospace Sciences* Vol. 42, No. 4, 2006, pp. 331-376.
[68] International Council on Systems Engineering. "Systems Engineering Handbook Version 3.0." INCOSE, 2000.
[69] Eisner, H. "Essentials of Project and Systems Engineering Management." Wiley, 1996.
[70] International Council on Systems Engineering. "Systems Engineering Handbook Version 2a." 2004, p. 11.
[71] Friedenthal, S., Moore, A., and Steiner, R. (2008), "OMG SysML Tutorial", Available: http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf
[72] Department of Defence (DoD), U. S. A. "Systems Engineering Fundamentals." Defence Acquisition University Press, Fort Belvoir, Virginia, U.S.A, 2001.
[73] Estefan, J. A. "Survey of Model-Based Systems Engineering Methodologies," International Council on Systems Engineering, 2008.
[74] Friedenthal, S. "SysML: Lessons from Early Applications and Future Directions," *INSIGHT*. Vol. 12, INCOSE, 2009.
[75] Levenchuk, A. "SysML is the Point of Departure for MBSE, Not the Destination," *INSIGHT*. Vol. 12, International Council on Systems Engineering, 2009.

[76] Moses, J. "The Anatomy of Large Scale Systems," *MIT Engineering Systems Division Internal Symposium*. Cambridge, Mass., U.S.A., 2002

[77] International Council on Systems Engineering. "Systems Engineering Vision 2020," INCOSE, 2007.

[78] Cloutier, R. "Introduction to this Special Edition on Model-based Systems Engineering," *INSIGHT*. Vol. 12, International Council on Systems Engineering, 2009.

[79] Cooper, D., and La Rocca, G. "Knowledge-based techniques for developing engineering applications in the 21st century," *7th AIAA ATIO Conference, Belfast, Northern Ireland*. 2007.

[80] Siemens (2013), "Open Knowledge Automation", Available: http://www.plm.automation.siemens.com/en_us/products/nx/design/mechanical/open_knowledge_automation.shtml

[81] Genworks (2013), "General-purpose Declarative Language (GDL)", Available: http://www.genworks.com/sessions/346d50934b4/index.html

[82] Verhagen, W. J. C., Bermell-Garcia, P., van Dijk, R. E. C., and Curran, R. "A critical review of Knowledge-Based Engineering: An identification of research challenges," *Advanced Engineering Informatics* Vol. 26, No. 1, 2012, pp. 5-15.

[83] MOKA Consortium (2000), "MOKA: Objectives", Available: http://web1.eng.coventry.ac.uk/moka/mokaobjectives.htm

[84] Curran, R., Verhagen, W. J., Van Tooren, M. J., and van der Laan, T. H. "A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD," *Expert Systems with Applications* Vol. 37, No. 11, 2010, pp. 7336-7350.

[85] McGoey, P. J. (2011), "Knowledge Based Engineering in Aerospace", INCOSE, Presentation, Available: http://www.incose.org/enchantment/docs/11Docs/11Mar_KBEinAerospace.pdf

[86] Delp, C., Cooney, L., Dutenhoffer, C., Gostelow, R., Jackson, M., Wilkerson, M., Kahn, T., and Piggott, S. "The Challenge of Model-based Systems Engineering for Space Systems, Year 2," *INSIGHT*. Vol. 12, INCOSE, 2009.

[87] iSight, Ver. 4.0, Engineous Software Inc., Cary, North Carolina, USA, 2009

[88] Dean, A. "Module and Whole Engine Parametric Analysis Issue 03 " Rolls-Royce plc. , 2006.

[89] Cowles, B. A. "High cycle fatigue in aircraft gas turbines - An industry perspective," *International Journal of Fracture* Vol. 80, No. 2-3, 1996, pp. 147-163.

[90] Huber, N. "Plan to Control High Cycle Fatigue in Engines," *1995 USAF Structural Integrity Program Conference*. San Antonio, Texas, 1995.

[91] IPAS (2010), "IPAS Project Website", Available: http://www.3worlds.org

[92] Kumar, K., Scanlan, J., Eres, M. H., and Reed, P. A. S. "Fatigue Life Prediction fo Life Cycle Costing of Gas Turbine Engines," University of Southampton, 2006.

[93] Adamides, E. D., Stamboulis, Y. A., and Varelis, A. G. "Model-based assessment of military aircraft engine maintenance systems," *Journal of the Operational Research Society* Vol. 55, No. 9, 2004, pp. 957-967.

[94] Warrington, L., Jones, J. A., and Davis, N. "Modelling of maintenance, within discrete event simulation," *Annual Reliability and Maintainability Symposium, 2002 Proceedings*, 2002, pp. 260-265.

