

University of Southampton Research Repository ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND THE ENVIRONMENT

Aeronautics, Astronautics and Computational Engineering

A framework for design rationale capture and use during geometry design

by

Jeroen Robbert van Schaik

Thesis for the degree of Doctor of Philosophy

March 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND THE ENVIRONMENT

Aeronautics, Astronautics and Computational Engineering

Thesis for the degree of Doctor of Philosophy

A FRAMEWORK FOR DESIGN RATIONALE CAPTURE AND USE DURING GEOMETRY DESIGN

Jeroen Robbert van Schaik

Despite broad agreement on the utility of design rationale use and capture, a review of the relevant literature shows that industrial usage remains limited, especially during geometry design.

An initial field study confirmed low design rationale capture during the geometry design stage. The lack of linking between design rationale and geometry models is identified as a factor holding back design rationale capture.

A toolset is presented to link entities in geometry models to design rationale, allowing the creation of design rationale referring to a specific geometry design decision. Using the design rationale links it is possible to create graphs of the structure of geometry models and attached rationale. Furthermore the presence and quantity of design rationale can be displayed as a coloured overlay on the geometry.

The toolset has been tested by 7 groups of student-designers, and although the uptake of the design rationale linking tool by the users was low, results show that groups using the tool captured relatively more design rationale during geometry design, although reservations have to be made regarding to self-selection bias. The study shows that the availability of design rationale linking tools is not by itself enough to improve design rationale capture during geometry design.

Contents

ABSTRACT	i
Contents.....	i
List of tables.....	vii
List of figures	ix
DECLARATION OF AUTHORSHIP	xiii
Acknowledgements	xv
Definitions and Abbreviations	xvii
1. Introduction.....	1
1.1 Motivation	2
1.1.1 Particular importance for the aeronautical engineering industry.....	3
1.2 Research Background	4
1.3 Objectives	5
1.4 Limitations	5
1.5 Research Methodology.....	6
1.5.1 Establishing success criteria	7
1.5.2 Descriptive 1 phase	9
1.5.3 Prescriptive 1 phase	9
1.5.4 Descriptive 2 phase	10
1.6 Outline of the thesis	10
2. Relevant works: geometry modelling.....	13
2.1 Usage of geometry modelling during the design process	13
2.1.1 The role of geometry modelling in the design process.....	14
2.2 Geometry modelling: Definition of terms and background	17
2.2.1 Computer Aided Design.....	17
2.2.2 Constructive solid geometry modelling.....	17
2.2.3 Boundary representation geometry	17
2.2.4 Feature-based geometry modelling	18
2.2.4.1 Structure of feature-based geometry models.....	21
2.2.4.2 Reordering operations to obtain different results.....	21
2.2.1 Parametric and associative geometry modelling.....	22
2.2.2 Design intent.....	24
2.3 Usage issues encountered during the use of feature-based geometry models as a means of communicating designs.....	26

2.3.1	Design rationale can be difficult to infer	28
2.3.2	The product information recorded does not describe the full design 28	
2.3.3	Design intent can be difficult to infer from the model	28
2.3.4	Geometry models do not capture design alternatives well	29
2.4	Annotation and mark-up of geometry models as a design documentation tool	29
2.4.1	Motivation for annotation	31
2.4.2	Geometry annotation anchor.....	33
2.4.2.1	Annotation anchoring method	33
2.4.2.2	Choice of geometry representation to annotate.....	34
2.4.2.3	Annotation anchor granularity	35
2.4.3	Annotation content.....	36
2.5	Summary	37
3.	Relevant work: Design rationale.....	39
3.1	Design Rationale: definitions	40
3.1.1	Design space	40
3.1.2	Design rationale: content.....	41
3.1.3	Design rationale: method of storage	42
3.2	Why capture design rationale	44
3.2.1	The costs versus the benefits of design rationale capture	46
3.3	Design Rationale argumentation structures.....	47
3.3.1	Toulmin arguments	47
3.3.2	Issue Based Information Systems	47
3.3.3	Questions, Options, Criteria (QOC).....	48
3.3.4	Decision Representation Language (DRL)	49
3.4	Design rationale capture and editing tools.....	50
3.4.1	Graphical design rationale editors.....	50
3.4.1.1	Compendium	51
3.4.1.2	DRed (Design Rationale editor).....	52
3.4.2	Design rationale capture in software design.....	55
3.4.3	Automated rationale capture.....	55
3.4.3.1	Record and replay.....	56
3.4.3.2	Automatic Generation	56
3.5	Integrating design rationale capture functionality into existing design tools.....	57
3.5.1	Examples of integration of design rationale capture with geometry models	58
3.5.2	The relationship between geometry model design intent and design rationale	59

3.5.3	The relationship between geometry annotations and linked design rationale	60
3.6	Data on design rationale capture	61
3.6.1	Quantitative data on design rationale capture	61
3.6.2	Design rationale is captured especially poorly during detail design.	62
3.6.3	Why is Design rationale capture avoided?	62
3.7	Summary	63
4.	Design rationale capture during an aeronautical engineering design process: a case study	65
4.1	Study goals	65
4.2	Choice of a design rationale capture metric	65
4.3	Study setup	66
4.3.1	Data collection	66
4.3.2	Design project	67
4.3.3	Designers	68
4.3.4	Design tools	69
4.4	Design Rationale Capture statistics	69
4.5	Discussion	74
4.6	Conclusions	75
4.7	Summary	76
5.	The approach: Linking design rationale to entities in geometry models	77
5.1	The need for linking design rationale to geometry model entities	77
5.2	Requirements for a tools to improve design rationale capture during geometry design	78
5.2.1	Integration of design rationale linking functionality into the geometry modelling tool	78
5.2.2	The need for a low granularity link between rationale and geometry	78
5.2.3	Inclusion of concept design rationale	79
5.2.4	Support the documentation of geometry model design intent	79
5.2.5	Systematic approach and audit	80
6.	Linking design rationale to geometry entities	81
6.1	Linking design rationales to feature-based geometry models	81
6.2	Database or file-based design rationales	82
6.3	Practical implementation of the RatLink tool	83
6.3.1	Choice of CAD tool	84

6.3.2	Link Implementation	84
6.3.2.1	Granularity of design rationale links.....	86
6.3.2.2	Limitations of this linking technology	87
6.3.3	User interface design	87
6.3.3.1	Design rationale toolbar.....	87
6.3.3.2	Design rationale linking menu	88
6.3.3.3	Tree view of database for rationale navigation	89
6.4	RatLink tool demonstration.....	91
6.4.1	Setup.....	91
6.4.2	Linking design rationale.....	91
6.4.3	Viewing the rationale link from the rationale editor.....	92
6.5	Summary	93
7.	Creating graphs of geometry model structure and associated design rationale.....	95
7.1	Related works: creating graphs of geometry model structures ..	95
7.1.1	The use of graphs as tools for design.	98
7.2	Requirements for a graphing tool	98
7.3	Practical implementation of a geometry and rationale graphing tool	99
7.3.1	Iterating through the operation tree.....	102
7.3.2	Establishing additional connections	102
7.3.3	Retrieving design rationale	103
7.3.4	Filtering graphs	103
7.3.5	Graph nesting.....	103
7.3.6	Serialization, layout and viewing of graphs	104
7.3.7	Colour scheme.....	107
7.3.8	User interface	108
7.4	Features of graphs produced by the GRapher tool.....	111
7.4.1	Feature relationships	111
7.4.2	Assembly relationships	113
7.4.3	Creating geometry graphs for large parts and assemblies	116
7.5	Applications of the graphs produced	118
7.5.1	Typical usage scenario.....	119
7.5.2	System usage recommendations	121
7.6	Summary	121
8.	Audit of geometry indicating the presence of linked design rationale	123
8.1	Related works	123

8.2	Requirements for a geometry and design rationale audit tool .	124
8.3	Implementation	124
8.3.1	Geometry model and database interactions	125
8.3.2	User interface	126
8.3.1	Colour scheme	127
8.4	Examples of the visual feedback of produced by the GReAT tool	128
8.5	Applications of the geometry audit functionality of GReAT.....	131
8.5.1	During design reviews	131
8.5.2	As a tool to assist editing of parametric geometry models	132
8.6	Summary	133
9.	Evaluation: field testing of the proposed design rationale linking toolset.....	135
9.1	Descriptive 2 study: a thought experiment	135
	Study setup	136
9.2	136	
9.2.1	Data collection	137
9.2.2	Design assignment	138
9.2.3	Tool availability	138
9.2.4	User group	138
9.2.5	Time schedule and work hours available.....	139
9.3	Study results	139
9.3.1	Design rationale capture.....	139
9.3.1	Use of design rationale linking tool	143
9.3.2	Did the use of the design rationale toolset correlate with higher design rationale capture during the geometry design phase?	144
9.3.3	Use of design rationale feedback functionality	144
9.3.4	Feedback from users	144
9.4	Discussion	145
9.4.1	Self-selection bias	145
9.4.2	Possible reasons for low tool use	146
9.5	Conclusion	147
10.	Conclusions	149
10.1	Review of objectives	149
10.2	Contributions	151
10.3	Recommendations for future work.....	152
10.4	Summary	153
	Appendices	155

Appendix 1: Algorithms for geometry model audit	157
Appendix 2: Filtering geometry model graphs	163
Appendix 3: graphml file for the graph of the structure of a simple geometry model, an engine bracket.	165
Appendix 4: Statistical significance calculation for test result.....	191
Appendix 5: individual peer review score versus design rationale captured	195
List of References	199

List of tables

Table 1 An overview of the motivations for the annotation of geometry.	31
Table 2 Type of annotation anchor used.	34
Table 3 geometry annotation classified by choice of geometry representation.	34
Table 4 Level of formality of geometry annotations in literature	36
Table 5 Definitions of design rationale found in literature	41
Table 6 possible reasons for low design rationale capture during geometry design.	74
Table 7 Content of metadata stored in graph	105
Table 8 GRapher node colours giving status feedback.....	108
Table 9 options of the design audit functionality.....	126
Table 10 Design status feedback colours.	128
Table 11 Use of design rationale linking tool	143
Table 12 Design rationale nodes captured during the second half of the design exercise.....	191
Table 13 Standard deviation and mean for groups using or not using the design rationale capture tool	192
Table 14 Critical t-values for a degree of freedom of 26	193

List of figures

Figure 1 Structure of research as an implementation of the Design Research Methodology (DRM) (Blessing and Chakrabarti 2009).....	7
Figure 2 Mechanical engineering design process with approval based iteration, based on Pugh (1990).....	14
Figure 3 The use of CAD for geometry modelling during the design process..	15
Figure 4 the reduced role of CAD geometry modelling in a KBE product development environment.....	16
Figure 5 A simple geometry model consisting of an output geometry, shown on the right and the set of operations defining the model in the form of a modelling tree on the left.	20
Figure 6 A modelling operation as a part of a block model.....	21
Figure 7 The four different possible geometries obtained by reordering the order of the three operations in the geometry model.	22
Figure 8 A geometry model viewed as an intermediary artefact between inputs and geometry representation.....	23
Figure 9 Design intent in a simple geometry model: geometry meaning is retained despite parameter changes	25
Figure 10 Changes to an input parameter of a geometry model which lead to a loss of meaning.....	26
Figure 11 an example design rationale map	43
Figure 12 The structure of a Toulmin argument	47
Figure 13 A design rationale using the IBIS argumentation structure created in Compendium.....	48
Figure 14 A QOC design rationale diagram, based on (MacLean, Bellotti et al. 1993).	49

Figure 15 Compendium node types and argumentation structure, an example from the Compendium user instructions.....	52
Figure 16 DRed node types, which include node status, from (Auricchio and Bracewell 2013b)	53
Figure 17 a typical DRed map, from http://www-edc.eng.cam.ac.uk/projects/capturetools/	54
Figure 18 The DECODE 1 UAV in flight	68
Figure 19 The distribution of design rationale node occurrences captured during the project by node type.....	70
Figure 20 The top level view of design rationale maps of the aircraft design.	71
Figure 21 Design rationale nodes created during the project by all design teams.	71
Figure 22 a quantitative view of design rationale captured by the design team during the design process.	72
Figure 23 a design rationale map produced during the design process detailing interface design.....	73
Figure 24 High level overview of the interactions between the design rationale linking tools and the design rationale database.	83
Figure 25 Linking rationale to a geometry model by attaching attributes to the operations or faces in the operations tree. The metadata which is stored on both ends of the link provides additional information on the entities linked.....	85
Figure 26 The design rationale toolbar providing access to design rationale linking functionality.....	87
Figure 27 Design rationale linking menu	89
Figure 28 A detail of the design rationale linking interface showing a tree-view of a set of rationale maps and nodes.	90
Figure 29 Design rationale linking user interface, allowing selection either graphically or from the operation-tree.	92

Figure 30 A linked rationale map, 'Fillet sizing rationale' has been created in Compendium and rationale for the decisions can be captured using the normal Compendium interface.	93
Figure 31 Parent-Child operation dependency display in Solidworks	96
Figure 32 Geometry model structure feedback displaying external references in Solidworks.....	97
Figure 33 Interactions of the GRapher tool with the geometry model and the design rationale database	100
Figure 34 Method used by the geometry graphing tool to create graphs of model structure and design rationale	101
Figure 35 Graphml header including definition of geometry metadata.....	105
Figure 36 an example node in the Graphml format	107
Figure 37 Grapher user interface, file selection menu.....	109
Figure 38 Options presented to the user to allow the customization of the graph created depending on points on interest in the model.	110
Figure 39 Rationale database selection menu.....	111
Figure 40 the basic geometry model structure for the part shown in Figure 29, including the design rationale map linked to the model as shown in Figure 30.....	112
Figure 41 the constituent parts of a simple assembly prior to adding mate constraints. Based on the example in (Shah and Rogers 1993).	113
Figure 42 an assembly of the parts in Figure 41 including mate relationships, an equation and an external reference.	114
Figure 43 The graph resulting from the assembly in Figure 42 including the mate relations between features, an equation and an external reference.....	115
Figure 44 Geometry model graph of a large assembly.	117

Figure 45 An example assembly to which parametric modifications have to be made.....	119
Figure 46 The graph created by the GRapher tool for the assembly in Figure 42 showing the chain of parameter blocks which pass from the design table dimension on the left, to the operation producing the geometry on the right.....	120
Figure 47 Structure of the geometry audit tool	125
Figure 48 The design audit menu shown integrated in Solidworks. The option selected here result in the geometry seen in Figure 49.	126
Figure 49 The example geometry from Figure 29 after design rationale and geometry audit.	129
Figure 50 The audit tool applied to a complex piece of geometry, an UAV fuselage.....	129
Figure 51 Fuselage design showing result of design audit after rationale has been added for the skin thickness.	130
Figure 52 A graph showing the cumulative design rationale node capture per team over time. The groups marked with an asterisk used the design rationale to geometry linking tool.....	141
Figure 53 Distribution of design rationale capture per team. Asterisks indicate use of design rationale linking tool, numbers in brackets the number of team members.	142
Figure 54 overall design rationale capture versus normalized peer review score for all users	196

DECLARATION OF AUTHORSHIP

I, Jeroen van Schaik

declare that the thesis entitled

A framework for design rationale capture and use during geometry 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: (van Schaik et al. 2011)

Signed:

Date:.....13th of September 2014.....

Acknowledgements

I would like to thank:

My supervisors Jim Scanlan and Andy Keane for their help and encouragement during throughout the PhD.

My colleagues in the DECODE project in general for their contributions to the beautiful aircraft we have built, and Dirk Gorissen in particular for his assistance to a beginning programmer.

My parents, Roel and Jacomine, and my grandmother, Omi, for their frequent encouragement and financial support, particularly during the last year of the PhD.

And last but not least Pauline Fossat for her long patience.

Definitions and Abbreviations

API	Application Programming Interface
B-rep	Boundary Representation
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CFD	Computational Fluid Dynamics
CSG	Constructive Solid Geometry
CSGM	Constructive Solid Geometry Model
DECODE	Decision Environment for COmplex DESign
DRed	Design Rationale Editor
DRL	Decision Representation Language
DXF	Drawing Exchange Format
FBGM	Feature-Based Geometry Modelling
FDAG	Feature Dependency Acyclic Graph
FEA	Finite Element Analysis
gIBIS	graphical Issue Based Information Systems
GRapher	Geometry and Rationale grapher
GReAT	Geometry & Rationale Audit Tool
IBIS	Issue Based Information Systems
IGES	Initial Graphics Exchange Specification
ISO	International Organisation for Standardization
itIBIS	indented text IBIS

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

KBE	Knowledge based Engineering
KIM	Knowledge and Information Management
KMI	Knowledge Media Institute at the Open University
NBS	National Bureau of Standards
PDM	Product Development Management
PLM	Product Lifecycle Management
QOC	Questions, Options, Criteria
QOC	Questions, Options, Criteria
RatLink	Rationale Link tool
STEP	STandard for the Exchange of Product model data
VoIP	Voice over IP
VRML	Virtual Reality Modelling Language

1. Introduction

The central question explored in this thesis is how design rationale can be linked to geometry models, how these links can be used in the design process and whether this aids the design process and improves design rationale capture and use.

Over forty years have passed since the seminal paper by Kunz and Rittel (1970) introducing Issue Based Information Systems, which is at the origin of today's design rationale systems. Since then numerous papers on design rationale capture and use in an engineering context have been published, and while there is broad agreement that design rationale capture is useful, it is still not used enough in practice (Burge 2005).

There has been substantial application of design rationale research to aeronautical engineering design problems in the form of the Design Rationale editor (DRed) tool (Bracewell and Wallace 2003, Bracewell, Gourtovaia et al. 2009). However design rationale is captured this is usually done early in the design process, or late during the evaluation stage (Bracewell, Gourtovaia et al. 2007), with a marked drop during geometry design (Eng, Bracewell et al. 2009).

While design rationale has been used to support a wide range of engineering activities, such as software design (Burge and Brown 2008) and civil engineering (Eng, Marfisi et al. 2011), the design activity considered in this thesis concerns mechanical engineering design processes which include a phase of geometry design.

Geometry models have replaced drafted engineering drawings and are now the main method of communicating designs (Patel, Ball et al. 2009) and are increasingly important both as a means of storing design information and a means of performing other geometry related design activities such as FEA, CFD and weight estimation. Despite the central function of geometry modelling in the design process it is widely recognized that the current CAD models, PLM systems and design reports do not capture the whole design process (Auricchio and Bracewell 2013b) and particularly the insufficiency of geometry models to support the capture of design decision information (McMahon, Lowe et al. 2004) and the resulting difficulty in understanding the design intent of the

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

geometry models during design reuse. (Robertson and Radcliffe 2009, Salehi and McMahan 2011)

The absence of sufficient context explaining the design intent of geometry models has been identified as a reason for design reuse failure (Busby 1999), while providing additional information helps speed up editing of geometry models during design reuse (Alducin-Quintero, Contero et al. 2011).

Numerous authors have called for the integration of design rationale into design tools and processes (Szykman, Sriram et al. 2001, Burge 2005, Bracewell, Courtovaia et al. 2007, Hoey and Foyle 2007), as well as the association of design rationale and geometry models (Klein 1993, Klein 1997, Lee 1997, Regli, Hu et al. 2000, Bracewell, Ahmed et al. 2004, Chan 2007, McKay, Kundu et al. 2009).

The work presented in this thesis investigates the opportunities to improve the documentation of geometry models by means of annotations with design rationale.

1.1 Motivation

The work presented in this thesis is motivated above all by the importance of knowledge management in engineering design. The issues related to knowledge management are most apparent during design communication and reuse. Demian and Fruchter (2006) emphasise the importance of understanding the context of a design during design reuse. Furthermore, understanding the context of an artefact or data source has been identified as a key difference between novice and expert designers (Ahmed, Blessing et al. 1999). Experienced designers also frequently refer to past designs, including trade-offs, alternatives considered and reasons for the original decisions (Ahmed and Wallace 2004), and during discussions of past designs references to design issues make up 56% of total references, which further increases with design experience (Ahmed and Wallace 2006).

Although the amount of information which is obtained directly from colleagues rather than from documentary sources is decreasing it still amounts to 70% of total information requests (Aurisicchio, Bracewell et al. 2010). However, the current state of the art poorly captures design information:

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

“Technical reports, the main digital corporate source for design information together with computer-aided design (CAD) and other models in product lifecycle management (PLM) systems, capture only fragments of design processes.” (Auricchio and Bracewell 2013). More particularly: “It is a lack of formal representations for product development information that creates a significant barrier to its effective capture and exchange.” (Szykman, Sriram et al. 2001). These factors illustrate both the insufficiencies of the current product data management (PDM) strategies and the reliance on information obtained directly from colleagues

1.1.1 Particular importance for the aeronautical engineering industry

The methods and tools presented in this thesis have been developed with a particular focus on the aeronautical engineering industry. Although the work presented in this thesis is oriented to the aeronautical engineering industry, which features highly complex products and long product development, production and use cycles, the tools proposed can be applied to any mechanical engineering design project using commercially available industry standard CAD toolsets such as Solidworks.

The following factors, associated with the aeronautical engineering industry, can be identified which make the detailed documentation of design decisions valuable:

- Long service and manufacturing life of products, for example the Chinook helicopter is now in its 50th year of production (Magnuson 2012).
- Frequent occurrence of upgraded variants, such as new engine upgrades, where large parts of the design are re-used, necessitating a thorough understanding of the design.
- High accountability is required to certify designs; design iteration in service is very expensive as it can require re-certification
- The introduction of service based business models such as ‘power by the hour’ as a key change which requires product knowledge to remain available for the lifetime of the product (McMahon, Giess et al. 2005).

Moreover changes in company structure and employment patterns make design knowledge retention in the collective memory more difficult:

- Higher personnel turnover amongst engineers leading to gradual loss of knowledge of a design (Ahmed 2005).

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Reduced collocation of design teams reduces awareness of design decisions outside the immediate environment.
- Multilingual and multinational design teams make explicit communication more important.
- Due to the longer product design cycles a designer may only experience a couple of cycles during the course of a career.
- High specialization and diffusely distributed design knowledge during detail design (Clarkson and Hamilton 2000).

Detail design knowledge and decision rationale is often known only to a small subset of designers. Clarkson and Hamilton (2000) describe a situation where there is little awareness of the practices, considerations and decisions of designers outside the specialist area. The decision knowledge for detailed design decisions will also be captured less in design reports and secondary sources such as emails and presentations, as there will have been little need to communicate the decision outside the direct stakeholders.

1.2 Research Background

The research presented in this thesis was conducted as a part of the Decision Environment for Complex Designs (DECODE) project, an EPSRC funded project at the University of Southampton. The DECODE project was aimed at creating a design system for the design, optimization and construction of search and rescue (SAR) unmanned aerial vehicles (UAV). DECODE is a multidisciplinary research project at the University of Southampton that has resulted in publications in the area of value driven design (Ferraro, Gorissen et al. 2012, Surendra, Ferraro et al. 2012), design optimization (Gorissen, Quaranta et al. 2012), operational simulation (Schumann, Scanlan et al. 2011), computational fluid dynamics and design rationale capture (van Schaik, Scanlan et al. 2011) in design situations.

A key aspect of this project is the completion of a series of full design cycles, allowing the design tools to be used for practical design problems and the whole design process to be considered. The importance of capturing design rationale, both for the design system as well as for the individual designs, is a key aspect of the DECODE system.

1.3 Objectives

The principal question of this thesis is:

How can a practical link be created between geometry model entities and design rationale, and would the availability of such a linking facility improve design rationale capture during geometry design.

From this principal question the following aims and objectives can be identified for this thesis:

Objective 1: To review the existing literature on design rationale, geometry modelling, computer aided design, geometry annotation and knowledge management to create an understanding of the issues concerning design rationale capture during geometry design.

Objective 2: To quantitatively study the capture of design rationale using the pre-existing design rationale and geometry modelling toolset to establish a baseline pattern of design rationale capture using a state of the art toolset during a mechanical engineering design process.

Objective 3: Based on the understanding of design rationale capture during the mechanical engineering design process as a result of objectives 1 and 2, to propose new approaches to aid capture and further integrate design rationale into the mechanical engineering design process.

Objective 4: To implement and integrate a new tool into the existing industrial toolset, according to the methods and requirements proposed for objective 3.

Objective 5: To evaluate the effectiveness of the new methods and tools during field testing.

1.4 Limitations

The work presented in this thesis investigates how entities in geometry models can be linked to design rationale, and how the linked design rationale can be used during the geometry design process.

There have been a large number of approaches in related fields such as:

- The function of geometry modelling in the design process

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- The capture of design knowledge in formal systems such as Knowledge Based Engineering (KBE) tools and Ontologies.
- The methods and representation used to capture design rationale

However, for practical purposes it is necessary to limit the scope of this research purely to the capture and use of design rationale during the geometry design phase without further changes to the current design process.

While in some cases matters outside the scope of this research will be discussed briefly for context and completeness, the following areas are outside the scope of the research presented here, and will not be considered in any depth:

- The archival of geometry models and future availability of CAD tools to edit them.
- The type of design rationale representation to be used
- The design and creation of design rationale editing software
- The use of non feature-based geometry modelling methods, as defined in section 2.2.3, to create or edit geometry.
- The different methods for use of CAD tools for geometry modelling, such as top-down or bottom-up modelling techniques.
- The transfer of design rationale links between CAD systems

1.5 Research Methodology

The research methodology used for this thesis is based on the Design Research Methodology (DRM) as proposed by Blessing and Chakrabati (2009). The method for the application of the DRM to computational design tool creation by Bracewell, Shea et al. (2001) provides a further framework for design research in the domain of the creation of software tools to support design activities. The DRM consists of four main stages (Blessing and Chakrabarti 2009):

- The definition of criteria for the success of the study.
- A Descriptive 1 study seeking to establish the current design process either experimentally or from literature.
- A Prescriptive 1 study to define potential solutions to issues found.
- A Descriptive 2 study to measure the effect of the solution on the design process.

The alternation of descriptive and prescriptive studies has been retained for this thesis. The implementation of the DRM for the research presented in this thesis is summarized in Figure 1.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

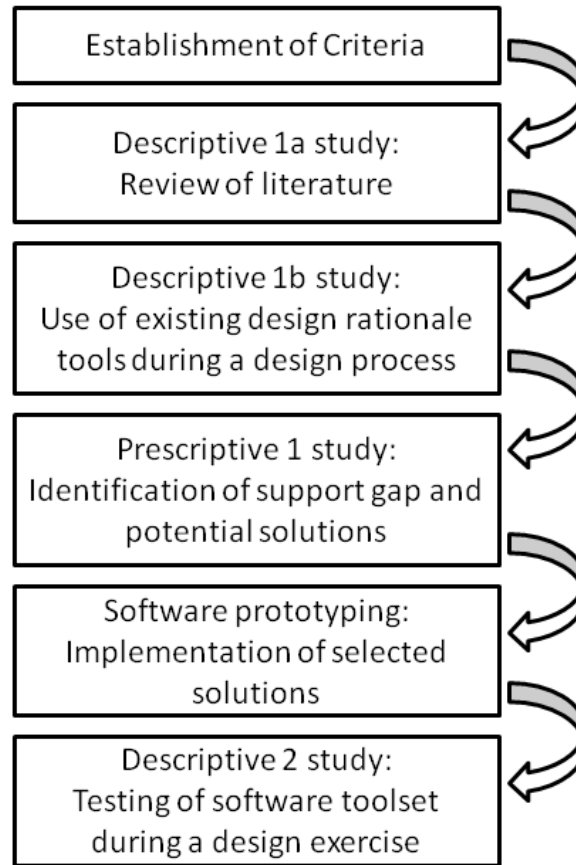


Figure 1 Structure of research as an implementation of the Design Research Methodology (DRM) (Blessing and Chakrabarti 2009).

The next few sections will discuss the implementation of the of the DRM framework in the context of this research.

1.5.1 Establishing success criteria

The criteria stage of the DRM process as implemented in the Computer-Aided Engineering Design Research Methodology (Bracewell, Shea et al. 2001) requires the establishment of measurable criteria for the success of the proposed toolset. Bracewell, Shea et al. (2001) suggest for success criteria to be broken down into an a measurable quantity which can be plausibly linked to the overall objective such as “reduce time to market” with an overall objective of increasing profit or “reduce structural mass” with an overall objective of reducing cost.

Although the preference for quantitative data should be treated with caution, as a holistic view of the design process can be lost (Eckert, Stacey et al. 2004), the absence of quantitative metrics makes it difficult to evaluate tool usage and

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

utility. Design rationale studies frequently give no quantitative data on the use of the design rationale tool; a review of existing quantitative design rationale studies is presented in section 3.6.1.

Where possible quantitative data, rather than qualitative data, has been used to evaluate the usage of design tools. As discussed in section 1.1 this thesis is motivated by the lack of in-context design information gathered during the design process, and the consequent lack of understanding of the design decisions. The success criterion was therefore defined as follows:

“Improve the gathering of in-context design decision information during computer aided design and geometry modelling”

Several possible metrics were considered as a measurable quantity for this success criterion; see section 4.2 for an overview. The two studies presented in this thesis use the number of design rationale nodes captured as the principal metric of design rationale capture. Precedents can be found for this metric: Bracewell and Wallace (2003) define the quantity of design rationale nodes captured per designer-hour as a success criteria for the success of a proposed design capture tool, and early studies of design rationale capture during design processes also use this metric (Conklin and Begeman 1988, Burgess Yakemovic and Conklin 1990, Conklin and Burgess Yakemovic 1991).

The quantity of design rationale nodes captured can plausibly be linked to the quantity of design information captured during the design process, as the interconnected nature of design rationale graphs provides context for the design argumentation, while the limited quantity of text stored per node forms no obstacle as the structure of the argument is in the relationships between the nodes and the type of node, rather than in the text a node contains. Two secondary success metrics have been identified during the study:

- The percentage of design rationale nodes created during the geometry modelling phase
- The number of design rationale to geometry model connections created during the design process.

Section 4.2 describes why these metrics were selected, and the implementation of the software used to gather the data from the design rationale database. As the descriptive 1 and descriptive 2 studies both use the design rationale database analysis software by (Gorissen 2011), this section also covers the

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

'Specify data collection software' activity which occurs prior to the Descriptive II study in Computer-Aided Engineering Design Research Methodology (Bracewell, Shea et al. 2001). In section 9.2.1 the additional data collection software used for the Descriptive II study is discussed.

1.5.2 Descriptive 1 phase

The second stage of the DRM is the Descriptive 1 stage. The following aims can be identified for this part of the research:

- To create an understanding of the issues.
- To establish the state of the art in the field.
- To serve as a baseline before any changes in tools or methodology are applied.

The absence of quantitative studies of design rationale capture during mechanical engineering design, discussed in section 3.6.1, necessitates an experimental component to the prescriptive 1 study. For this reason the descriptive I study was split into two stages: a review of the existing literature to describe the state of the art for the current design process consisting of chapters 2 and 3, and a quantitative study of design rationale capture during a mechanical engineering design process which seeks to provide a baseline for design rationale capture during mechanical design processes and allows the identification of design stages poorly supported by the current toolset in chapter 4.

1.5.3 Prescriptive 1 phase

The Prescriptive 1 study can be divided into two broad phases: firstly the identification of an approach to resolve some of the issues outlined and secondly the definition of a prototype tool which seeks to implement the approach. Chapter 5 lays out an approach to support design rationale capture and use during geometry design while chapters 6, 7 and 8 focus on defining the required addition to the existing toolset and the creation of a software tool prototype.

1.5.4 Descriptive 2 phase

According to the design research methodology the purpose of Descriptive 2 study is to evaluate the changes made to the design process as a result of the Prescriptive II study. Chapter 9 is a descriptive 2 study in which the toolset described in chapters 6, 7 and 8 is used during a design process and the changes to the design rationale capture quantified.

The Methodology for Computational Design Tool Research proposed by Bracewell, Shea et al. (2001) calls for the definition of data collection software prior the experimental stage of the Descriptive 2 study, however the data collection and analysis software used in this particular case is largely identical to the software used for the Descriptive I study in chapter 4, with some additional functionality to measure the use of the design rationale linking tool, discussed in section 9.2.1.

1.6 Outline of the thesis

The thesis is structured as follows:

Chapter 1 describes the motivation and objectives of this thesis, as well as the research methodology used.

Chapter 2 will consider feature-based solid geometry modelling, define several geometry modelling terms, discuss the use of geometry modelling in the design process and the usage issues related to understanding geometry models during collaborative design and reuse.

Chapter 3 presents a review of design rationale, the evolution of the design rationale concept, a review of existing definitions and the current state of the art in design rationale editors. Specific attention is given to efforts to integrate design rationale capture and use into the design process. A review of the evidence for capture and usage of design rationale during mechanical engineering design processes completes this chapter.

Chapter 4 presents the results of an initial quantitative study into the capture of design rationale during the design process.

Chapter 5 describes the general approach to improve design rationale capture during geometry design by enabling the linking of entities in the geometry model to design rationale.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Chapter 6 deals with the practical implementation of a tool to link geometry model entities to design rationale.

Chapter 7 describes a tool, GRapher to create graphs of the structure of geometry models and their associated design rationale and the use of these graphs as means of understanding the underlying geometry model and structure and as a means of viewing the linked design rationale

Chapter 8 describes the tools for design audit which can be created based on the design rationale linking functionality described in chapter 6 and the geometry model structure graphing functions described in chapter 7. The tool gives a visual feedback of the status of the rationale through the colour of faces or features of the design

Chapter 9 details a further study of design rationale capture and a case study of how the tools described in the previous chapters could be used during the design process.

Chapter 10 presents the conclusions and future work.

2. Relevant works: geometry modelling

This chapter considers the functionality of geometry modelling in the design process. The focus is on the creation of detailed product definition geometry, rather than geometry creation for simulation, optimization or visualization purposes.

Section 2.1 discusses the role of geometry modelling in the design process, with a particular focus phases or tasks which are not well supported by the current generation of commercially available tools.

To begin a discussion on geometry modelling and knowledge management it is first necessary to define a set of terms to describe the structure of geometry models: see section 2.2 for a definition of terms and an analysis of the structure of geometry models.

Subsequently the usage of geometry models during the design process will be considered, with a particular focus on knowledge capture and difficulties of knowledge capture and retention in geometry models.

Finally section 2.4 discusses annotation of geometry models as a means of associating additional design information with entities in the geometry model.

2.1 Usage of geometry modelling during the design process

Since the first entirely 3D computer aided design process of the Boeing 777 in the early 1990's (Snyder, Snyder et al. 1998), computer-aided geometry modelling has become the industry standard. The increasing use of Computer Aided Design (CAD) and Product Lifecycle Management (PLM) tools has created an entirely digital product design environment.

Feature-based geometry modelling, also known as history based or operation based modelling is currently the dominant form of modelling (Bai, Gao et al. 2010), this is confirmed by user surveys of people working in industry (Robertson and Radcliffe 2009).

This section will review the motivations for the creation and usage of geometry models during the design process.

2.1.1 The role of geometry modelling in the design process.

Before discussing the use of geometry modelling during the design process it is necessary to define the stages of the design process. A four stage engineering development process can be distinguished (Pahl, Beitz et al. 2007 p. 128):

- Planning and task clarification,
- conceptual design,
- embodiment design
- detail design

Pugh (1990) proposes a similar set of stages, but merges embodiment and detail design into a single stage. For the purposes of this thesis embodiment and detail design will be considered to be one stage of the development process, as both of these stages tend to use geometry modelling. Figure 2 illustrates this process.

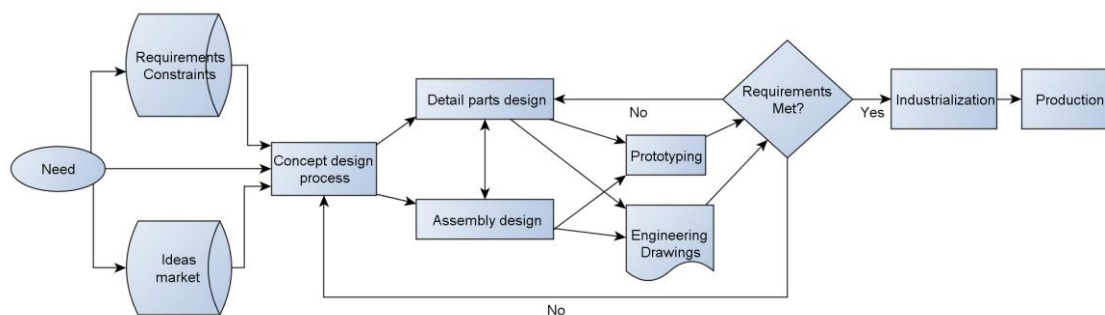


Figure 2 Mechanical engineering design process with approval based iteration, based on Pugh (1990).

The needs for creating a model or representation of design geometry are diverse; however the following main categories can be identified:

- To define the design's geometry.
- To use the definition of the geometry of a design as a means to instruct the manufacture of the part or assembly, either directly or by means of derived two dimensional representations of the geometry such as engineering drawings.
- To express and communicate design ideas or concepts.
- To define the variation of a design within the design space for optimization.

Each of these uses of geometry models and representations during the design process places its own requirements on the geometry, and requires different modelling approaches.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

During optimization the need for geometry representations as an input for simulation requires highly flexible geometry models to create a representation of the design space rather than a detailed definition. An example of this is the highly parametric UAV model discussed by Sóbester and Keane (2006).

If geometry modelling is used during concept design is mainly focussed on creating a representation of a potential solution in 3D form, rather than an explicit model with precisely defined geometry (Robertson and Radcliffe 2009). The main focus is on exploring potential configurations, rather than defining a precise shape. A physical equivalent would be a clay model or a sketch. In general if geometry models are created during concept design, they will not be carried over to the embodiment design stage. While there is evidence that concept design is poorly supported by geometry modelling tools (Robertson and Radcliffe 2009), that is outside the scope of the research presented in this thesis.

The main focus of this thesis is the use of geometry modelling to create and communicate the shape, dimensions and properties of the design. In this case geometry modelling is a subtask of embodiment and detail design. Figure 3 illustrates this functionality. Geometry models serve both as an input for design validation processes such as FEA and cost and weight control, but also as an input to CAM and as documentation of the shape of the design. Geometry models now are main method of communicating designs (Patel, Ball et al. 2009).

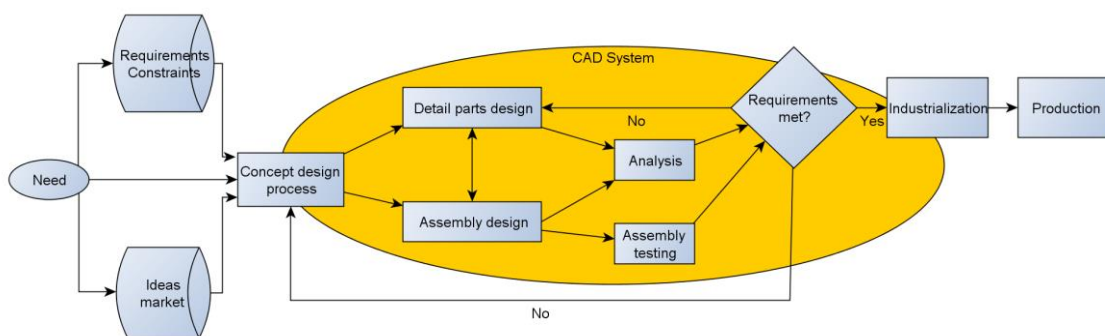


Figure 3 The use of CAD for geometry modelling during the design process.

Several distinct needs that motivate the creation of geometry models and representations can be identified:

- Define the properties of the artefact to be manufactured
 - Shape

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Dimensions and tolerances
- Assembly
- Materials
- Communicating the design
 - Within the design team as a part of collaborative design
 - With suppliers, manufacturing and maintenance teams
- As an input for further geometry-based design activities such as simulation for design validation and mass prediction
- A means to create two dimensional specifications of the design in the form of drafted engineering drawings.
- Long term storage of the specification of the design.

The editor of a parametric feature-based geometry model has three main objectives:

- To create a geometric representation of the design that can be used for further design activities.
- To create a model that adapts to changes in design parameters while retaining meaningful geometry.
- To structure the geometry model in such a way that model rebuilds or changes to parameters don't cause model failures.

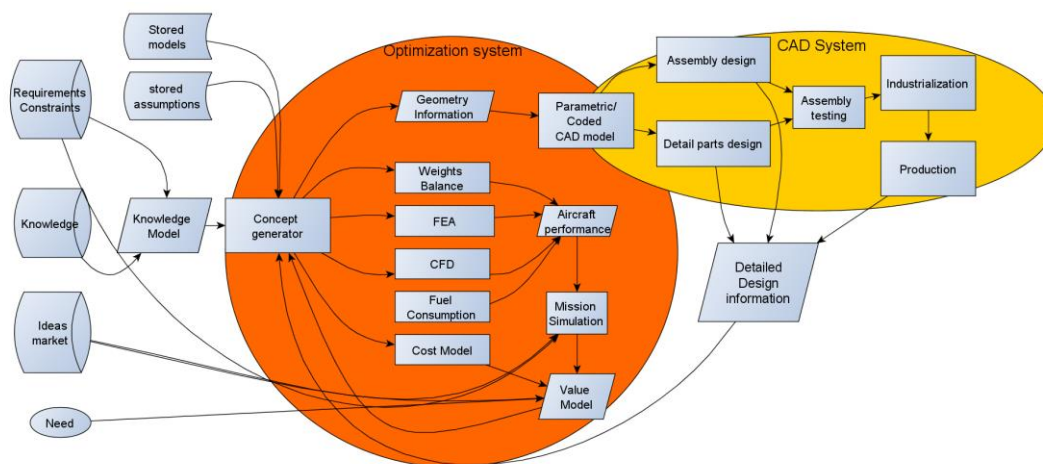


Figure 4 the reduced role of CAD geometry modelling in a KBE product development environment.

In the cases where a knowledge based engineering (KBE) design process is used, geometry modelling in a feature based CAD environment could be limited to the detail design phase. Figure 4 illustrates an example of such a product development cycle. This type of development process can be found in Gorissen, Quaranta et al. (2012, 2013) where the rough embodiment geometry was

defined by an integrated optimization system and an automated system used to generate a geometry in the CAD program with identical properties to the optimized shape.

2.2 Geometry modelling: Definition of terms and background

To precisely discuss the various aspects of geometry modelling it is necessary to define a precise terminology, as some of the terms in common usage can be misleading.

2.2.1 Computer Aided Design

The commonly used term Computer Aided Design (CAD) is not very specific and could apply to any design activity with computer support. In practical engineering parlance the term CAD is used for a suite of software design tools built around a geometry modelling functionality complemented by additional geometry based analysis tools such as FEA and CFD. CAD suites such as Catia V5, Solidworks and Siemens NX represent examples of commercially available toolsets.

The term geometry modelling tool will be used when the geometry modelling functionality is considered, while the term CAD will be reserved to designate the entirety of computer-aided design tools.

2.2.2 Constructive solid geometry modelling

Constructive Solid Geometry (CSG) (Shah 1991) also known as synthesis by features (Shah 1991), is a technique for the creation of solid, three dimensional representations of parts and assemblies. In constructive solid geometry modelling the successive additions or subtractions of geometry primitives, which can represent machining operations, are used to build up the model. Each modelling step is an operation which consists of a Boolean addition or subtraction of geometry primitives to the geometry (Shah 1991).

2.2.3 Boundary representation geometry

Boundary representation geometry, also known as B-rep geometry, lightweight geometry or dumb geometry, is a method for defining three-dimensional

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

geometries by means of the definition of their outer envelope or boundary of the solid (Zhang and Luo 2009). Common standards for the exchange of boundary representation models include IGES and STEP (Zhang and Luo 2009). However these methods lack design intent information in the form of construction structure, parameters and constraints (Zhang and Luo 2009).

STEP (ISO 1994) and IGES (Nagel, Braithwaite et al. 1980) have been defined by the International Association for Standardization (ISO) and National Bureau of Standards (NBS) respectively as a means to store and exchange geometry specifications. As STEP and IGES are not proprietary standards, the ability to view or edit models in these formats is not dependent on usage of a particular CAD suite. While boundary representation geometry models are more difficult to edit than feature based geometry models (Pratt, Anderson et al. 2005), they provide a more stable store of design information than commercial CAD formats (Patel, Ball et al. 2009).

Efforts are on-going to include design intent information in non-proprietary geometry models in the STEP format (Pratt, Anderson et al. 2005, Ball, Ding et al. 2008, Zhang and Luo 2009). However, the removal of design intent can be an advantage when sharing geometry with outside stakeholders, such as suppliers and collaborators, as the design intent can contain sensitive information.

To facilitate the editing of boundary representation geometry models a new generation of direct geometry editors have been brought to market, which facilitate the direct editing of boundary representation models by inferring design intent from the geometry. A notable example of a direct geometry editor is SpaceClaim¹ and direct editing technology has been integrated into CAD suites such as Siemens Solid Edge (Kurland 2008). At the moment tools such as SpaceClaim remain a niche product, targeted at concept design and preparation of geometry for simulation (SpaceClaim Corporation 2013).

2.2.4 Feature-based geometry modelling

Feature-based geometry modelling (FBGM) also known as history-based geometry modelling or Procedural geometry modelling (Pratt, Anderson et al. 2005) is the basis of most present day CAD suites (Ding, Davies et al. 2009).

¹ See <http://www.spaceclaim.com/en/default.aspx>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Feature based geometry modelling is a hybrid approach (Ding, Davies et al. 2009) which uses the step based, procedural, definition found in CSG as a means of defining the geometry and a boundary representation definition of the envelope of the solid.

The primary advantage of feature-based geometry modelling is that the user experience is centred on defining geometry based on functional relationships, unlike CSG or B-rep which are primarily a method of defining or storing the geometry itself. In feature-based geometry modelling the user defines a set of operations, which can reference the previously defined geometry to make a flexible model which can adapt relatively easily to subsequent changes.