[95] Burkett, M. "DMTrade – A Rolls-Royce tool to model the influence of design changes and maintenance strategies on lifetime reliability and maintenance costs," *ASME Turbo Expo 2006: Power for Land, Sea and Air*. 2006.

[96] Scanlan, J. P., Eres, M. H., Yu, T. T., and Wong, J. S. "Life Cycle Modelling Report, IPAS project Work-package 3," University of Southampton, 2008.

[97] Ramakumar, R. "Engineering Reliability." Prentice-Hall, Inc., 1993.

[98] Kapur, K. C., and Lamberson, L. R. "Reliability in Engineering Design." Jon Wiley & Sons, Inc., 1977.

[99] O'Connor, P., and Kleyner, A. "Practical Reliability Engineering." 2012, p. 512.

[100] Jones, J., Warrington, L., and Davis, N. "The use of a discrete event simulation to model the achievement of maintenance free operating time for aerospace systems," *Annual Reliability and Maintainability Symposium, 2001 Proceedings*, 2001, pp. 170-175.

[101] Boydstun, F., Graul, M., Benjamin, P., and Painter, M. "New perspectives towards modeling depot MRO," *Proceedings of the 2002 Winter Simulation Conference, Vols 1 and 2*, 2002, pp. 738-746.

[102] Gatland, R., Yang, E., and Buxton, K. "Solving engine maintenance capacity problems with simulation," *Proceedings of the 1997 Winter Simulation Conference*. 1997, pp. 892-899.

[103] Gupta, P., Bazargan, M., and McGrath, R. N. "Simulation model for aircraft line maintenance planning," *Annual Reliability and Maintainability Symposium. 2003 Proceedings (Cat. No.03CH37415)|Annual Reliability and Maintainability Symposium. 2003 Proceedings (Cat. No.03CH37415)*, 2003, pp. 387-91|xviii+622.

[104] Jones, J. A., Warrington, L., and Davis, N. "Integrated modelling of system functional, maintenance & environmental factors," *Annual Reliability and Maintainability Symposium, 2002 Proceedings*, 2002, pp. 399-403.

[105] Painter, M. K., Erraguntla, M., Hogg, G. L., Beachkofski, B., and Ieee. "Using simulation, data mining, and knowledge discovery techniques

for optimized aircraft engine fleet management," *2006 Winter Simulation Conference*. Ieee, Monterey, CA, 2006, pp. 1253-1260.
[106] Augenbroe, G. "Trends in building simulation," *Building and Environment* Vol. 37, No. 8-9, 2002, pp. 891-902.
[107] Augenbroe, G., and Hensen, J. "Simulation for better building design," *Building and Environment* Vol. 39, No. 8, 2004, pp. 875-877.
[108] Hensen, J. L. M., and Nakahara, N. "Building and environmental performance simulation: Current state and future issues," *Building and Environment* Vol. 36, No. 6, 2001, pp. 671-672.
[109] Imagine That Inc. (2013), "ExtendSim Product Website", Available: http://www.extendsim.com/index.html
[110] Yu, T.-T.,"The Development of a Hybrid Simulation Modelling Approach Based on Agents and Discrete-Event Modelling", Ph.D. dissertation, University of Southampton, 2008.
[111] Tewoldeberhan, T. W., Verbaraeck, A., Valentin, E., and Bardonnet, G. "An Evaluation and Selection Methodology for Discrete-Event Simulation Software," *Proceedings of the Winter Simulation Conference* Vol. 1, pp 67-75, 2002.
[112] Seila, A. "Spreadsheet simulation," *Winter Simulation Conference*. 2005, pp. 33-40.
[113] Panko, R. R. "Spreadsheet Errors: What We Know. What We Think We Can Do," *Proceedings of the Spreadsheet Risk Symposium*. European Spreadsheet Risks Interest Group (EuSpRIG), Greenwich, England, 2000.
[114] Paine, J. "Spreadsheet Structure Discovery with Logic Programming," *Arxiv preprint arXiv:0802.3940*, 2008.
[115] Liu, Z., Mavris, D. N., and Volovoi, V. V. "Creep life prediction of gas turbine components under varying operating conditions." Vol. 2, American Society of Mechanical Engineers, New Orleans, LA, United states, 2001, pp. 483-489.
[116] Venter, G., Haftka, R. T., and Chirehdast, M. "Response surface approximations for fatigue life prediction." Vol. 2, AIAA, Kissimmee, FL, USA, 1997, pp. 1383-1396.
[117] Wallace, J. M., and Mavris, D. N. "Creep life uncertainty assessment of a gas turbine airfoil." Vol. 1, American Inst. Aeronautics and Astronautics Inc., Norfolk, VA, United states, 2003, pp. 746-754.
[118] Smith, M. E. B.,"A Parametric Physics Based Creep Life Prediction Approach to Gas Turbine Blade Conceptual Design", Ph.D. dissertation, School of Aerospace Engineering, Georgia Institute of Technology, 2008.
[119] Doyle, A. "R-R details three-shaft RB285 study for 150-seat market," *Flight International*. 2008.
[120] Flight International. "Green sky thinking - carbon credits and the propfan comeback?." 2007.