In an idealized case the user defines the functional relationships between parts of the geometry and the software takes care of the rest: “Features encapsulate the engineering significance of portions of the geometry of a part or assembly” (Shah 1991).

For a detailed discussion of feature-based geometry modelling it is necessary to distinguish between features and operations. The term ‘feature’ will be used to refer to a portion of the geometry that is of design significance (Bai, Gao et al. 2010). A feature is a set of geometrical entities and is considered to contain rich semantic and shape information which relates the description of the artefact (Zhang and Luo 2009). When using this definition for a feature, both CSG and B-rep geometry can be said to consist of features. A given geometry can be divided into a set of features in multiple ways (Shah 1991, Chan 2007 p. 94).

In the context of feature based geometry modelling, an ‘operation’ is defined as a discrete step in the procedural order of the modelling tree, which may create one or more features. The modelling tree on the left of Figure 5 shows the operations used to create the geometry.

While most operations will create one feature, it is possible for an operation to create multiple features. However, when a modelling operation creates geometry which can be subdivided into multiple features this has a negative effect on operation retention during geometry model editing (Johnson and Diwakaran 2011, Diwakaran and Johnson 2012).

A feature-based geometry model can be edited by altering the procedural steps, or changing their order of operation. Subsequently to an alteration or addition of

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

the procedural model a regeneration of the geometry representation defined as a boundary representation takes place. This regeneration of the boundary representation model means that references to entities in the boundary representation model are not necessarily retained between geometry representations (Ding, Davies et al. 2009).

Feature recognition technology can be used to attempt to transform a non feature-based definition of geometry, such as an STL or IGES into a feature based geometry model. The current state of the feature recognition the technology deals well with prismatic geometries, however compound curves for example remain difficult.

Figure 5 shows a simple geometry model and, on the left of the image, the set of operations which create the model. Each operation can add to, subtract from or alter the previously existing geometry.

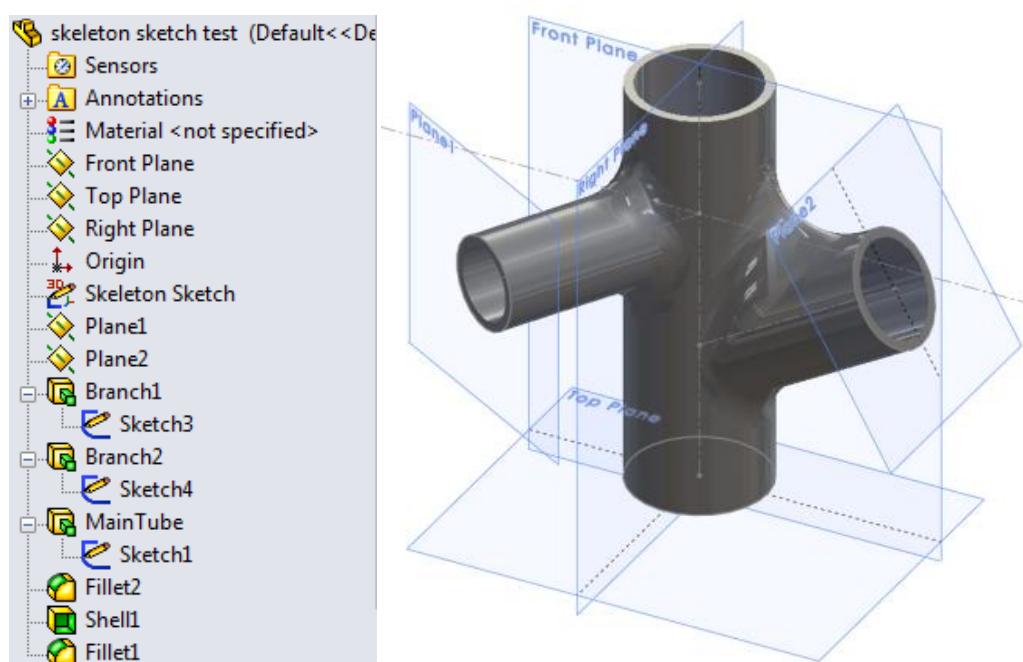


Figure 5 A simple geometry model consisting of an output geometry, shown on the right and the set of operations defining the model in the form of a modelling tree on the left.

In this work the term geometry model refers to a feature-based geometry model unless an explicit reference is made to another modelling technique.

2.2.4.1 Structure of feature-based geometry models

The succession of geometry creation operations is a linearly ordered set (Li, Zhang et al. 2010). However the underlying dependencies of the model are not necessarily a linearly ordered set. Figure 6 shows the structure of single modelling operation, where a distinction is made between the inputs for the operation, as chosen by the modeller, and the input in the form of the geometry as it existed before the operation. The order of operations is an important aspect to manage when editing a FBGM. The operations are in a hierarchical order of precedence; all the elements used to define an operation must exist earlier in the hierarchy. The features associated with the operation are the parts of the geometry which are created, removed or changed as a result of the operation: the difference between the pre and post operation geometry.

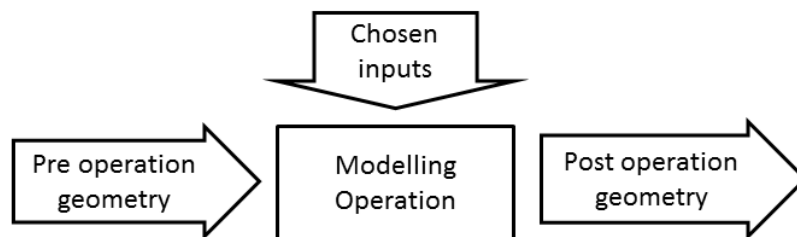


Figure 6 A modelling operation as a part of a block model.

If only the inputs chosen by the user are considered, it becomes possible to consider the structure of the model not as a linearly ordered set, but as a series of connections between operations. The structure generally is an acyclic partially ordered set (Li, Zhang et al. 2010), although a few cyclic structures can be found, notably with equations, but these often produce circular reference warnings, and should be avoided.

2.2.4.2 Reordering operations to obtain different results

To illustrate the separation from chosen user inputs and the existing geometry before the operation is applied consider the model in Figure 7. If features have no hierarchical constraints on their order they can be reordered in the feature tree, which may or may not change the geometry that the model produces. The features Fillet1 and Fillet2 shown in Figure 7 are commutative; reordering them doesn't change the model, as their areas of influence don't overlap.

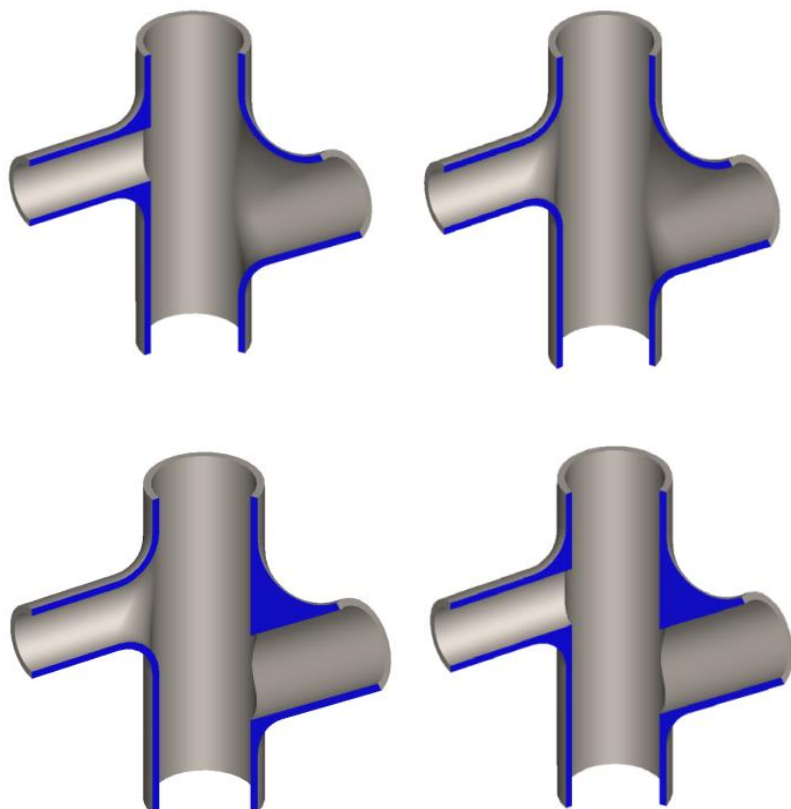


Figure 7 The four different possible geometries obtained by reordering the order of the three operations in the geometry model.

In the case of the geometries in Figure 7 the reordering of the two fillet features and one shell feature clearly produces a different geometry due to the different order in which features are applied.

2.2.1 Parametric and associative geometry modelling

The term ‘parametric geometry model’ will be used for a structured model used to create 2 or 3 dimensional representations of geometry, and where the output geometry can be changed by changing model inputs without having to recreate parts of the model. As is stated by Davies: “Parametric having come to refer to models that allow adjustment of certain values of a model and propagate consequential changes, without reference to the solution method used.” (Davies 2008 p 74.)

The geometry model is an intermediate model of the artefact (Davies 2008) between the input parameters and the output geometry. The term ‘geometry model’ will be used to refer to such an intermediate model, while the term

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

'geometry representation' will be used for the output geometry. Figure 8 shows the position of a parametric geometry model as an intermediate between the inputs and the resulting geometry.

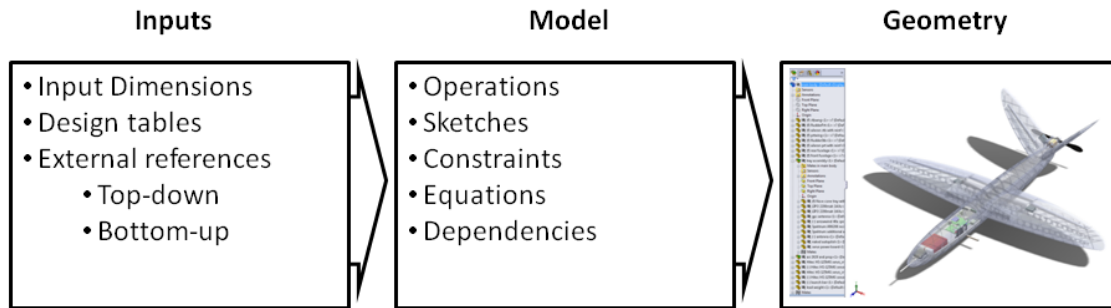


Figure 8 A geometry model viewed as an intermediary artefact between inputs and geometry representation.

A geometry model has to be of itself designed: the designer creates both a representation of geometry as well as a model that creates this representation. As such it is a model in the sense of (Buur and Myrup Andreassen 1989): “A model reproduces properties of an object.”

Parametric geometry models have an area of validity of input parameters for which the model produces meaningful geometry. Figure 10 illustrates the resulting loss of design meaning or intent when parameters are changed outside this area of validity. In practice highly parametric geometry models, which create good geometry for a wide range of input parameters are difficult to create and require substantial time, expertise and effort to model. Highly parametric geometry is therefore used mainly for variant design of highly structured products and for design search and optimization. In practice therefore the goal of most parametric geometry modelling is to create geometry models which are editable, which respond to relatively small parameter changes while keeping design meaning, but which don't have a defined and tested range of parametric flexibility.

Whereas the term parametric geometry modelling describes the change of a part as a result of changes to design dimensions, the term associative geometry modelling refers to the capability of geometry models to have references to another part, for example through an assembly, and change in geometry based on the changes made to the other part, rather than to changes made to input parameters directly.

2.2.2 Design intent

Design intent is a widely used term in geometry modelling. However, depending on context, it can be either understood as the designer's general intention or specifically as the underlying structure of a geometry model which creates parametric functionality. Several definitions will be reviewed and the definition of the term in the context of this thesis will be established.

Ullman (2002) gives the following definition for design intent:

“In the CAD community, the term intent is used to describe the ordering of geometric constraint equations in a parametric system. This ordering defines the geometric dependency needed by the system in order to make changes and is not necessarily the cognitive ordering that was followed by the designer in the development or refinement of the part or assembly”

Whereas Iyer and Mills (2006) define design intent as:

“Design intent is application, domain and context dependant knowledge that describes design space, represents design alternatives and process history, justifies design solutions and decisions and determines the characteristics of features and entities and the relationship among them.”

Considering that the broader definition proposed by Iyer and Mills (2006) includes virtually all design knowledge, a further definition specific to geometry modelling is given by Iyer and Mills (2006):

“Design intent contained in legacy CAD is the insight into the design variables (design objectives, constraints, alternatives, evolution, guidelines, manufacturing instructions and standards) implicit in the structural, semantic and practical relationships between the geometric, material, dimensional and textual entities present in the CAD representation.”

Design intent governs the relationships between modelling features in a part, and between parts in assemblies (Alducin-Quintero, Contero et al. 2011).

The following elements of the definitions above will be retained:

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Design intent is the correlation between the model structure and the functional structure of the artefact.
- The structure of relationships in the model, including equations, dependencies, parameters and constraints, which allow the model to adapt to changes to the model or model inputs.
- Not necessarily the same as the development history of the part.
- Design intent is the structure of the geometry model, and although this structure can model the variation of the design within the design space, it doesn't explain design decisions and is therefore separate from design rationale.

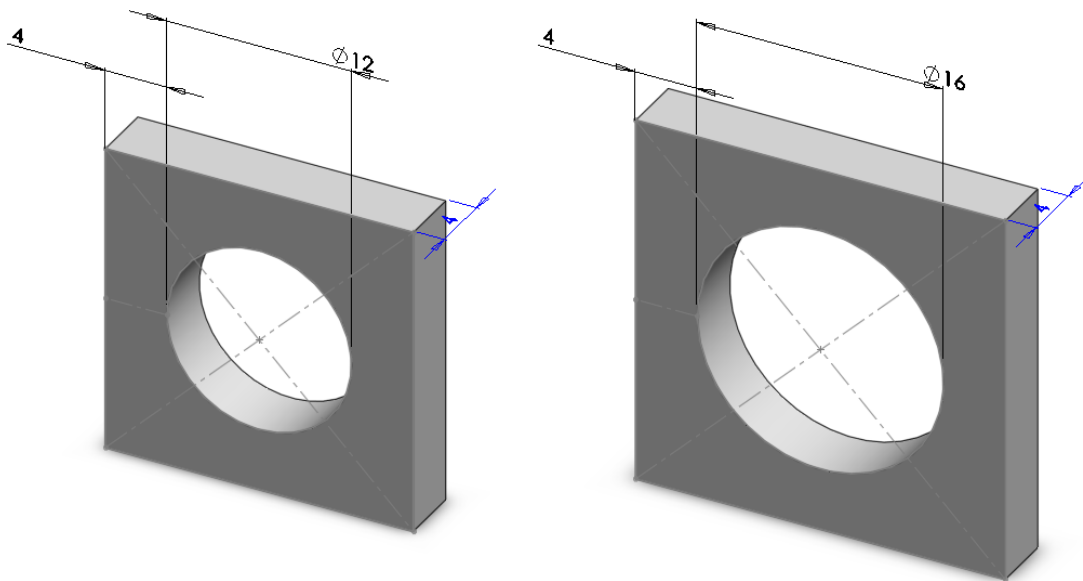


Figure 9 Design intent in a simple geometry model: geometry meaning is retained despite parameter changes

A key notion in the concept of design intent is the creation of a geometry model to reflect functional relationships present between elements of the geometry. As a consequence modifying one dimension in the design can result in change propagation which spreads to other parts of the geometry which have only an indirect functional relationship to the geometry directly driven by the dimension.

The principle of design intent is illustrated in Figure 9 and Figure 10. Both geometry models have an identical shape at the outset, but are defined differently. In Figure 9 the dimension of the square is defined as a distance from the centre hole, while in Figure 10 it is defined directly. Changing the dimension driving the size of the central hole of the part in Figure 9 and Figure 10 results in a different topology. The structure of the geometry model in Figure 9 reflects the design intent in the model in and ensures that the design meaning is not

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

affected while the change to the input parameter in of the geometry in Figure 10 leads to a loss of design meaning.

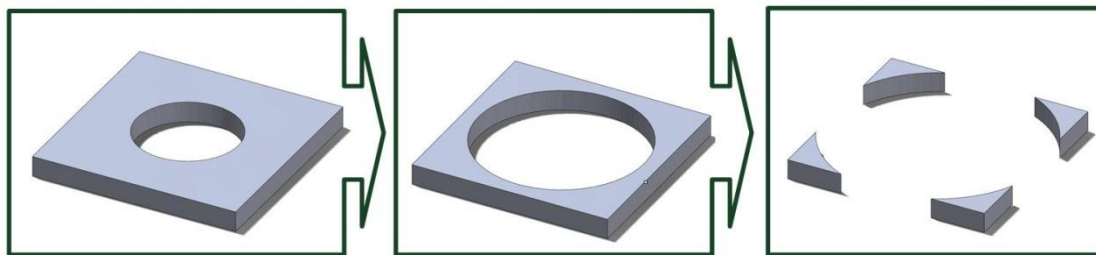


Figure 10 Changes to an input parameter of a geometry model which lead to a loss of meaning.

The term design intent is applied here to a parametric design example; however this principle can also be applied to the ordering of operations or sets of operations in the geometry model.

2.3 Usage issues encountered during the use of feature-based geometry models as a means of communicating designs.

Despite the utility of geometry models as both a model and a means of creating a specification of the design, several issues relating to their usage can be distinguished. These issues occur mainly during design reuse and communication, when engineers who were not involved in the development of the original product have to create an understanding of the product from the geometry model and design definition documentation.

The reuse of existing design and design ideas is common in industry and is an economic imperative, as it reduces design effort and risk (Keller, Eckert et al. 2009), However, it is important to understand the evolution and rationale of an existing design before it can be reused (Demian and Fruchter 2009). Three main categories of design reuse can be identified:

- Reuse of the entire design of a part without change, e.g. part libraries.
- Reuse of the geometry model or parts of the geometry model of previous designs.
- Use of existing design information concepts, or features.

While design reuse is attractive from an economic point of view there are a number of issues which make design reuse difficult:

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- The original design team may have dispersed, with people changing jobs, companies or retiring leading to a loss of product knowledge.
- The original computer hardware and software used to create the digital design specifications and geometry models may no longer be available.
- The project data, both paper and digital, may not have been curated effectively.
- The designers involved will have forgotten some of their personal information on the project.

As discussed in section 1.4, the work presented in this thesis does not consider the question of how to best archive digital product data, see for example (Ball, Patel et al. 2008) for a discussion. The following sections assume that the design re-users have full access to the digital models, documents and specifications present at the end of the original design process.

The reuse of geometry models is likely to increase due to the increasing number of available geometry models and tools to retrieve existing designs from repositories for example: Urbanic and ElMaraghy (2009) propose a framework for recovering mechanical components for design reuse while the reuse of parts of geometry model by searching for features in a design database is proposed by Bai, Gao et al. (2010). The reuse of geometry model of the design as a basis for design reuse is attractive:

- Removal of the costs associated with reverse engineering and remodelling the parts.
- The design intent of the geometry model can help the understanding of the functioning of the design.

However:

- The re-user has to understand the design of the geometry model as well as the design itself
- The accuracy and quality of the geometry models of the design is not always known.

The following sections will discuss issues with design reuse for cases in which the geometry model provides the principal source of design information.

2.3.1 Design rationale can be difficult to infer

The lack of information explaining design decisions has been identified by Busby (1999) as a cause of design reuse failure: “The difficulty of inferring a design’s rationale from the design itself cropped up a number of times as an explanation for reuse failure.” Design rationale for a part of a geometry can be especially hard to reconstruct when a geometry feature performs multiple functions (Busby 1999).

Chan (2007) identifies several problems in the reuse of existing designs: poor documentation of design decisions, and situations in which even the original designers have difficulty in recreating the design rationale for the product.

2.3.2 The product information recorded does not describe the full design

Geometry models contain mainly structural information, focused on the definition of the shape or state of an artefact, while semantic information explaining why the artefact is constructed this way is often absent (Lowe, McMahon et al. 2004). The current industrial toolset doesn’t capture the complete design process:

“Technical reports, the main digital corporate source for design information together with computer-aided design (CAD) and other models in product lifecycle management (PLM) systems, capture only fragments of design processes.” (Auricchio and Bracewell 2013b)

2.3.3 Design intent can be difficult to infer from the model

During design collaboration, there is a need for accurate and explicit interpretation of design intent among teams or designers (Pahl, Beitz et al. 2007) cited in (Li 2012). Understanding feature-based geometry models is difficult: Salehi and McMahon (2009, 2011) give the results of a study of 153 engineers and found that 76% were not able to find the right parameters and associative relationships in large and complex CAD parts and assemblies. These issues are worse for models created by others: 81% of engineers in the study agreed that it is quite difficult to change models of parts and assemblies created by other designers, while 86% thought it would be very helpful to have more information about the construction and structure of the parametric associative CAD parts and assemblies created by others.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Results of a user survey of geometry modelling users by Robertson and Radcliffe (2009) indicate that geometry modellers are disinclined to make changes to the geometry model structure even when it would remove or reduce design problems due to the high effort required to edit geometry models to propagate the changes.

The importance of including annotations with the geometry model when performing the digital curation of designs is emphasized by Patel, Ball et al. (2009).

2.3.4 Geometry models do not capture design alternatives well

While geometry models can precisely specify the variation of output geometry with respect to input parameters for one design, the alternative solutions to the design problem are not generally retained.

However, the knowledge of previously considered design alternatives could prevent the design re-user from exploring the same fruitless alternatives again (Busby 1999).

Approaches exist to show the different versions of the geometry model such as the history based system for documenting engineering design activities proposed by Conway and Ion (2013) and Demian and Fruchter (2009). These systems seek to display the evolution of editing of a geometry model with the view of helping to understand the evolution of the model and the alternatives considered.

2.4 Annotation and mark-up of geometry models as a design documentation tool

Annotation is the addition of information to an existing document, while markup is a subtype of annotation with a formal structure (Davies 2008 p. 98) An annotation consists of an anchor in an existing document, and an attached quantity of information, often in the form of text (Ovsiannikov, Arbib et al. 1999). Annotations have a long history as a form of in-context information storage, for example notes in the margins of books. A generic annotation is composed of two components, an annotation anchor and annotation content. (Li, McMahon et al. 2011)

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Although many present day CAD suites include a native geometry annotation function, a wide range of additional tools for the annotation of geometries, either in 2d or 3d form has been proposed. These tools cover a wide variety of geometry representations: lightweight 3d models (Ding, Davies et al. 2009, Hisarciklilar and Boujut 2009), secondary supports such as the annotation of geometry model screenshots (Hisarciklilar and Boujut 2007) and technical drawings (Bracewell, Gourtovaia et al. 2009, Keraron, Bernard et al. 2009) and feature-based geometry models (Alducin-Quintero, Contero et al. 2011, Sandberg, Lundin et al. 2013).

Sandberg, Lundin et al. (2013) propose a system which combines text and hyperlink annotations with a rule based KBE system as a means of capturing design knowledge during simulation driven design. The proposed system is implemented for the Catia V5 (Sandberg and Näsström 2007) and Siemens NX (Lundin, Sandberg et al. 2010, Sandberg, Lundin et al. 2013) CAD packages. Sandberg, Lundin et al. (2013) stress the importance of the designer capturing design knowledge during the design process and while editing the geometry.

Ding, Matthews et al. (2009) propose an XML based annotation system with the view of enabling the structured collection of in-service information for a product as annotations to the geometry model. Annotations are anchored to the geometry by means of persistent pointers or references. The XML document can be viewed separately from the geometry model, and in this way provides a record of the attached information independent from the geometry model. The annotations created by means of this system can be applied to the same geometry model stored in different CAD formats provided the persistent reference to the geometry has been retained, thus creating tool independence. Ding, Davies et al (2009) show that annotations created with this system to geometry in the Siemens NX CAD suite remain anchored to the geometry exported to a variety of lightweight boundary representation formats.

The work by Alducin-Quintero et al. (2011) is an example of an annotation functionality integrated into a feature-based geometry modeller, Siemens NX. These annotations are focussed on explaining the structure and design intent of the model, rather than the design decisions. Experimental use of this system shows that it speeds up editing of complex models during design reuse by 10-20%.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Davies (2008) proposes the annotation of feature based geometry models by means of attributes added to entities in the geometry model, a facility which is supported by the geometry kernel used, Parasolid (Siemens Product Lifecycle Management Software Inc 2008). The view is to create a mark-up system which can support multiple engineering viewpoints, most notably the annotation of CAD parts with boundary conditions for finite element analysis.

Li, McMahon et al. (2011) propose an ontology based annotation system, OntoCAD, for the annotation of boundary representation geometry models in the STEP format (Li, McMahon et al. 2011, Li 2012). This system is applied to integrate a product-cost ontology into the geometry model (Li, McMahon et al. 2011).

2.4.1 Motivation for annotation

A large number of tools and methods for the digital annotation of geometry models have been published for a large variety of reasons. The main motivations for annotation of geometry models and representations are laid out in Table 1. Where multiple motivations for annotation are given in one publication, each is presented in the table.

Table 1 An overview of the motivations for the annotation of geometry.

Motivation:	Occurrences in literature:
Clarify design intent	(Alducin-Quintero, Contero et al. 2011, Li, McMahon et al. 2011)
Contain design decision information	(Boujut 2003, Boujut and Dugdale 2006)
Link to formal design representations such as: <ul style="list-style-type: none">• ontologies• knowledge based engineering	<ul style="list-style-type: none">• (Li, McMahon et al. 2011, Li 2012)• (Sandberg and Näsström 2007, Lundin, Sandberg et al. 2010, Sandberg, Lundin et al. 2013)
Contain in-service product feedback	(Ding, Davies et al. 2009, Ding, Matthews et al. 2009, Ding, Ball et al.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

	2011)
Share ideas between stakeholders	(Boujut and Dugdale 2006, Davies 2008, Ding, Davies et al. 2009, Hisarciklilar, Rasoulifar et al. 2009, Li, McMahon et al. 2011)
Collaborative design	(Boujut 2003, Aubry, Thouvenin et al. 2007, Hisarciklilar and Boujut 2007, Lenne, Thouvenin et al. 2009)
Speed up geometry editing	(Alducin-Quintero, Contero et al. 2011)
Reduce communication ambiguity	(Boujut 2003, Hisarciklilar and Boujut 2007, Hisarciklilar and Boujut 2009)
Build or preserve knowledge	(Boujut 2003, Sandberg and Näsström 2007)
Contain or link to design rationale	(Boujut and Dugdale 2006, Hisarciklilar 2008, Hisarciklilar and Boujut 2009, McKay, Kundu et al. 2009)
Identify geometry entities for analysis	(Davies 2008, Ding, Davies et al. 2009)
Record design review outcomes	(Hisarciklilar and Boujut 2007, Hisarciklilar 2008)

In general the annotation of geometry models is motivated by the desire to include or attach additional information to the geometry model. Particularly the inability of geometry models to contain contextual information such as design decisions and rationale has led to a great amount of published work. The motivations for the annotation of product geometry can be categorized in two broad streams:

- To store and communicate design and model knowledge.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- To link geometry entities to design tools outside the geometry model, such as KBE systems, ontologies and simulation tools.

2.4.2 Geometry annotation anchor

The anchor of the annotation is the means of attachment to the document, in this case the geometry. Annotations can either be 'inline', where the content of the annotation is stored inside the document being annotated, or 'stand-off', where the content of the annotation is stored externally (Davies 2008 p. 138).

Most commercial feature-based CAD suites provide a built in annotation facility for in-line annotation (Ding, Matthews et al. 2009). For in-line annotation the type of data contained in the annotation is constrained by the annotation facility provided by the CAD environment (Ding, Matthews et al. 2009), in this case the formality of the annotation is generally low.

The use of stand-off annotations is preferred (Ding and Liu 2010) because:

- Multiple engineering viewpoints can be presented in different sets of annotations, which can be updated throughout the lifetime of the product.
- Standoff annotation does not increase the file-size of the original document while the annotation can contain rich design information such as design rationale and semantic context, as well as metadata.
- Stand-off annotations can be easily removed from the document enabling easy management of proprietary design knowledge.

Three principal factors can be identified concerning annotation anchors:

- The method which is used to anchor the annotation to the geometry.
- The choice of geometry representation to be annotated.
- The granularity of the anchor: how large a 'chunk' of geometry is identified, and how precisely this is delimited.

Each of these factors will be reviewed in the following sections.

2.4.2.1 Annotation anchoring method

The annotation anchoring method used is closely tied to the opportunities afforded by chosen geometry representation. For example as the boundary representation of a feature-based geometry model is regenerated after model edits (Ding, Davies et al. 2009), it is not possible to create an annotation by

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

means of a persistent reference to a face or edge in such a model. An overview of the methods of anchoring annotations can be found in Table 2.

Table 2 Type of annotation anchor used.

Type of anchoring method used:	Occurrences in literature:
3d pointers to locations in space	(Ding, Ball et al. 2011)
Persistent references for entities in the geometry model	(Ding, Davies et al. 2009, Ding, Matthews et al. 2009)
The use of attributes to operations or entities in the geometry	(Davies 2008)
The use of native 'in-line' annotation systems present in feature-based CAD systems.	(Sandberg and Näsström 2007, Lundin, Sandberg et al. 2010, Alducin-Quintero, Contero et al. 2011, Sandberg, Lundin et al. 2013)

2.4.2.2 Choice of geometry representation to annotate

Table 3 gives an overview of the existing annotation approaches and the choice of geometry representation used. The large similarity between the classification of Table 2 and Table 3 shows that the choice of annotation method is closely related to the choice of geometry representation.

Table 3 geometry annotation classified by choice of geometry representation.

Geometry type:	Occurrences in literature:
Boundary representation geometry	(Ding, Davies et al. 2009, McKay, Kundu et al. 2009, Li, McMahon et al. 2011, Li 2012)
Feature-based solid geometry	(Boujut and Dugdale 2006, Sandberg and Näsström 2007, Lundin, Sandberg et al. 2010, Alducin-Quintero, Contero

	et al. 2011, Sandberg, Lundin et al. 2013)
2D geometry representations (technical drawings, screenshots)	(Hisarciklilar and Boujut 2007, Bracewell, Gourtovaia et al. 2009, Keraron, Bernard et al. 2009)
Format independent approaches with 3d pointers	(Ding, Ball et al. 2011)
Format independent with persistent references	(Davies 2008, Ding, Matthews et al. 2009)

2.4.2.3 Annotation anchor granularity

The choice of granularity or how large a ‘chunk’ of geometry the annotation refers to is a recurring point of discussion. Two issues can be identified relating to annotation granularity:

- What is the appropriate level of annotation granularity.
- What level of annotation granularity is provided or supported by the geometry representation.

The use of geometry annotation for the automation of simulation as in Li (2012) and Davies (2008) requires the identification of a precisely defined area of the geometry, whereas explanatory annotation approaches targeted at the communication of the design between humans are less demanding. Li (2012) believe it is important for an annotation system to have the ability to refer to a wide range of geometry entities, ranging from vertices to edges to features bodies. The levels of annotation granularity of the ontology based system proposed by Li (2012 p. 180) are: assembly, body, face, edge and vertex, whereas the annotation system proposed by Davies (2008) which uses attributes as a means of attaching mark-up to feature-based geometry models, additionally allows the annotation of geometry creation operations.

The annotation approaches which refer to a point in space have only one level of granularity available, and it is up to the user to infer how large an area of geometry is referred to.

2.4.3 Annotation content

Three principal levels of annotation formality can be distinguished:

- **Informal:** generally textual annotations which have no formal structure.
- **Semi-formal:** annotations referencing semi-formal knowledge representations such as design rationale, discussed in further detail in section 3.5, and structured data formats.
- **Formal:** annotations referencing formal knowledge representations such as ontologies and knowledge based engineering tools.

Table 4 Level of formality of geometry annotations in literature

Level of formality:	Occurrences in literature:
Informal annotations	(Boujut 2003, Boujut and Dugdale 2006, Aubry, Thouvenin et al. 2007, Hisarciklilar and Boujut 2007, Ding, Davies et al. 2009, Hisarciklilar and Boujut 2009, Hisarciklilar, Rasoulifar et al. 2009, Sandberg, Lundin et al. 2013)
Semi-formal annotations	(Ding, Matthews et al. 2007, Ding, Matthews et al. 2009, McKay, Kundu et al. 2009, Lundin, Sandberg et al. 2010)
Formal annotations	(Sandberg and Näsström 2007, Lundin, Sandberg et al. 2010, Li, McMahon et al. 2011, Li 2012, Sandberg, Lundin et al. 2013)

Table 4 gives an overview of the level of formality of the annotation content found in the annotation literature. Certain of the published annotation approaches give no information on the content of the annotations, and are not included in the table, while some annotation approaches involve multiple levels of formality and occur multiple times.

2.5 Summary

This chapter defines general concepts related to geometry modelling and the structure of feature-based geometry models. It describes the usage of geometry models and the issues regarding knowledge management during geometry model reuse. Lastly it describes the state of the art of geometry model annotation, which has been created to support knowledge capture and communication by including or linking design knowledge to the model.

3. Relevant work: Design rationale

This chapter describes the state of the art of design rationale as well as the evolution of the design rationale concept since its inception.

Design rationale has had a long-standing interest from the engineering design community since the seminal paper by Kunz and Rittel (1970). Kunz and Rittel (1970) did not particularly envisage the use of design rationale for engineering design processes, but rather for political decision making.

Today's design rationale systems are based on Issue Based Information Systems (IBIS) proposed by Kunz and Rittel (1970) which was intended to support political and social decision processes, particularly for complex, 'wicked', problems (Rittel and Webber 1973).

Earlier representations of design spaces can be found such as the design decision diagrams used by Marples (1961) which can be considered a very early form of design rationale, while Toulmin arguments (1958) have a similar argumentation structure.

The initial application of IBIS as an aid to engineering design processes was predominantly in software engineering. For example Lee and Lai (1991) give a user interface design example, while Conklin and Begeman (1987) and Burgess Yakemovic and Conklin (1990) give an example of computer hardware and software requirements management and selection. Rationale capture during software engineering remains an area of work producing recent publications such as (Burge 2005, Burge and Brown 2008, Capilla 2009, Roldan, Gonnet et al. 2010, Shahin, Liang et al. 2010).

However a substantial portion of recent publications in the design rationale field are for mechanical engineering design, in particular for aerospace engineering, for example (Bracewell and Wallace 2006, Huet, Culley et al. 2007, Huet, McMahon et al. 2007, Bracewell, Gourtovaia et al. 2009, Bracewell, Wallace et al. 2009, Aurisicchio, Bracewell et al. 2012, Aurisicchio and Bracewell 2013b).

3.1 Design Rationale: definitions

To be able to discuss design rationale a few concepts have to be defined, particularly as some, such as the term 'design space', have a different meaning than in the wider engineering terminology. Design space will therefore be defined first. There are two aspects of design rationale which need to be defined:

- The information content of the design rationale
- The method of capturing and storing the design rationale

3.1.1 Design space

In the context of design rationale the term 'design space' has a related, but subtly different meaning from the general engineering parlance. Particularly in the context of computational design and optimization, where the term design space is used to designate a multidimensional volume of design parameters within the boundaries imposed by the design constraints. In the context of design rationale the term 'design space' is used to describe the alternatives; the 'space of possible solutions' (MacLean and McKerlie 1995) rather than formal constraints placed on the design solution. Creating a design space in the design rationale context implies a broadening search for possible solutions, rather than a search for a solution which respects the design constraints.

An example of a design space can be as follows: An engineer has identified a need to locally fix two metal plates to each other. The technical need for the connection has a set of attributes and constraints:

- Temperature range
- Expected forces on the connection in shear and axial directions
- Access for assembly
- Etc.

Several possible principal solutions can be identified, for example:

- Spot welding
- Riveting
- Bolting
- Etc.

And each of the possible solutions has a set of attributes and constraints:

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Maximal shear force
- Maximal traction force
- Stress resistance
- Corrosion resistance
- Ease of installation
- Ease of removal
- Cost
- Etc.

The design space is the totality of the needs and solutions considered for this particular connection, as well as the considerations which allow the engineer to come to a conclusion and specify one solution for the design problem.

3.1.2 Design rationale: content

Over the last 44 years a large number of design rationale structure paradigms such as IBIS (Kunz and Rittel 1970), COQ (MacLean, Young et al. 1991), DRL (Lee 1990) have been proposed. See (Lee and Lai 1992) for a comparative analysis of the different design rationale representations. Each design rationale paradigm has its corresponding definition of the design rationale concept, as summarized in Table 5.

Table 5 Definitions of design rationale found in literature

Definitions for design rationale:	Author:
"An explicit representation which allows us to describe a design space rather than a specific artefact."	(MacLean, Young et al. 1991)
"A design rationale is <i>not</i> a record of the design process – it is a co-product of the design along with the artefact and itself has to be designed."	(MacLean, Young et al. 1989b)
a description of a design space as a logical rather than a chronological account	(MacLean, Young et al. 1989b)
A historical record of the reasons for the choice of an artefact	(Burgess Yakemovic and Conklin 1990)

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

“Design rationale in the most general sense is an explanation of why an artefact is designed the way it is.”	(Lee and Lai 1991)
“A design rationale expresses elements of the reasoning which has been invested in the design of an artefact”	(Buckingham Shum 1996)
"Design rationale includes not only the reasons behind a design decision but also the justification for it, the other alternatives considered, the trade-offs evaluated, and the argumentation that led to the decision."	(Lee 1997)
“In our approach, design rationale means statements of reasoning underlying the design process that explain, derive, and justify design decisions”	(Fischer, Lemke et al. 1991)

In this thesis the definition of a design rationale as being a representation of the design space for an artefact, including design reasoning, argumentation and alternatives considered.

3.1.3 Design rationale: method of storage

The definition of design rationale as a representation of the design space of an artefact does not address the form in which this representation is stored. For the design rationale representations derived from IBIS (Kunz and Rittel 1970) design rationale take the shape of a semi-formal graph, called a map. A design rationale map takes the form of a directed graph in which some nodes take the form of issues which are linked by edges to other nodes representing potential solutions which in turn are linked to nodes containing arguments for or against these solutions (Bracewell and Wallace 2003).

Three main classes of objects in a design rationale can be identified:

- **Maps** a container for a design rationale graph
- **Nodes** representing Issues, Answers and Arguments

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- **Edges** creating relationships between nodes

The graph containing the design rationale can either be displayed as a tree based view, or laid out graphically.

The example design rationale map in Figure 11 shows a design rationale graph for some of the design space discussed in section 3.1.1. This illustrates one of the key differences between a design space and a design rationale: the design space includes all the elements of the design which have been considered and used to come to a design decision, while a design rationale is a representation of those design arguments which have been captured by the designer.

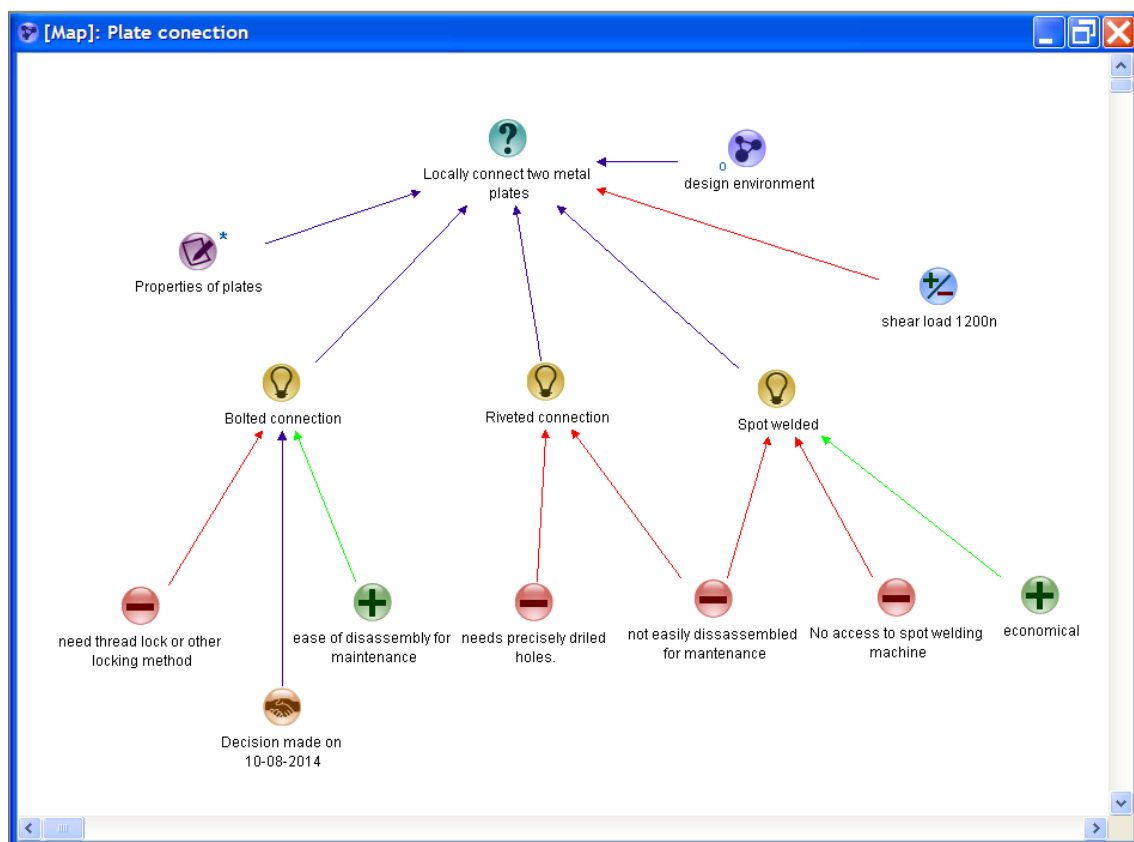


Figure 11 an example design rationale map

The distinction between a design rationale and a design space can be defined as follows: A design space is the totality of solutions, constraints and combinations of solutions which were considered by the design team to meet a particular need; while a design rationale is a method of recording and storing such a design space for as a means of developing the decision making process and for future use.

3.2 Why capture design rationale

The goal of design rationale in the form of IBIS as proposed by Kunz and Rittel (1970) was to aid decision making processes for problems of a political or social nature by providing a documentation format in the form of an argumentation structure. In the initial view the benefit of design rationale creation was primarily to assist decision making processes by providing a means to lay out argumentation concerning an issue. In an engineering design context the potential benefits of design rationale capture can be divided into those occurring during the design process and those that occur after the design process. Capturing design rationale is a time consuming activity, and as is stated by Buckingham Shum, Sevin et al. (2006):

“No designer can be expected to altruistically enter quality design rationale solely for the possible benefit of a possibly unknown person at an unknown point in the future for an unknown task. There must be immediate value.”

However, capturing the design rationale can aid decision processes (Lee 1997) and “... *the principal role of an engineer, in the design of an artefact, is to make decisions*” (Bras and Mistree 1991), and design rationale can serve both as an aid to decision makers and a documentation of the design process (Lee 1997).

It is possible to distinguish three areas which could benefit from the capture and use of design rationale:

- **Immediate:** the creation of the design rationale aids the decision process and improves the decisions made, leading to immediate benefits to the designer.
- **Intermediate:** benefits are realized during the design process due to improved awareness and communication of decisions and standpoints within the design teams.
- **Long term:** Design reuse is facilitated when design rationale is present because of improved understanding of design decisions made previously.

There is some evidence for the utility as both a means of communication design information and a means of improving the design, for example, it is easier and quicker to understand the identical information in design rationale rather than report form (Bracewell and Wallace 2003). The cost of design rationale capture is

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

easily offset by the increase in the ability to foresee design decision consequences (Burgess Yakemovic and Conklin 1990).

In an engineering design context there are three main benefits to be drawn from design rationale (Eng, Bracewell et al. 2009):

- Creating a coherent view of the information available.
- Improving the design thinking and decision processes.
- Improving documentation to facilitate design reuse and design reviews.

Design rationale helps designers (Fischer, Lemke et al. 1991):

- Improve their own work
- To cooperate with other people holding stakes in the design
- To understand existing artefacts

Buckingham Shum (1996) lists the following benefits from design rationale capture:

- structuring and analysing novel design problems
- keeping track of decisions
- communicating design reasoning within projects
- maintaining consistency in decision-making
- tracking progress in projects and identifying recurring and unresolved issues
- supporting the building of cumulative design knowledge, through reusing design rationale
- assisting the integration of perspectives from multiple stakeholders on decisions

Ball, Lambell et al. (2001) emphasise the assistance design rationale can provide during design reuse and propose a system to retrieve existing design rationale to support design reuse. Considering that most design is design reuse (Pahl, Beitz et al. 2007) the importance of the potential benefits gained from the additional understanding of the design space for previous designs due to the presence of design rationale should not be underestimated. Understanding which alternatives have been considered and why they have been rejected can help avoid design reuse failure (Busby 1999).

The short and intermediate term benefits of design rationale capture can be summarized as follows:

- Communicate a shared view of the design space and the design decisions which have been made within the design team.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Improve and clarify design thinking.
- Serve as a personal memory of design factors considered and of advantages or disadvantages of potential solutions.

While the long term benefits can be:

- To facilitate design reuse and review.
- Create an understanding of the alternatives considered.

3.2.1 The costs versus the benefits of design rationale capture

The costs and benefits of design rationale capture have been frequently discussed in literature; however, no economic data is available to support these analyses. Conklin and Begeman (1988) make the distinction between the cost of capture and the cost of usage of the design rationale. For example: the cost of automated design rationale capture methods such as video recording is low, but the lack of structure of the resulting data makes retrieving precise information difficult and costly, therefore the cost of capture is low but the cost of usage is high. Conklin and Burgess Yakemovic (1991) believe that the cost of creating rationale is more than offset by the gains made by allowing the designer to document thought processes and in this way expand the working memory available to the designer. In this case the design rationale creation costs are offset by the improvement in quality and value of the design. Buckingham Shum and Hammond (1994) discuss the possible utility of design rationale offset against the cost of creating it, and note that in some cases the design rationale capture seems to have been an impediment to the design work.