[121] Cumpsty, N. "Jet Propulsion." Second ed., Cambridge University Press, 2003.
[122] Wickerson, J. "Holistic Gas Turbine Training Course Notes." Rolls Royce plc, 2007.
[123] Walsh, P. P., and Fletcher, P. "Gas Turbine Performance." Blackwell Science, 2004.
[124] Wilcock, R. C., Young, J. B., and Horlock, J. H. "The Effect of Turbine Blade Cooling on the Cycle Efficiency of Gas Turbine Power Cycles," *Journal of Engineering for Gas Turbines and Power* Vol. 127, No. 1, 2005, pp. 109-120.
[125] Rolls-Royce plc (2012), "Design systems and tools", Available: http://www.rolls-royce.com/about/technology/systems_tech/design_systems_tools.jsp
[126] Value Driven Design Institute (2010), "AIAA Value Driven Design Program Committee Homepage", Available: http://www.vddi.org/vdd-home.htm#WhatVDD
[127] Value Driven Design Program Committee (2012), "About Value-Driven Design", Available: https://info.aiaa.org/tac/pc/VDDPC/Web%20Pages/About%20VDDPC.aspx
[128] Collopy, P. D., and Hollingsworth, P. "Value-Driven Design," *9th AIAA Aviation Technology, Integration and Operations (ATIO) Conference*. South Carolina, U.S.A., 2009.
[129] Keeney, R. L. "Creativity in Decision Making with Value-Focused Thinking," *Sloan Management Review* Vol. 35, No. 4, 1994, pp. 33-41.
[130] Hazelrigg, G. A. "A Framework for Decision-Based Engineering Design," *ASME Journal of Mechanical Design* Vol. 120, 1998, pp. 653-658.
[131] Brathwaite, J., and Saleh, J. H. "On the Concept of Value and its Importance to Space Systems Design and Acquisition," *AIAA Space Conference and Exposition*. San Diego, CA, U.S.A., 2008.
[132] Cheung, J., Scanlan, J., Wong, J., Forrester, J., Eres, H., Collopy, P., Hollingsworth, P., and Wiseall, S. "Application of Value-Driven Design to Commercial Aero-Engine Systems," *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. Fort Worth, Texas, U.S.A., 2010.
[133] Brown, O. C., Eremenko, P., and Collopy, P. D. "Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary from Phase 1 of the DARPA System F6 Program," *AIAA SPACE 2009*. Pasadena, California, U.S.A., 2009.
[134] O'Neil, M. G., Yue, H., Nag, S., Grogan, P., and Weck, O. L. d. "Comparing and Optimizing the DARPA System F6 Program Value-

Centric Design Methodologies," *AIAA SPACE*. Anaheim, California, U.S.A., 2010.
[135] Castagne, S., Curran, R., and Collopy, P. "Implementation of Value-driven optimisation for the design of aircraft fuselage panels," *International Journal of Production Economics* Vol. 117, 2009, pp. 381-388.
[136] Peoples, R., and Willcox, K. "Value-Based Multidisciplinary Optimization for Commercial Aircraft Design and Business Risk Assessment," *Journal of Aircraft* Vol. 43, No. 4, 2006, pp. 913-921.
[137] Collopy, P. "Surplus Value in Propulsion System Design Optimization," AIAA, No. 97-3159, 1997.
[138] P. Collopy, J. Forrester, S. Wiseall, P. Hollingsworth, J. Scanlan, H. Eres, J. Cheung, J. Wong, and Hoptroff, J. "Advanced Cost Modelling Methodologies: Value Driven Design Scoping and Feasibility Study," Rolls-Royce plc, No. DNS153982, 2009.
[139] Flight Optimization System (F.L.O.P.S), Ver. Release 6.11, McCullers, L. A., 2004
[140] Jenkinson, L. R., Simpkin, P., and Rhodes, D. "Civil Jet Aircraft Design." Butterworth-Heinemann, 1999, p. 436.
[141] Soban, D. S., Hollingsworth, P., and Price, M. A. "Defining a Research Agenda in Value Driven Design: Questions That Need to be Asked," *Air Transport and Operations: Proceedings of the Second International Air Transport and Operations Symposium 2011*. Delft, the Netherlands, 2011.
[142] InterCAX LLC (2012), "Intercax Homepage", Available: http://www.intercax.com/