The difficulty of design rationale capture is that the cost of capture is paid up front, and that even if the long term benefits may be beneficial on a company level, there is little incentive for individuals to capture design rationale now, unless they can expect sufficient short term or long term return on the investment in time required.

The following means of improving design capture during the initial design process can be identified:

- Lower the cost of capturing design rationale by better tools
- Improve the utility of design rationale in the short and medium term so that the persons who initially capture the design rationale have more benefits from doing so.

- Provide external incentives for capturing design rationale, such as design rationale capture targets and audits.

3.3 Design Rationale argumentation structures

A number of different argument structures have been proposed for design rationale capture. The question of which argument structure is the most suitable for recording and aiding engineering design processes is debatable in the absence of any comparative trials. However, since the introduction of gIBIS (Conklin and Begeman 1987), in which the argument structure has a graphical layout, rather than a tree-based structure, the need for a fixed argument structure has decreased, as the graphical layout allows the creation of ad hoc and hybrid argumentation structures.

The following sections will review the most important design rationale argumentation structures.

3.3.1 Toulmin arguments

Toulmin (1958), who calls his reasoning-structures arguments, proposes to structure an argument as follows: starting from a Datum we use a Warrant to arrive at a Conclusion, unless an exception applies. The diagram in Figure 12 illustrates this reasoning process.

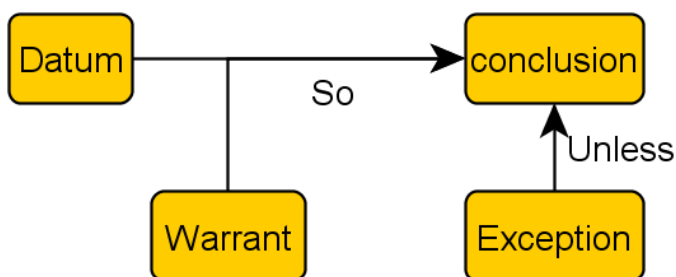


Figure 12 The structure of a Toulmin argument

3.3.2 Issue Based Information Systems

The IBIS structure as proposed by Kunz and Rittel (1970) has a different approach. It starts with an Issue, in the form of a question, which is then

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

followed by one or more positions. Then arguments are applied for or against a position. See Figure 13 for an example graph.

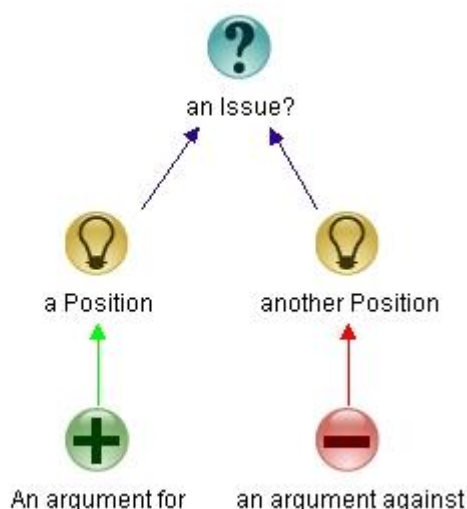


Figure 13 A design rationale using the IBIS argumentation structure created in Compendium.

The IBIS argumentation structure is at the heart of most present day design rationale systems such as: DRed (Auricchio and Bracewell 2009) and Compendium (Selvin, Buckingham Shum et al. 2001). IBIS arguments can be stored either as indented text IBIS (itIBIS) (Burgess Yakemovic and Conklin 1990) or as graphical IBIS (gIBIS) which is a graphical development of IBIS (Conklin and Begeman 1988, Conklin and Burgess Yakemovic 1991).

3.3.3 Questions, Options, Criteria (QOC)

The Questions, Options Criteria (QOC) design rationale format as proposed by MacLean, Young et al. (1989a) as a method for mapping a design space. Design rationale is laid out graphically, with solid lines representing a positive relationship and dashed lines representing a negative relationship. See Figure 14 for an example.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

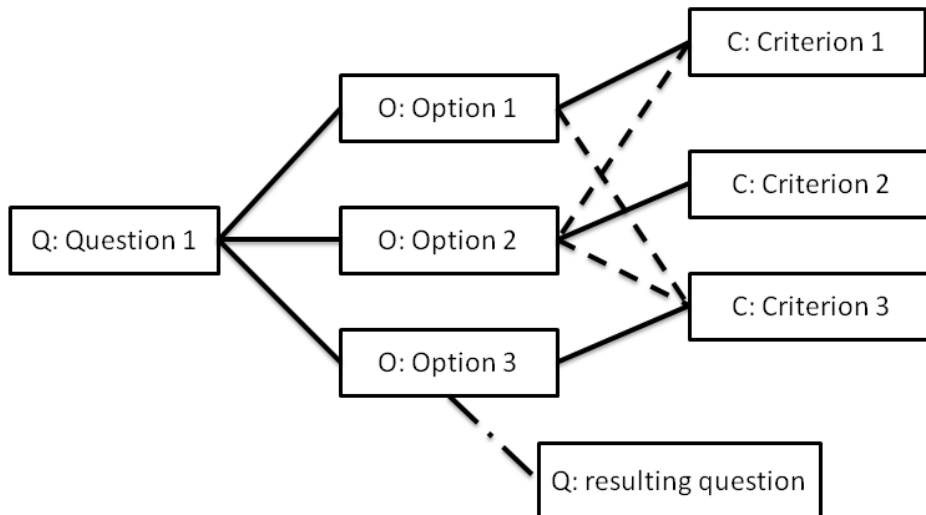


Figure 14 A QOC design rationale diagram, based on (MacLean, Bellotti et al. 1993).

While the structure of QOC is similar to IBIS the intention of QOC is to record the design space rather than to record the design process, hence the emphasis on the alternative Options in the design which mark loci in the design space, as well as the criteria used to select between these options (MacLean, Young et al. 1989b).

During the design process the goal of creating QOC design rationale maps is to perform 'design space analysis' which is intended as a multidisciplinary evaluation of the choices and argumentation related to the specification of an artefact (MacLean, Young et al. 1991, MacLean, Bellotti et al. 1993, MacLean and McKerlie 1995).

Ball, Lambell et al. (2001) believe the QOC structure is a better reflection of the design process than an IBIS based rationale.

3.3.4 Decision Representation Language (DRL)

The Decision Representation Language (DRL) (Lee 1990, Lee and Lai 1991, Lee and Lai 1992) is significant for the inclusion of goals and decisions in design rationale. The use of goals or objectives and decisions allow the design decision loop to be closed: a decision is recorded as being made which satisfies a goal. Burge (2005 p. 64) believes DRL to be the most comprehensive design rationale notation.

3.4 Design rationale capture and editing tools

The general problem with capturing rationale is the balance between the cost and intrusion of capture, and the structure and ease of searching the design rationales produced (Buckingham Shum, Selvin et al. 2006). The capture of design rationale can be performed by the designer, another entity such as an external facilitator or computer program, or by a combination of the two.

Five capture methods can be identified (Lee 1997):

- Reconstruction: the creation of design rationales by inference from the design specification, interviews with the original designers and personal knowledge.
- Record and replay: the design rationale is captured chronologically as it unfolds by the capture of video-conferencing or email discussions
- Methodical by-product: the design rationale is captured as a by-product of the methods used in the design process.
- Apprentice: A system observes the designer and asks questions when it doesn't understand, or agree with, the designer and generates a design rationale based on these interactions.
- Automatic generation: A system observes the designer and generates design rationale, but doesn't interact with the designer.

Design rationale captured by computers without user input is generally more focused on design reuse and seeks to minimize intrusion on the designer as a result of the rationale capture process, see section 3.4.3 for a discussion.

3.4.1 Graphical design rationale editors

While the original proposal for IBIS does contain a graphical 'decision map' (Kunz and Rittel 1970), the early digitally stored design rationales were in the form of indented text such as in the case studies by Burgess Yakemovic and Conklin (1990). Since the introduction of a graphical layout of design rationale with the introduction of gIBIS (Conklin and Begeman 1987) most design rationale editors have been graphical, although the Seurat system proposed by Burge (2008) is a notable exception.

Two principal graphical design rationale editors can be identified as representing the current state of the art: Compendium² (Selvin, Buckingham

² <http://compendium.open.ac.uk/institute/>

Shum et al. 2001) and DRed³ (Bracewell and Wallace 2003). The following two sections will consider the salient features of both the Compendium and the DRed design rationale editors.

3.4.1.1 Compendium

Compendium⁴ is an open source graphical design rationale editor (Selvin, Buckingham Shum et al. 2001, Buckingham Shum, Selvin et al. 2006) which is based on the IBIS design rationale argumentation structure (Selvin, Buckingham Shum et al. 2001), as well as the gIBIS graphical layout of design rationale (Conklin, Selvin et al. 2001). The Knowledge Media Institute (KMI) of the Open University was the host of a substantial part of the development of Compendium, but although development of Compendium at the Open University has ceased, the open source nature of the Compendium project has allowed development to continue under the name CompendiumNG⁵ (Boell 2013).

Compendium is intended as a general tool for knowledge capture during meetings (Selvin, Buckingham Shum et al. 2001) and has been used in a wide variety of contexts, including civil engineering (Eng, Marfisi et al. 2011) and software architecture design (Shahin, Liang et al. 2010). Compendium maps are an evolution of the gIBIS and QOC design rationale representation paradigms (Buckingham Shum, Selvin et al. 2006). Meetings are seen as key events during which to capture design rationale using Compendium (Conklin, Selvin et al. 2001, Selvin, Buckingham Shum et al. 2001). During the design meetings, the presence of a facilitator is suggested as a means to capture design rationale and integrate the use of design rationale into the meeting.

Figure 15 gives an example Compendium map, which is part of the Compendium documentation. A typical Compendium map for a design problem is shown in Figure 23.

³ <http://www-edc.eng.cam.ac.uk/projects/designrationale/>

⁴ <http://compendium.open.ac.uk>

⁵ <http://compendiumng.org/>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

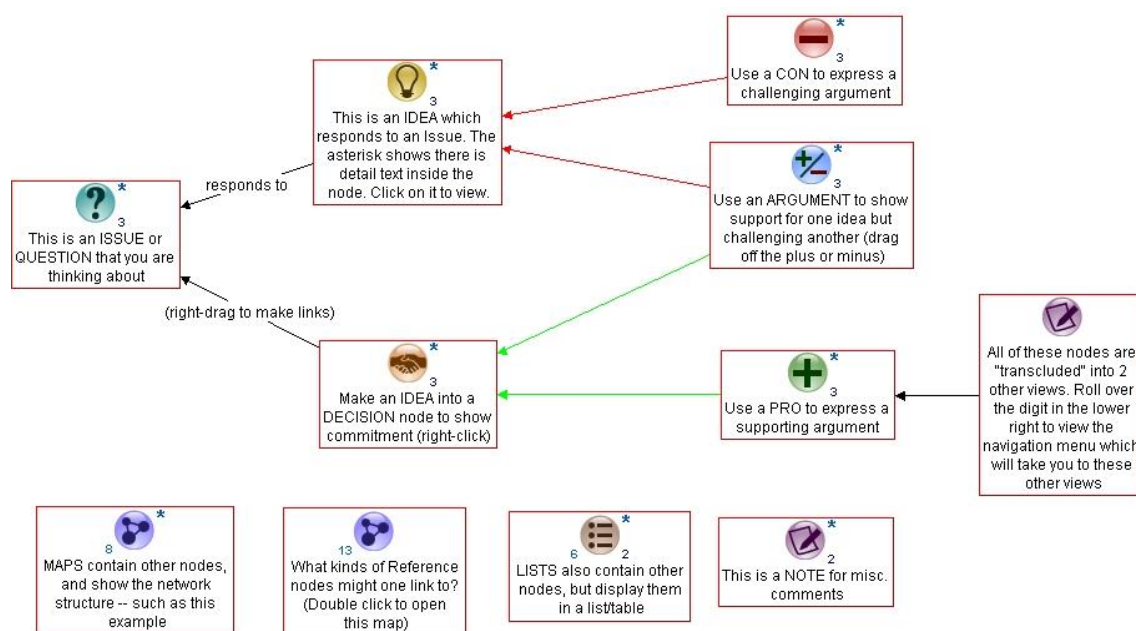


Figure 15 Compendium node types and argumentation structure, an example from the Compendium user instructions.

Compendium nodes and maps can be transcluded, which is the occurrence of one node in more than one map. In the case of the transclusion of maps this can be used to create a reference to a rationale map which discusses a related issue.

Compendium includes a facility for the annotation of text as well as the linking of text to an knowledge ontology (Selvin, Buckingham Shum et al. 2001).

3.4.1.2 DRed (Design Rationale editor)

DRed is a design rationale capture tool created to aid aeronautical engineering design at Rolls-Royce (Bracewell, Gourtovaia et al. 2009). As such it is the only design rationale editor which was created explicitly to support mechanical engineering design processes. Although there is no quantitative data available on the usage of the DRed tool, its use at Rolls-Royce is well established, as is shown by the number and time span of the papers published, as well as its inclusion in the companies' standard toolset (Bracewell and Wallace 2003, Bracewell, Ahmed et al. 2004, Bracewell and Wallace 2006, Bracewell, Gourtovaia et al. 2007, Aurisicchio and Bracewell 2009, Bracewell, Gourtovaia et al. 2009, Bracewell, Wallace et al. 2009, Aurisicchio, Eng et al. 2011, Armstrong, Bracewell et al. 2012, Aurisicchio, Bracewell et al. 2012, Aurisicchio and Bracewell 2013a, Aurisicchio and Bracewell 2013b).

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

DRed is an IBIS derivative (Auriscchio and Bracewell 2009), and is based on the Graphlet⁶ graph creation toolset (Bracewell and Wallace 2003). It is owned and controlled by Rolls-Royce Plc. (Auriscchio and Bracewell 2013a). Figure 16 shows the design rationale node types, which are designed to be a background to the text, and convey information on both node type and status. DRed icons are a background to the text and exist mainly to convey information on the text that they contain. See (Salustri, Weerasinghe et al. 2007) for an overview of DRed Icons. Each node class, e.g. issue or answer, having more than one icon to convey information on the status of the node, such as open, accepted or rejected.

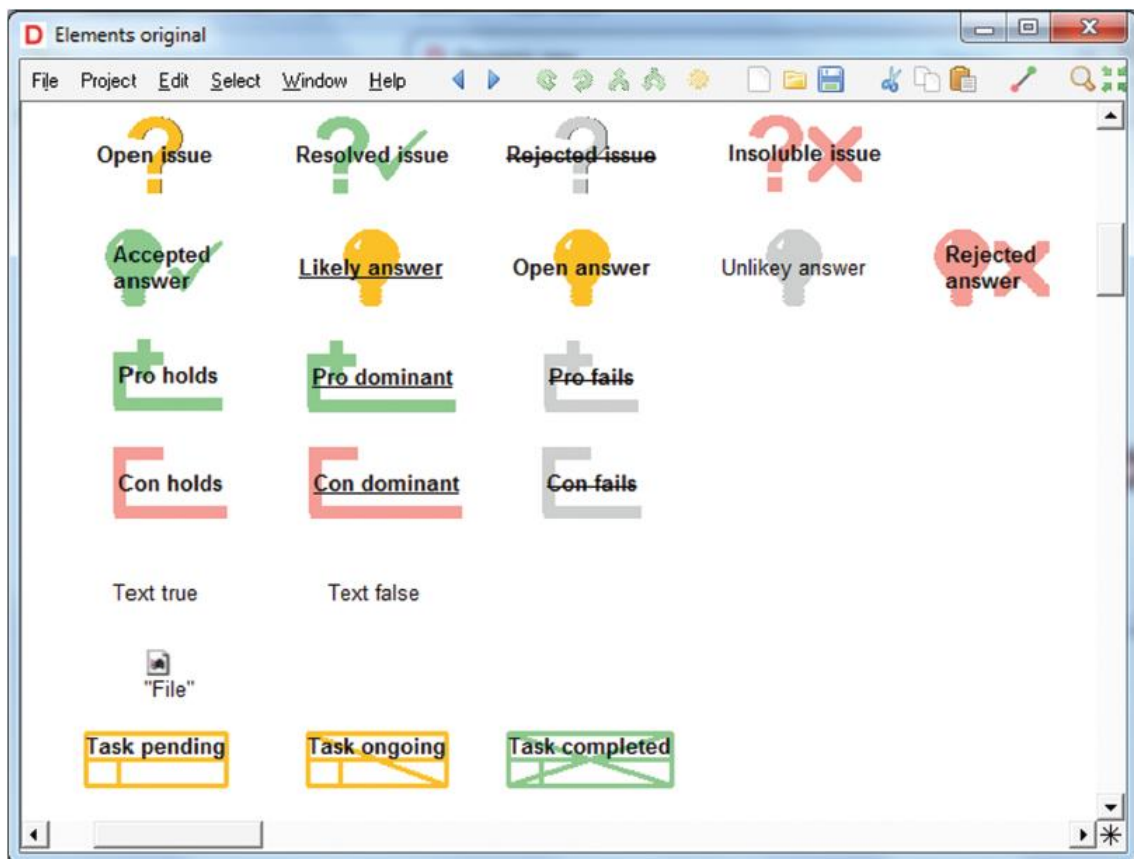


Figure 16 DRed node types, which include node status, from (Auriscchio and Bracewell 2013b)

DRed design rationale charts are stored in a file based format, as opposed to the database format used by Compendium. The file based format is perceived as being more user friendly and easier to integrate with PLM systems, although it

⁶ www.sai.msu.su/sal/E/2/GRAPHLET.html

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

does make linking between rationale maps more difficult (Bracewell, Wallace et al. 2009).

DRed uses tunnelling links (Bracewell, Ahmed et al. 2004), which fulfil the function of links between design rationale charts, in the same manner that transclusion is used in Compendium. Furthermore external links to and from DRed diagrams can be made, e.g. to Microsoft office documents (Bracewell, Courtovaia et al. 2007).

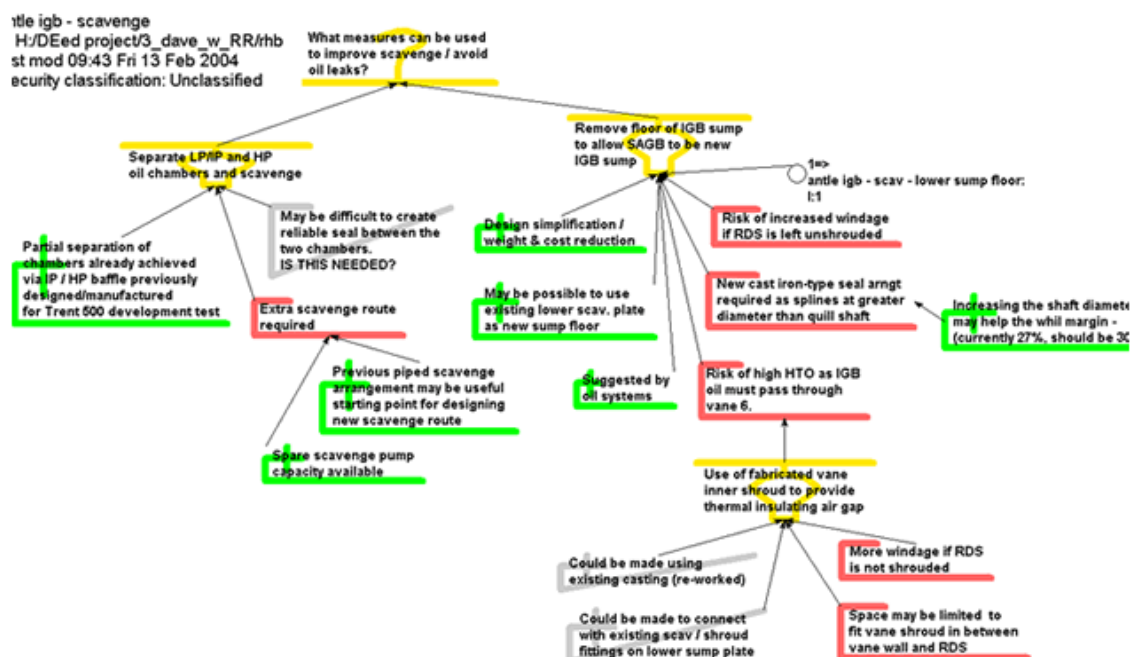


Figure 17 a typical DRed map, from <http://www-edc.eng.cam.ac.uk/projects/capturetools/>

The graphs created are explicitly intended to be printable, therefore all information contained in nodes is visible and there is no hidden text (Bracewell and Wallace 2003). The status of the each node is defined explicitly (Bracewell and Wallace 2003). The explicit definition of node status makes it possible to do a more elaborate checking of the rationale syntax for logical consistency (Bracewell, Wallace et al. 2009), an extension of the rationale checking functionally described by Conklin and Burgess Yakemovic (1991).

Recent developments of the DRed tool have focussed on mapping product interactions and functions in the form of a Function Analysis Diagram (FAD) (Auricchio and Bracewell 2009, Auricchio, Eng et al. 2011, Auricchio, Bracewell et al. 2012, Auricchio and Bracewell 2013b).

3.4.2 Design rationale capture in software design

Since the early case studies of rationale capture during software design, such as (Conklin and Begeman 1988, Burgess Yakemovic and Conklin 1990, Conklin and Burgess Yakemovic 1991), further advances have been made to support design rationale capture during software design.

The Seurat system (Burge 2005, Burge and Brown 2008) allows the capture of design rationale during software design and implementation, with a view of supporting software maintenance. The Seurat system is based on the DRL design rationale representation (Burge 2005 p. 64), and is available as an add-in to the Eclipse integrated development environment (IDE). Seurat includes comprehensive checking functions of design rationale for logical consistency (Burge 2005 p. 90).

Roldan, Gonnet et al. (2010) describe a combined versioning and rationale system for software architecture as a means of capturing and tracing engineering design processes.

3.4.3 Automated rationale capture

Automated design rationale capture systems integrated in the geometry modelling system could provide a solution to improve design rationale capture during geometry design, if able to capture the design process in sufficient detail, as they are not dependent on efforts of the user to capture design rationale, and therefore can capture design rationale at low cost.

The existing automated design rationale capture methods can be defined as being either 'record and replay' or 'automated generation' in the classification by Lee (1997). As stated by Lee (1997), automated design rationale capture methods by their nature fail to support decision processes.

In the context of geometry modelling two types of automated design rationale capture can be distinguished: firstly connecting design information to geometry, and secondly inferring design intent from the editing of the geometry model.

3.4.3.1 Record and replay

The system proposed by Mix, Jensen et al (2010) records Skype⁷ Voice over IP (VoIP) conversations and instant messages between the geometry model editor and other engineers or stakeholders and relates these to features in the geometry edited by the engineer as a result of these conversations, and allows these data sources to be retrieved later as a form of design documentation.

3.4.3.2 Automatic Generation

Automatic design rationale generation involves a design rationale system observing designers during the design activity and inferring design rationale from the interactions with the computer program used. Attempts have been made to extract rationale from the geometry editing process by recording a procedural record of model edits (Myers, Zumel et al. 2000), however these approaches do not aid the designers with their design task, and only information which is explicitly expressed during the design process can be captured. However, these systems may aid future designers who are reusing the designs.

A system to track and analyse design changes during an editing and simulation cycle of a very simple geometry model, with the view to extract design rationales from is proposed by Sung, Ritchie et al. (2011) and is applied to the capture of design rationale during iterative variant design. The extraction of design rationale during electrical harness design editing in a virtual reality environment, using a system to interact with the model described in (Robinson, Ritchie et al. 2007), is described by Ritchie, Sung et al., this system asks questions concerning design edits to the designer to clarify the motivations for the design changes (Ritchie, Sung et al. 2008, Sung, Ritchie et al. 2009). However both these systems are very specific to a particular design activity and the extension of this tool to creative design activities, which do not follow a predefined pattern, seems difficult.

It is dubious that the output of the current generation of automated design rationale capture tools is a representation of a design space as defined by MacLean, Young et al. (1989b, 1991).

⁷ www.skype.com

3.5 Integrating design rationale capture functionality into existing design tools

The utility of further integration of design rationale systems with existing design tools has long been widely recognized: Reeves and Shipman (1992) and Fischer, Lemke et al. (1991) call integration of design rationale in design tools. Hooey and Foyle (2007) identify requirements for a design rationale capture system to support complex system design at NASA which includes the ability to integrate well with other design tools.

The integration of design rationales in the design process and tools is essential: “A design rationale system is not effective as a standalone system.” (Regli, Hu et al. 2000)

The analysis of DRed design rationale maps shows that rationale nodes often refer to parts of products or processes (McKay, Kundu et al. 2009). The need to integrate design rationale capture functionality within CAD suites has been long recognized, for example by Lee (1997), Regli et al. (2000), Chan (2007) and McKay et al. (2009).

Numerous examples can be found for the integration of design rationales in existing design tools. The integration of design rationale capture and editing tools in an Integrated Development Environment (IDE) is proposed and implemented by (Burge 2005, Burge and Brown 2008). The annotation of images, geometry model screenshots or drawings in the DRed design rationale editor is a frequent occurrence, particularly for Function Analysis Diagram (FAD) (Auricchio and Bracewell 2009, Auricchio, Eng et al. 2011, Auricchio and Bracewell 2013a). The linking of design rationales to other information supports such as Microsoft office documents is described by Bracewell, Gourtovaia et al. (2007).

The inclusion of design rationales in PLM systems is described by Pavkovic, Bojetic et al. (2010), who propose the inclusion of bidirectional links between IBIS diagrams and entities in the PLM system as a means of capturing relevant design rationale. Huet, McAlpine et al. (2009) propose the inclusion of concept design sketches in PLM systems as a means of supporting design re-users in understanding the design process.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Design rationale should be recorded independently from the definition of an artefact for two reasons: because rationale which relates to many artefacts should be in one place and because this avoids having to search through an artefact's definition for rationale which may not be there, although it may be valuable to insert the design rationale into the artefact as a set of annotations (Potts and Bruns 1988).

3.5.1 Examples of integration of design rationale capture with geometry models

An early example of the association of design rationale and geometry models can be found in (Klein 1993, Klein 1997) this system is targeted at concurrent engineering design teams, to fulfil the need to capture dependencies between decisions captured by existing design tools such as CAD, requirements and manufacturing plans. Geometry models are transformed into a secondary format, either as screenshots or lightweight geometry model in the Virtual Reality Modelling Language (VRML). Design rationale is then associated with user selected areas of the geometry, which represent a meaningful feature or function as defined by the tool user (Klein 1997).

Design rationale in the file-based DRed format can be associated with geometry models by placing both files in one file folder (Bracewell, Ahmed et al. 2004, Du, Jing et al. 2012). The connection of DRed to the Microsoft office suite allows the indirect manipulation of geometry parameters from inside a rationale map, by means of a design table spreadsheet which is linked to the design rationale map (Bracewell, Gourtovaia et al. 2007).

McKay, Kundu et al. (2009) propose the creation of links between boundary representation geometry in a lightweight viewer, Solidworks e-drawings, DRed design rationale and a product process model, as a means of facilitating the transition to a service based engineering world. While a promising concept, only the link between the geometry model and the bill of materials is implemented, making an evaluation of the utility of the system difficult.

The annotation of lightweight geometry representations with a design rationale based annotation format is proposed by Hisarciklilar and Boujut (2009), with a view of reduce ambiguity in design communication. The main target of this method is to aid design reviews, and Hisarciklilar and Boujut (2009) propose the

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

inclusion of geometry screenshots with attached annotations in the design review minutes.

Zdrahal, Mullholland et al. (2007) propose a prototype system to integrate design rationales into a modelling and simulation workflow for mechanical engineering components, including a system to limit the diffusion of proprietary design knowledge outside company boundaries.

The creation of a dimension-rationale connection system to enable design rationale capture for geometry models is proposed by Chan (2007). The proposed system is "product-centric, dimension driven", with a predominant focus on parametric design and design intent, rather than a design rationale in the sense of a representation of a design space as defined by MacLean, Young et al. (1989b). The focus of Chan's (2007) method is the reuse of existing designs, in an adaptive design process, where the design task consists mainly of changing design parameters, in which case understanding the impact of dimension changes on the geometry model is essential. Chan (2007) objects to issue-based design rationales because rationale cannot be created until an Issue has been identified. The details of the method proposed by Chan (2007) to associate design rationale with geometry models remain unclear, due to the lack of implementation or even possible method of implementation of the ideas presented, while the near exclusive focus on the parametric aspects of the design process ignores topological and semantic design knowledge.

3.5.2 The relationship between geometry model design intent and design rationale

The term 'design intent' will be used here using the definition in the geometry modelling context as given in section 2.2.2. Iyer and Mills (2006) argue that design intent and design rationale are narrowly related. The structured variation of the model geometry created by the design intent could be considered as a model of the design space within one solution. While design intent models the variation of geometry within one topology, it does not generally retain other topologies considered, or explain why this particular solution was chosen.

Design intent is the ability of a structured geometry model that allows the definition of a range of possible geometries, but which does not explain or defend why a particular solution was chosen, what alternatives were

considered, and the anticipated impact of a particular design decision on the larger design process or product.

As such model design intent and design rationale are complimentary: design rationale explains the broader design decisions and alternatives considered, and design intent drives the variation of geometry in one solution.

3.5.3 The relationship between geometry annotations and linked design rationale

The existing work on annotation of geometry discussed in section 2.4 shows that frequently annotations are intended to explain the design decisions and aid communication of design. While digital annotations are not explicitly design rationales, annotations of geometry can contain semantic design information, such as alternatives considered and decision information. As such semantic annotations can contribute to the recording of the design rationale if they include information describing the decisions made during the design or the design space in which the design is created. There are a few differences between annotations and design rationales: annotations are individual comments on parts of the design, which are structured only by the connection to a part of the geometry or image, while design rationales tend to have a self-contained structure describing the design space for a part of the design process. Design rationale has a semi-formal structure (Bracewell and Wallace 2003) and therefore takes a place in-between the non-formal textual annotations and the formal annotation approaches based on KBE systems or ontologies.

The current research on annotations during design processes focuses more on annotations as a tool for asynchronous collaborative design and communication, where users annotate a geometry model consecutively, and use these annotations as a means of aiding storing and sharing information on the design, than the potential function of annotations as design rationale, see (Hisarciklilar and Boujut 2009, Lenne, Thouvenin et al. 2009) for an overview.

Surprisingly there seems to be little connection between the design rationale and annotation development, despite the similarities which exist, and many of the motivations for geometry annotation, discussed in section 2.4.1, could be supported by design rationale.

3.6 Data on design rationale capture

Design rationale has had a long-standing interest from the engineering design community. However, despite a large number of tools and capture methods which have been published and used over the years, design rationale capture in general and capture during geometry design in particular remains low

3.6.1 Quantitative data on design rationale capture

While there are numerous tool-oriented and case study papers on design rationale, there are few published quantitative studies on design rationale capture during the design process, which makes it difficult to make precise statements regarding quantity and methods used to capture design rationale during the various stages of the design process.

A study by Conklin and Begeman (1988) describes a project during which 2091 text based IBIS entries were captured during a software design process, however little information is provided on the evolution of the design rationale during the design process. Conklin and Burgess Yakemovic (1991) give the result of a software engineering project during which 8000 nodes of largely text based rationale were captured, as well as an analysis of the impact on the design process and design communication. Further results of the same design project (Burgess Yakemovic and Conklin 1990) including a graph of design rationale captured over time, showing that very little design rationale was captured once coding started.

Although there is little quantitative data available on the quantity of design rationale captured using the DRed tool, its use at Rolls-Royce is well established, as can be found from both the number and time span of the papers published, as well as its inclusion in the companies' standard toolset (Bracewell, Ahmed, and Wallace 2004; Aurisicchio and Bracewell 2013). Over 700 engineers have been trained to use DRed (Eng, Aurisicchio et al. 2012).

Kim, Bracewell et al. (2007) give some quantitative information on design rationale in two sets of DRed maps which are used as a dataset for the test of a design rationale retrieval tool, including some analysis of the most common design rationale structures indicating that many design rationale maps remain open; i.e. that no solution is selected.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

While design rationale capture is seen favourably where discussed, the extension of industrial use of design rationale, such as the use of Compendium for civil engineering design in Eng, Marfisi et al. (2011) remains novel, indicating that such industrial use is not standard practice.

3.6.2 Design rationale is captured especially poorly during detail design.

The lack of quantitative studies of design rationale capture during mechanical engineering design makes it difficult to evaluate how much design rationale is captured during each design stage. However the existing literature points to low design rationale capture during geometry design: Bracewell, Gourtovaia et al. (2007) note that the use of the DRed tool has been mostly limited to capture design rationale during the early design stages and to finding the root cause of, and solution to, problems encountered whilst a product is in service. The low design rationale capture during geometry design has also been noted by Eng, Bracewell, and Clarkson (2009).

Low design rationale capture during the implementation stage of the design process can also be found in software design: The results presented in a case study of a software engineering project show the amount of design rationale captured during coding was minimal (Burgess Yakemovic and Conklin 1990).

Huet, Culley et al. (2007) argue that design reviews are a key moment to capture design rationale, however retrospectively captured design rationales cannot realize all the benefits of capturing design rationale as listed by Lee (1997), such as aiding the decision process.

3.6.3 Why is Design rationale capture avoided?

Despite general agreement that capturing design rationale is good, the industrial use of design rationale seems to be quite low outside the published success stories of the use of DRed at Rolls-Royce such as (Auricchio and Bracewell 2013b). As early as 1992 it was noted that there are few published success stories on design rationale capture (Reeves and Shipman 1992) despite the many tools and published papers. Several possible causes could be identified from the slow uptake of design rationale in industry:

- The benefits of rationale capture do not outweigh the cost of capture.
- The current rationale capture methods are too costly.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Despite the long existence of design rationale industry awareness remains low.
- The short term costs of design rationale discourage capture, irrespective of the long term benefits.
- The current design rationale capture methods are not integrated in the design process.

While the existing studies of design rationale capture during mechanical engineering design processes indicate that concept design is well supported by the current generation of design rationale editors such as DRed, the low design rationale capture during geometry design indicates that either this stage is not well supported by the current tools. Three reasons for the low design rationale capture can be suggested:

- Lack of association: the tools which are used to successfully capture design rationale during early design stages don't provide any functionality to link entities in the geometry model to design rationale making it difficult to capture rationale for decisions concerning design geometry.
- Poor discipline: geometry design is an absorbing task, and documenting design decisions is easy to put off, especially when design work is done under pressure from deadlines.
- Lack of incentive: designers feel that rationales created during geometry design have insufficient relevance during the further design process, and that they are unlikely to benefit from capturing design rationale. Brunsmann and Wilkes (2009) believe knowledge is not captured because engineers think the effort is not justified.

3.7 Summary

This chapter introduces and defines design rationale, as a means of describing a design space. The different design rationale representations and tools which have been published over the years are discussed, with a particular focus on the integration of design rationale capture facilities into existing design tools. Further attention is given to the available data on the capture and use of design rationale during engineering design processes, and the evidence for the low capture and use of design rationale during geometry design is discussed.

4. Design rationale capture during an aeronautical engineering design process: a case study

The case study presented here performs a quantitative analysis of design rationale captured during the design of a light UAV system, and correlates the design rationale capture with the different stages in the design process.

Despite the large amount of academic work which has been published on design rationale, see chapter 3 for an overview, there are few published studies on quantitative design rationale capture during mechanical engineering design processes, as discussed in section 3.6.1.

The lack of data on when design rationale capture occurs during the design process makes it difficult to propose tools to support capture, as it is difficult to identify gaps in the support from the current toolset.

4.1 Study goals

This study is a part of the Descriptive 1 study in the DRM discussed in section 1.5.2. The lack of quantitative design rationale capture studies identified in section 3.6.1 is the main motivation for this study, as without data on design rationale capture it is difficult to propose tools to support designers. The following objectives can be identified:

- Establish quantitatively the level of design rationale capture during the different stages of a geometry design process.
- Provide a baseline of design rationale capture before any novel tools have been introduced.
- Seek indications on which, if any, support gaps exist.

4.2 Choice of a design rationale capture metric

To measure the capture of rationale it is necessary to define a metric to measure the quantity of rationale. The following requirements for a design rationale metric can be identified:

- Be based on data available in the design rationale database.
- A good proxy for the quantity of information in the design rationale

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Several possible metrics of design rationale capture can be identified:

- Number of rationale maps created
- Number of rationale nodes created
- Number of rationale edges created
- Quantity of text created

The capture of design rationale nodes was selected as the most promising metric for the following reasons:

- Maps are not a good measure of the quantity of information in the design rationale as they can be empty or contain hundreds of nodes.
- Edges frequently have no label, in which case their only information content is to create a connection between two nodes.
- Measuring the quantity of text in the design rationale conflicts with the objective of brevity for design rationale node labels.

Previous studies presenting quantitative results of design rationale capture such as (Burgess Yakemovic and Conklin 1990) have also selected the design rationale node as the principal metric.

The rationale editor used, Compendium, allows node transclusion; the occurrence of a node in more than one map. The rationale metrics used in this case study are based on node occurrence, so transcluded nodes will be counted multiple times, while nodes which were deleted before the end of the study are not counted. Although the number of design rationale nodes captured is a useful quantitative metric, it provides no indication of the quality of design rationale captured.

4.3 Study setup

4.3.1 Data collection

This case study was set up after the design process was complete, using the design rationale database, containing all the design rationale created during the design process, as the principal data source. The design rationale database was analysed using the software developed by (Gorissen 2011) to produce design rationale capture statistics. Advantages of this approach to creating design rationale capture statistics are:

- Non-intrusive creation of design rationale statistics

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- Statistics can be generated for past projects if the database has been conserved.

4.3.2 Design project

The design project during which design rationale capture data was gathered was the design process of a light (sub 20 kg) UAV, primarily designed for a maritime search task, as a part of a search and rescue mission including lifeboats and helicopters. This aircraft was the first iteration of three UAVs which will be designed, optimized, built and flown as a part of the DECODE research project. The design rationale capture study concerns the design of the Decode 1 UAV, shown in Figure 18.

This design process was part of the DECODE project described in section 1.2. The design project had two principal goals:

- The development of a prototype UAV
- The development of an integrated UAV design search and optimization system.

The integrated design and optimization workflow was being developed in parallel with the first aircraft design discussed in this case study. A prototype parametric optimization tool was used during the design process of the UAV discussed in this study, which didn't include the full CFD, mission simulation and value optimization functionality which were the eventual project goals.



Figure 18 The DECODE 1 UAV in flight

The UAV design process described includes the concept and embodiment design, construction, flight testing and recording the findings for use in the next design iteration. It was a new design, with no reuse of any in-house designed parts from previous design iterations.

4.3.3 Designers

The 10 person team working on various tasks in this project all had different roles in the design process. Four distinct working groups can be distinguished: management and advisory team (3 persons), aircraft design team (4 persons), analysis software team (2 persons), and mission simulation (1 person). All team members had access to all the rationale and could edit all of it, including maps and nodes created by others.

The aircraft design team and the mission simulation team consisted of novice designers, while the management team and the analysis software team consisted of experienced designers and developers.

4.3.4 Design tools

The following design tools saw substantial use during the design process:

- The Solidworks 2009 CAD package used for geometry design.
- A Subversion (SVN) document repository system was used for version management.
- Aircraft parametric modelling and optimization were done using a combination of spreadsheets and an aircraft concept design tool, PaceLab⁸.
- The design rationale was captured using Compendium, using a shared database for all team members.

During the design process discussed in this case study, design rationale was captured as a methodical by-product of the design as defined by Lee (1997).

4.4 Design Rationale Capture statistics

During the design project a total of 1716 nodes occurrences were created between the 6th of March 2010 and the 21st of January 2011, for 1645 unique nodes. Figure 19 shows the design rationale captured by node type. The low frequency of Decision nodes in the final design rationale shown in Figure 19 can be explained by the lack of a formalized decision making system which would transform questions where sufficient alternatives and evidence had been examined into decisions.

⁸ See: <http://www.pace.de/>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

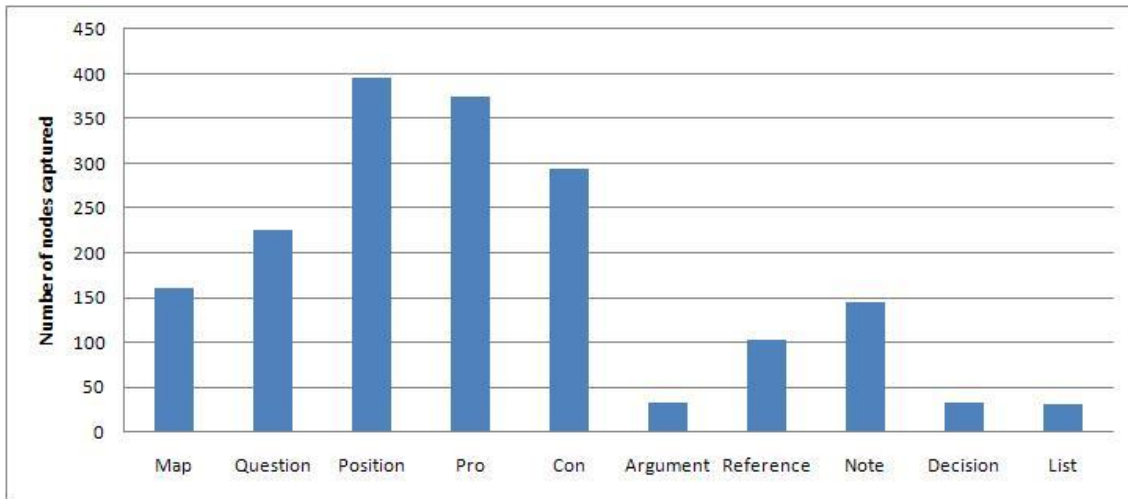
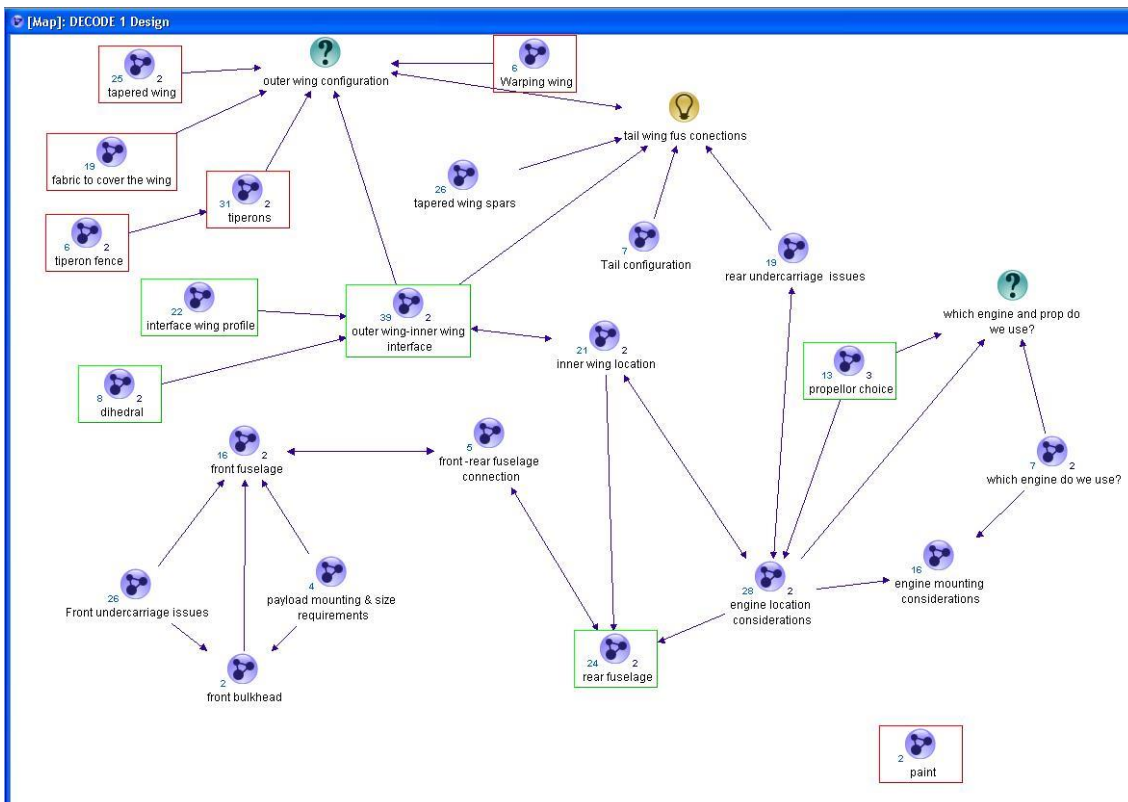


Figure 19 The distribution of design rationale node occurrences captured during the project by node type.

Figure 20 shows the top level structure of the aircraft design rationale, where adjacent decision areas are mapped out. It should be noted that the rationale is organized into maps which contain topological areas or interfaces between areas of the aircraft, as well as some specific high level design issues. Arrows are used to map the dependencies between adjacent rationale maps. This type of dependency mapping is similar to the Function Analysis Diagram (FAD) proposed by Bracewell, Gourtovaia et al. (2009).



DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Figure 20 The top level view of design rationale maps of the aircraft design.

As the design progressed decisions were made for issues outlined in the design rationale, but often they were implemented in the geometry model directly without the relevant rationale being updated.

The contribution of nodes per person ranges from 2 to 839 nodes. In general those with a management oriented task contributed fewer nodes than those with engineering or software design tasks. Design rationale capture per team during the project is shown in Figure 21.

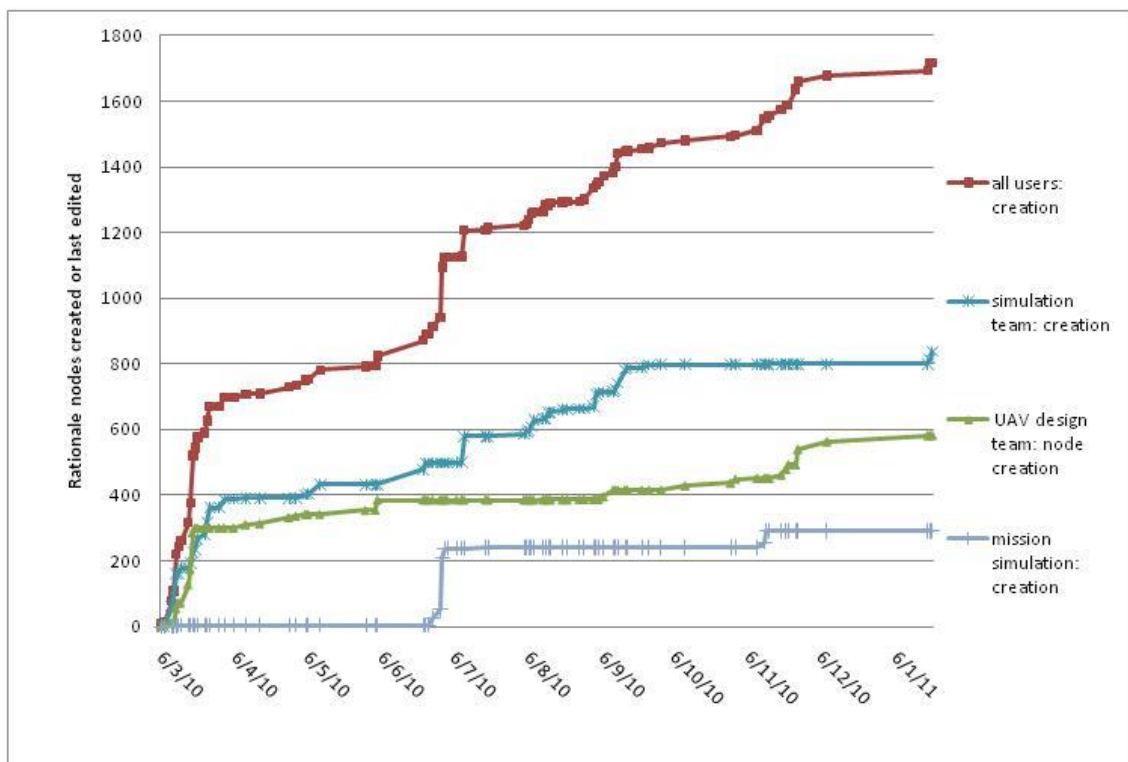


Figure 21 Design rationale nodes created during the project by all design teams.

It is clear that there is a substantial variation of rationale capture with time, especially when the rationale capture per team is considered. The rationale capture by the management team is not shown as the total number of rationale nodes created, less than 50, was very low. The mission simulation team had a later start, June 2010, explaining the slow start of rationale capture. A common pattern in all three teams is that the rationale capture has an intense start, often followed by a period where no or very little rationale is captured and then renewed period of capture at the end of the design process.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Figure 22 shows the design rationale capture by members of the UAV design team. The rationale capture by the UAV design team maps quite clearly these points in the rationale capture process can be seen to coincide quite clearly with key points in the design.

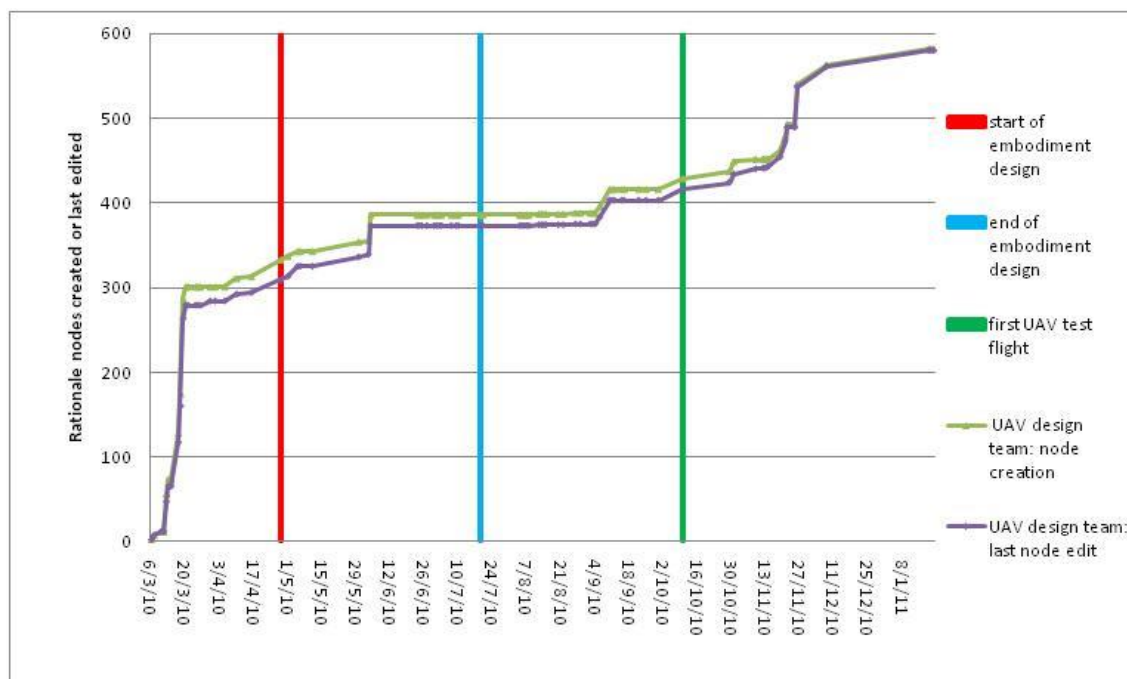


Figure 22 a quantitative view of design rationale captured by the design team during the design process.

It is clear that very little rationale was captured during the embodiment and design phase, even though a substantial amount was captured during the concept design stage. The 'node creation' curve shows the number of nodes created to date; the 'last node edit' curve shows the number of nodes unedited after the date. The vertical proximity of the 'node creation' and 'last node edit' curves show that very little of the rationale nodes were edited after their creation. Most cases the editing of rationale nodes can be explained by the changing of a Position into a Decision to reflect the final design. The design rationale map in Figure 23 is an example of a map in which positions have been changed to decisions once sufficient information was available.

The design rationale map in Figure 23 shows one of the key uses of design rationale during the design project: the collaborative creation of design rationale to define interfaces between parts which were designed by different engineers.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

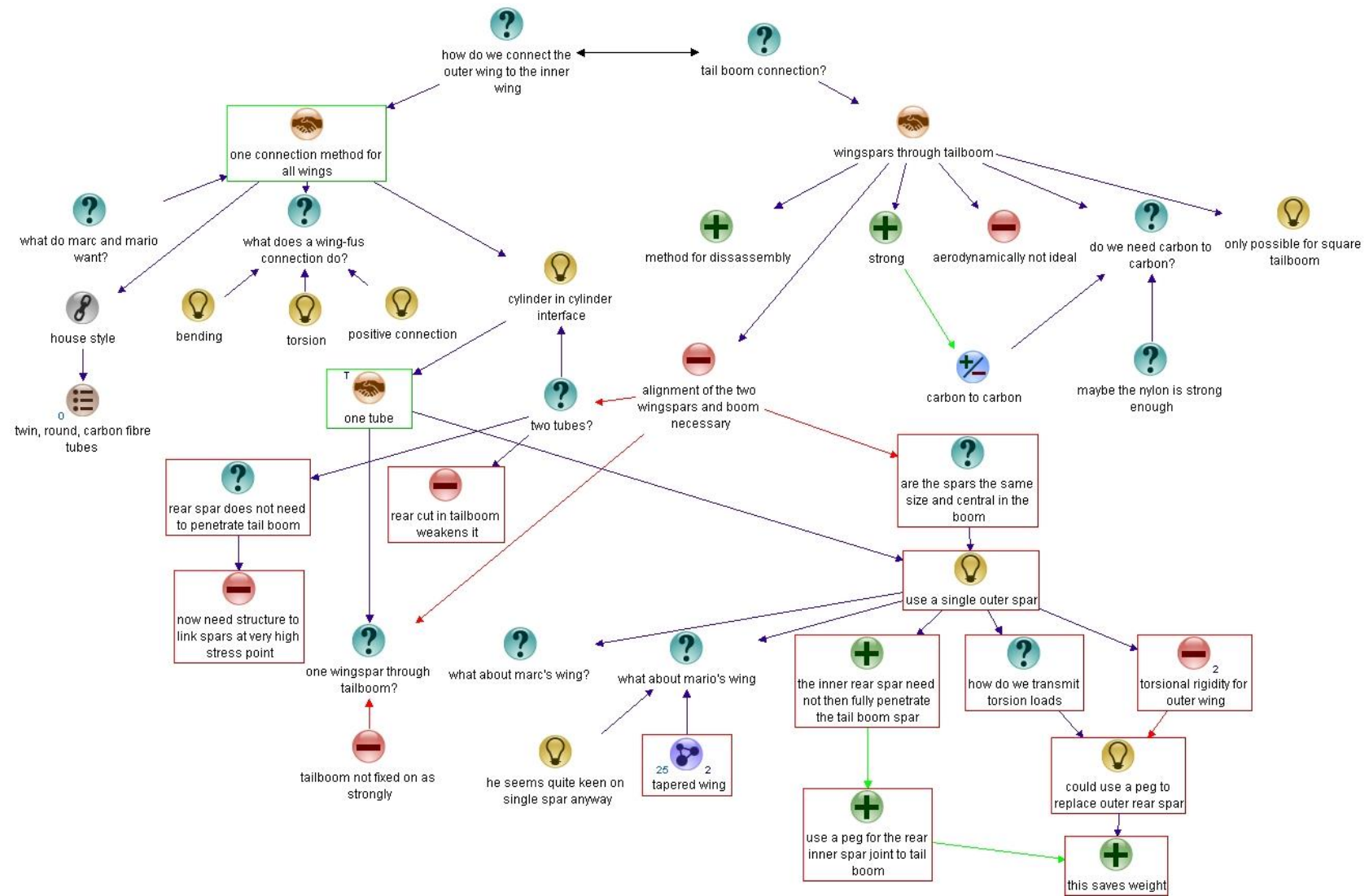


Figure 23 a design rationale map produced during the design process detailing interface design

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

The embodiment design started around the 28th of April 2010, which is defined here as the point where the first new design was created which was used in the aircraft. Embodiment design is considered to have finished the 19th of July, the last preproduction change to the design. The first test flight took place on the 9th of October 2010.

Weekly progress and design review meetings were used to monitor the design process and solve conflicts between the designers. During the UAV pre-flight and flight testing process there was once again an increase in design rationale capture, which can be explained by the design team revisiting the rationale and added more precise data or information gathered during the tests. However this took place in the form of adding nodes, rather than changing existing ones; the graph shows very few nodes were changed after their creation.

4.5 Discussion

The design team used Compendium extensively to capture design rationale during the concept design, but their rationale capture dropped once geometry design started. A similar pattern can be identified in the mission simulation and optimization teams. These observations on the pattern of design rationale capture are consistent with the graph of design rationale capture over time in during a software development process (Burgess Yakemovic and Conklin 1990) and the observations by Bracewell, Gourtovaia et al. (2007) and Eng, Bracewell et al. (2009) of design rationale capture during mechanical engineering design processes.

All the existing evidence points towards a reduction of design rationale capture during the implementation stage of the design processes, which is not exclusive to mechanical engineering design processes. Although the pattern of low design rationale capture during the implementation phase of the design is repeated across disciplines, and not unique to geometry design, the design rationale created during the implementation phases of software engineering may be stored as comments in the software code, rather than as a design rationale map. Several possible reasons for the low design rationale capture during geometry design can be identified in Table 6.

Table 6 possible reasons for low design rationale capture during geometry design.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Factor:	Type of factor:
There is no structured method for linking rationale to geometry, making the capture of rationale referring to a specific part/area or feature of the geometry difficult.	Technical
The implementation phase of the design process is frequently performed under time pressure. The designer is focused on finishing the design and for this reason has little time for secondary activities such as design rationale.	Environmental
The embodiment design process using a geometry modelling tool is an immersive experience. Designers are so focused on creating and editing geometry that they are disinclined to interrupt this immersion to use other tools.	Environmental
Design rationale for the more detailed design decisions encountered during the implementation stage feels less	Motivational
The benefits of improved design documentation though design rationale capture do not accrue to the geometry model editor, and are therefore easy to put off.	Motivational

4.6 Conclusions

The pattern of design rationale capture in this study shows substantial design rationale capture during concept design and testing, and low design rationale capture during the embodiment and detail design phase. This pattern is repeated in the data of the two non-geometry design teams. The low design rationale capture during the geometry design stage confirms the earlier observations in Bracewell, Gourtovaia et al. (2007) and Eng, Bracewell, and Clarkson (2009). Several possible causes are identified for the observed drop in design rationale capture due to motivational, environmental and technical

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

reasons. The main technical reason identified is that it is difficult to create meaningful design rationale during the detail design stage, as it is difficult to connect parameters and entities in the geometry model with the relevant rationale.

4.7 Summary

This study describes the design rationale capture during the design process of a small UAV. The evolution of design rationale by the UAV design team during the design project shows that substantial design rationale capture occurred during the initial phases of the design process, but that design rationale capture during geometry design was low. Several possible reasons for the low capture during geometry design can be identified, however, the absence of a method to link design rationale to geometry entities is identified as a key technical factor holding back capture.

5. The approach: Linking design rationale to entities in geometry models

This chapter will propose a methodology for linking geometry model entities to design rationale. The proposed method builds on the previous work on the annotation of geometry models in section 2.4, and on the integration of design rationale capture functionality in design tools in section 3.5.

5.1 The need for linking design rationale to geometry model entities

The research presented in the previous chapters of this thesis has established that:

- Knowledge management is an essential part of modern engineering design.
- Geometry models are used as the main method of communicating and storing mechanical engineering designs.
- Geometry models do not contain sufficient information regarding design rationale or design intent to effectively communicate the design.
- Numerous existing approaches exist for annotation of geometry models with the view of providing additional information on design decisions.
- Few existing approaches exist to integrate design rationale capture into geometry modelling.
- Design rationale capture can represent decision processes efficiently.
- Numerous calls exist in literature for the integration of design rationale into the design process and geometry modelling tools, as discussed in section 3.5.
- The study of design rationale capture during geometry design in chapter 4 confirms the observation of low design rationale capture during geometry design.
- The one of the likely technical reasons identified for low design rationale capture during geometry design is the inability to create a precise connection between a design rationale map and an area of the geometry.

In the light of all the factors outlined above, the integration of design rationale capture functionality into geometry modelling tools seems to be a promising area of investigation. The creation of such a tool is therefore required to establish whether the creation of such functionality would indeed improve design rationale capture and use during the geometry design stage. The purpose

of this chapter is to establish what functionality such a tool should provide to be a successful remedy to the issues identified previously.

5.2 Requirements for a tools to improve design rationale capture during geometry design

The list of requirements presented by Burge (2005) for facilitating design rationale capture during software development can be translated and adapted to a mechanical engineering geometry modelling environment:

1. Design rationale capture tools should integrate in the geometry modelling environment.
2. There should be an explicit way to associate design rationale to geometry model entities.
3. The existing design rationale should integrate seamlessly with the design rationale created during geometry design.
4. Geometry model users should be informed that design rationale is available, without having to search for it.
5. Design rationale should be integrated in geometry checking functions.
6. Linked design rationale should be able to support both the design decisions and the geometry modelling decisions.

The following sections will define and expand these adapted requirements.

5.2.1 Integration of design rationale linking functionality into the geometry modelling tool

There should be a low effort barrier to the linking of design rationales; As the CAD suite is the main tool used during geometry design, design rationale linking should be initiated from the geometry modelling environment. This can be achieved by integrating the functionality to create links to design rationales into the geometry modelling environment itself.

5.2.2 The need for a low granularity link between rationale and geometry

Highly detailed design decisions require an intimate link between design rationale and geometry to allow the capture of meaningful design rationales. Preferably such a link should work in both directions, from rationale to geometry and from geometry to rationale.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

To provide an efficient reference to a geometry entity, the annotation granularity should be sufficiently precise to make it clear which geometry decision is being referred to. See the discussion of annotation granularity in section 2.4 for a review of the granularity of existing annotation approaches.

5.2.3 Inclusion of concept design rationale

The main purpose of capturing rationale during detail design is to document new design issues which have become apparent during geometry design, as well as to complete and update the existing design rationale

During concept design, when the acceptance, capture and usage of design rationale capture has been higher in the past, the design rationale and options for decisions are laid out, including issues which can be anticipated but not yet resolved. This can be seen in the many open design rationale structures in (Kim, Bracewell et al. 2007). During geometry design most of these issues will come to a conclusion, and therefore it is a good occasion to update the design rationale to complete the record of decision processes.

The inclusion of concept design rationale prevents a recreation of similar design rationales which could lead to conflicting documentation of the design. Furthermore the referencing of the concept design rationale encourages updating of the rationale to reflect the final design, making additions and changes where required. This will not only improve the documentation of the detail design, but improve the completeness of the concept design rationale.

5.2.4 Support the documentation of geometry model design intent

As discussed in section 2.3.2 the design intent present in geometry models can be difficult to understand for those not familiar to the particular geometry model or even to the original designer after a number of years.

In addition to documenting the decisions of the design itself, design rationale capture during geometry editing should also allow the geometry model design intent to be documented.

5.2.5 Systematic approach and audit

Design reviewers should be given an opportunity to get a clear insight into the design (Parnas and Weiss 1985). However it is difficult to get a clear insight into the structure of a geometry model, as discussed in section 2.3.2.

CAD programs contain several visual feedback functions using colours or patterns to display curvature or thickness. The extension of these visual feedback functions to indicate the presence of design rationales, external references, under defined sketches and input dimensions can provide an additional method of feedback on the design status

Geometry models are complex entities whose underlying structure is often obscure. This makes design reviews difficult, as these require an intimate understanding of the product by outsiders to the design process (Parnas and Weiss 1985). Auditing the design to reveal 'hotspots', such as input dimensions, lack of rationale and under-defined sketches, and contrasting them with the rationale for these decisions should give design reviewers an additional tool to find weak areas in the design thinking.

It can be argued that design rationale capture is sufficiently important that providing a strong incentive for the designers to capture design rationale is warranted. Allowing design reviewers to monitor how much rationale has been captured for each region of the geometry could provide a metric to motivate designers to capture design rationale in the first place.

6. Linking design rationale to geometry entities

This chapter describes a practical implementation of a software prototype in partial fulfilment of the requirements outlined in chapter 5, particularly the following:

- The creation of a permanent reference to the design rationale, which is a part of the structure of the geometry model and will remain attached to a geometry entity for the life of this entity.
- A seamless integration of design rationale linking functionality in the geometry modelling tool.
- The ability to navigate existing design rationale and link to design rationale created previously.

6.1 Linking design rationales to feature-based geometry models.

While most of the existing annotation approaches have chosen to annotate lightweight geometry models, the toolset presented here uses a feature-based geometry modeller. The following reasons motivate linking rationales to the full feature-based geometry models:

- Geometry models contain a substantial amount of rich design information in the form of the parametric structure of the model. This can consist of design decisions implicit in the model, as well as explicit decisions in the form of equations, design tables and inter-part links. Abandoning the structured, feature-based parametric geometry model in favour of a non-structured, non feature-based light-weight approach results in the loss of explicit and implicit design information which was built into the model originally. Although efforts are on-going to retain design intent when exchanging models between CAD systems, such as (Kim, Pratt et al. 2008), current tools do not easily support such exchange.
- A substantial part of the detail design will be done while editing the geometry model, as and when the need for decisions becomes clear, without first having been considered in other design tools external to the geometry modeller. As decisions are being made during geometry modelling this would be the logical place to capture the design rationale.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Design rationale should be captured as and when design decisions are made (Brunsmann and Wilkes 2009).

- Geometry models are increasingly used as the authoritative definition of the shape and properties of the design, and are increasingly integrated into the production processes by means of CAM, especially for rapid manufacturing. Having design information available to the downstream processes can aid design changes for manufacturability, such as parameter changes.
- If any change is made to the design at a later date, the geometry model is likely be reused and edited to accommodate the changes, and the design rationale for the original decisions would aid this form of design reuse, as additional information on the geometry model structure can speed up geometry model editing (Alducin-Quintero, Contero et al. 2011).

While the risk of poor future availability or compatibility of geometry modellers certainly has to be taken into account and managed, especially for products which have a very long production and service life such as aeronautical engineering products (Patel, Ball et al. 2009), the use of feature-based geometry modellers for geometry editing remains the most logical solution as both a design tool and a form of design documentation.

6.2 Database or file-based design rationales

The principal advantage of the database-stored rationale is that multiple users and tools can interact with the rationale collaboratively. While the toolset proposed here seeks to cover the entire design process and design life, it is not exclusive and other approaches to link or integrate design rationale can be used in conjunction. Although Bracewell, Wallace et al. (2009) argue that file-stored design rationales are preferable to database stored rationales due to the difficulty of managing databases and the unfamiliarity of users with database management, the ability to have distributed, multi tool access to the design rationale, and to be able to build seamlessly upon earlier rationale outweigh the possible user inconvenience.

Furthermore graphical interfaces to database management systems have made database maintenance easier, while users have become more used to using programs where the data is stored on an external database, for example the Mendeley⁹ reference manager.

6.3 Practical implementation of the RatLink tool

The prototype tool, called RatLink, which stands for Rationale + Link, takes the form of an add-in for the Solidworks CAD package, and communicates directly with a MySQL database containing the design rationale. Figure 24 shows the basic structure of the interactions between a design rationale database and the various tools which are used to edit or modify the design rationale. All the design rationale tools share access to a common database allowing the distributed and simultaneous editing of design rationale.

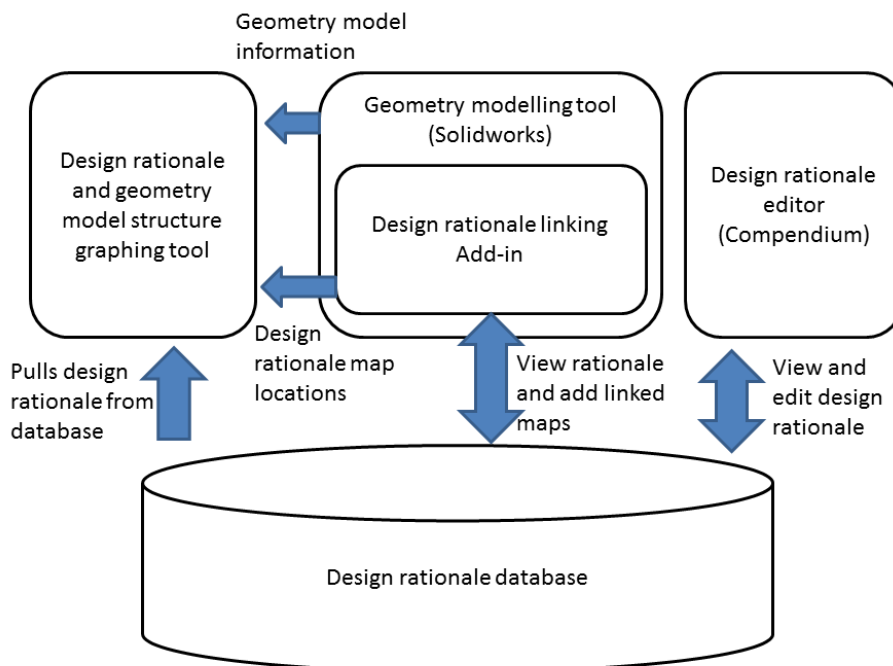


Figure 24 High level overview of the interactions between the design rationale linking tools and the design rationale database.

The implementation of the tool as an Add-in has the following advantages:

⁹ See <http://www.mendeley.com/>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- The tool can access to the properties of the geometry model being edited.
- It enables seamless integration in the geometry modelling tool using the controls familiar to the users, facilitating adaption of the tool.

The RatLink tool interacts with a MySQL database storing the design rationale directly. Reverse engineering the database format used by Compendium has allowed the design rationale linking tool to create maps and nodes which can then further be edited in the design rationale editor.

6.3.1 Choice of CAD tool

A feature-based geometry modelling tool, Solidworks was selected as it is used for design education to undergraduate students, allowing them to serve as a test population for the tools created. Even though the tool implantation shown here is for a specific CAD suite, it should be possible to create similar tools for other feature-based modelling platforms, provided they give sufficient API access and allow the creation of attributes linked to model entities to store the design rationale links.

6.3.2 Link Implementation

To create a permanent, bidirectional link it is necessary to leave a marker, both in the design rationale and in the geometry model indicating the presence of the link, and storing basic metadata to facilitate retrieval of the linked information. Figure 25 shows the implementation of the geometry to design rationale link.

The idea to use the attributes supported by the geometry kernel for the mark-up of geometry models is discussed by (Davies 2008 p. 140). Solidworks is based on the Parasolid geometry kernel (Maher 2013). The Parasolid geometry kernel provides for the definition of attributes attached to geometry entities (Siemens Product Lifecycle Management Software Inc 2008).

The attribute interface provided by the geometry modeller is principally used by 3rd party add-ins for FEA and CFD as a means of defining boundary conditions and inputs on the geometry, e.g. (Davies 2008 p. 144).

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

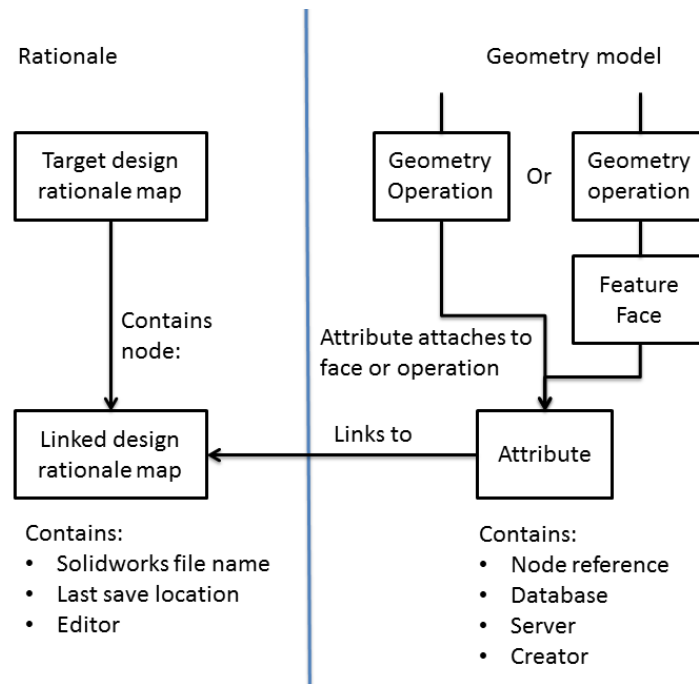


Figure 25 Linking rationale to a geometry model by attaching attributes to the operations or faces in the operations tree. The metadata which is stored on both ends of the link provides additional information on the entities linked.

The durable attachment to an operation or face which persists as long as the face or operation is retained in the model, matches the requirement for a permanent link between geometry and rationale. The presence of an attribute can be displayed in the operation tree, giving an indication of the presence of the design rationale even when the design rationale linking software is not used.

Figure 25 shows the practical implementation of a single link and how this link exists within both the geometry model and the design rationale. The attribute will remain associated with the feature or face for the duration of the presence in the model. If the operation becomes part of a 'library feature', it is possible to retain the attributes attached to the operations and therefore the design rationale link even when the operations are reused in a different model.

6.3.2.1 Granularity of design rationale links

The annotation of geometry representations or geometry models requires the choice of an annotation granularity. See the discussion of annotation granularity in section 2.4.2 for a review of the existing approaches.

The implementation of the model attributes in Solidworks allows the creation of attributes for other objects in the model such as bodies, geometry operations, planes and sketches, as well as geometry entities such as faces, vertices and edges. However, the design rationale linking tool allows selection to only two of these entities: either a face of the geometry selected from the graphical model view or an operation and its associated features from the operation-tree.

This is motivated by two considerations:

- The desire to integrate design rationale links into the structure of the geometry model.
- The need for reference to a geometry entity for visual feedback functions.

As the operation is the main entity manipulated in CAD geometry design, this is the prime attachment points for any design rationale links created. If the selection of all of the features created by an operation would be a link of too high a granularity to be practical, a face of the geometry can be selected as the reference for a design rationale link.

Furthermore the ability to create design rationale links to faces allows the linking of design rationale to imported boundary representation geometries, which lack the operation structure present in CSG models.

The visual feedback functionality based geometry properties and design rationale presence discussed in chapter 8, requires that design rationale links refer to a part of the geometry, rather than to sketches, axis or planes.

The current implementation of the creation of attributes does not allow direct linking of design rationale to dimensions or equations so the design rationale for decisions reflecting these entities has to be stored with the owner feature of the dimension.

6.3.2.2 Limitations of this linking technology

The main limitation of linking design rationales by means of model attributes is that the geometry models have to remain within their native CAD system, as exporting the model to lightweight formats such as STEP or IGES using the native export functionality present in the CAD system will remove the attributes and therefore the linkage to the design rationale.

However approaches have been created to mitigate this issue, such as the geometry annotation system using persistent geometry references (Ding, Davies et al. 2009) or using a 3d reference point as a location anchor (Ding, Ball et al. 2011).

6.3.3 User interface design

The user interface of the tool is fully integrated in the geometry modelling tool and is presented to the user using the same style as the standard selection boxes and buttons used in Solidworks. A design rationale management toolbar allows the user access to all the functions relating to the creation, use and removal of the design rationale links, as well as a user manual.

6.3.3.1 Design rationale toolbar

Figure 26 shows the design rationale toolbar, which allows the creation and management of design rationale links.



Figure 26 The design rationale toolbar providing access to design rationale linking functionality

The toolbar integrates seamlessly into the geometry modelling tool, and presents users with easily accessible functions. From left to right these are:

- **Add Rationale:** accesses the design rationale linking menu discussed in section 6.3.3.2.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- **Select Database:** accesses a menu which allows the design rationale database to be selected.
- **Set user ID:** accesses a menu to log on to the Compendium project relevant to this geometry model.
- **Remove all links:** removes all design rationale links from the model.
- **Show all links:** makes all the design rationale links in this model visible in the feature tree.
- **Hide all links:** hides all design rationale links in the feature tree.
- **Show rationale:** accesses design rationale and geometry audit functionality described in chapter 8.
- **Usage instructions:** accesses a user manual.
- **About:** provides general information on this tool.

6.3.3.2 Design rationale linking menu

When the 'Add Rationale' button is pressed and a connection to the design rationale database is available, the design rationale linking menu is displayed. An example of the user interface is provided in Figure 27. To create a design rationale link three elements are needed:

- **A geometry entity to link:** selected either the graphical model view or from the operation tree.
- **A name for the design rationale link:** entered by the user.
- **A target design rationale map:** selected through the design rationale navigator.

The tree based design rationale navigator is discussed in the next section.

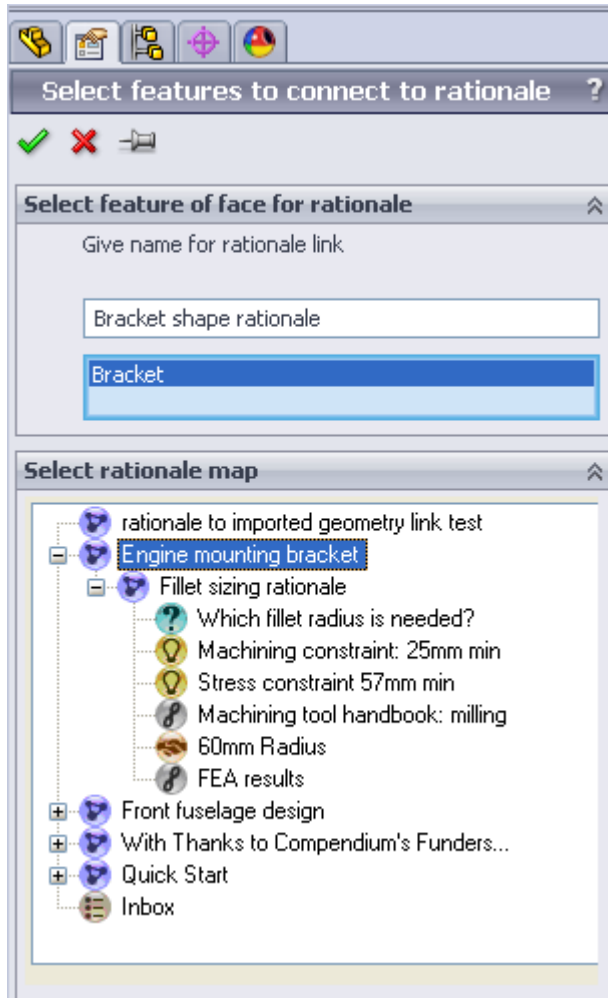


Figure 27 Design rationale linking menu

6.3.3.3 Tree view of database for rationale navigation

The proposed system uses a tree view of the design rationale for the navigation the pre-existing rationale, as well as selection of the target design rationale map for the link. This is not a new concept: early studies of design rationale capture, for example (Burgess Yakemovic and Conklin 1990, Conklin and Burgess Yakemovic 1991), which took place before the introduction of gIBIS (Conklin and Begeman 1987), used indented text to create a tree based display of design rationale.

The tree-view was chosen for compactness of the user interface, which allows navigation of the existing design rationale without having to scroll through large design rationale maps on a small window. Figure 28 shows the implementation of the design rationale tree view interface, showing the same design rationale as in Figure 30.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

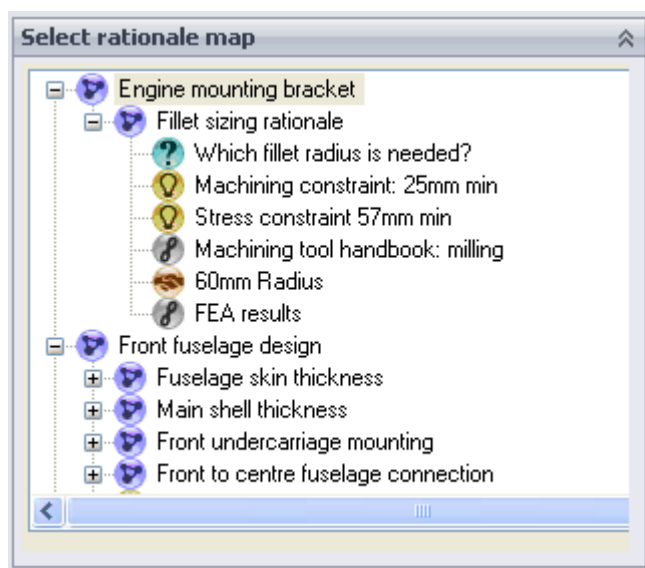


Figure 28 A detail of the design rationale linking interface showing a tree-view of a set of rationale maps and nodes.

A graphical interface, such as used by present day design rationale editors would have required considerably more development effort and would have created redundant functionality with the existing graphical design rationale editors. Furthermore a graphical editor would require either considerable space in the user interface layout, which would hinder the seamless integration with the geometry modeller or frequent scrolling on the part of the user to find the relevant rationale nodes.

For these reasons only a list based view of the existing design rationale is provided and only the link between the geometry model and the pre-existing rationale is created in the add-in, with all further editing of the design rationale done in Compendium. To facilitate the identification of the rationale the icons used are identical to those used in Compendium.

The purpose of the design rationale database view is to navigate the design rationale to locate the target map for the rationale link. In practice this method of viewing the pre-existing design rationale is sufficient to find the relevant design rationale maps. To facilitate adding rationale links for multiple items in the model, the last used design rationale map is saved, and when the rationale navigator is reopened the tree-view shows the tree to the last used map.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

The transclusion functionality in Compendium allows circular references; i.e. it is possible for a map to contain a reference to itself. To accommodate circular references the rationale tree view doesn't load the whole database instantly, but a layer at a time based on user selections.

6.4 RatLink tool demonstration

This section will demonstrate the functionality of the system by providing a walkthrough of the steps needed to create a link to the design rationale from within the geometry modelling environment.

6.4.1 Setup

The system is packaged in a windows installer, which will install the add-in and add it to the registry. Once the add-in is installed the geometry modeller can be started and the user can connect to the database and project using the same password and user identification as used for the Compendium rationale editor. Figure 29 shows the 'Select Database' and 'Set User ID' icons which provide access to the logon windows. The add-in will check for connectivity to the design rationale database, and confirm to the user that connection is possible.

6.4.2 Linking design rationale

Whenever a new decision is made during geometry design, or a previously made decision is implemented, the design rationale of this decision can be captured and linked to the geometry model. As the geometry modelling tool is the primary tool in this design stage the design rationale linking is initiated there.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

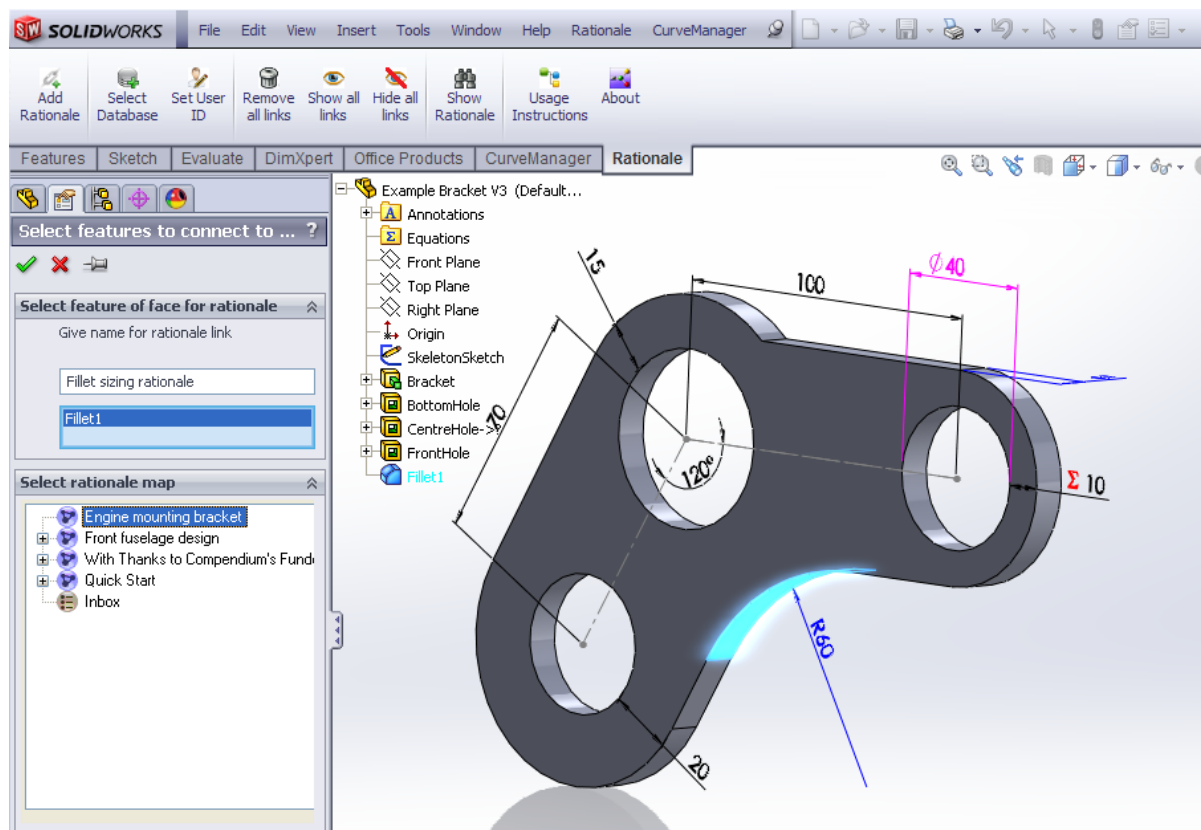


Figure 29 Design rationale linking user interface, allowing selection either graphically or from the operation-tree.

Once the user has connected to the database, rationale can be added through the menu in Figure 29, this is done by selecting a feature, face or sketch to which the rationale is to be connected, either from the feature tree or from the geometry of the model. A rationale map, to which the link is to be added, is also selected using a tree-based view of the rationale maps. To complete the link the user selects a name for the rationale link, by which the rationale link will be identified in the design rationale map and modelling tree.

6.4.3 Viewing the rationale link from the rationale editor

The rationale link takes the form of a Compendium map, allowing the user to associate the link with the earlier rationale, while creating new rationale inside the linked map. See Figure 30 for the Compendium view of the linked map, 'Fillet sizing rationale' and the rationale captured for this decision.

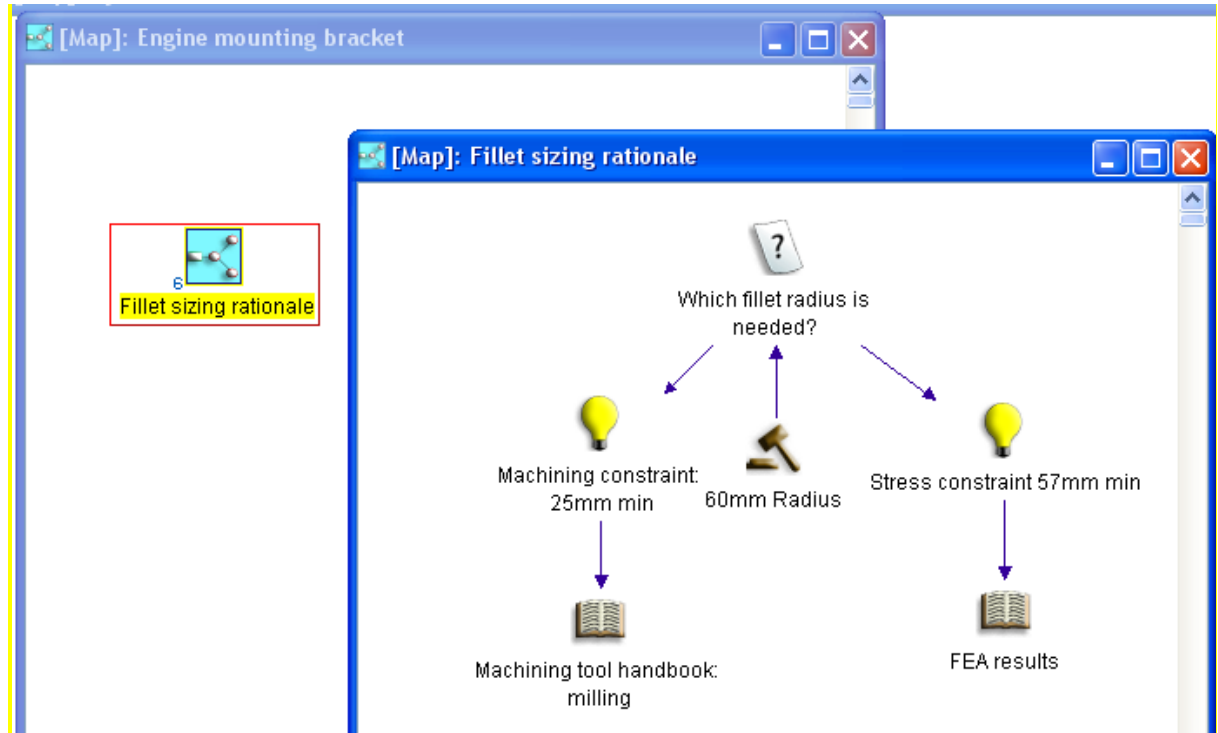


Figure 30 A linked rationale map, 'Fillet sizing rationale' has been created in Compendium and rationale for the decisions can be captured using the normal Compendium interface.

6.5 Summary

This chapter proposes an implementation of a design rationale linking tool. Links between geometry entities are created by means of attributes attached to geometry modelling operations or faces of the geometry. It further discusses the implementation of such a link, the choice of CAD suite for which to implement to tool and the user interfaces available to the user. The chapter concludes with a demonstration of the completed prototype system.

7. Creating graphs of geometry model structure and associated design rationale

This chapter presents a novel approach to create combined graphs of geometry model design intent and associated design rationale. Geometry model design intent can be hard to understand from inspection of the geometry, see section 2.3.2 for a review of the relevant research.

The main purpose of this tool, called the GRapher (Geometry + Rationale grapher), is to aid the understanding of both the geometry model structure implementing the design intent, as well as the design rationale associated with the geometry by means of the linking tool described in chapter 6.

Specifically two goals can be identified:

- To create a complete view of all design intent content of a geometry model, including mates, equations, dimensions, sketches and external references.
- To retrieve all design rationale which was associated with the geometry model using the design rationale linking tool presented in chapter 6.

The resulting graphs can be used as a form of design documentation, as well as an aid to geometry model editing.

The GRapher tool approaches the problems in understanding geometry model design intent from the opposite direction of that taken by Salehi and McMahon (2011, Salehi-Douzloo 2012), which is focussed on establishing the model relationships to be implemented in the geometry model before the model is created, while the GRapher tool displays the relationships present in an existing model.

7.1 Related works: creating graphs of geometry model structures

Geometry models can contain numerous relationships which can each be viewed and managed in separate tools in the CAD program, but for which it is difficult to get a general overview. Figure 31 and Figure 32 show two forms of geometry model relationship viewers.



Figure 31 Parent-Child operation dependency display in Solidworks

Figure 31 displays the 'Parent/Child Relationships' tool which displays a small selection of a feature dependency graph which can be navigated by clicking on an operation. While certainly a useful aid to understanding design intent, this display does not show further design dependencies created by mates, dimensions or equations. Figure 32 shows the external reference manager, displaying links between the model currently being edited and external models.

See (Shah 1991) for an overview of early work on various methods for the decomposition of geometry part models into graphs, while Shah and Rogers (1993) propose assembly structure graphs including mate relations on a part level.

The search through libraries of existing models for models with a certain topology or arrangement of features has been a major driver for research into creating graphs of the structure of geometry models for example see: (Ma, Huang et al. 2009, Bai, Gao et al. 2010, Li, Zhang et al. 2010).

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

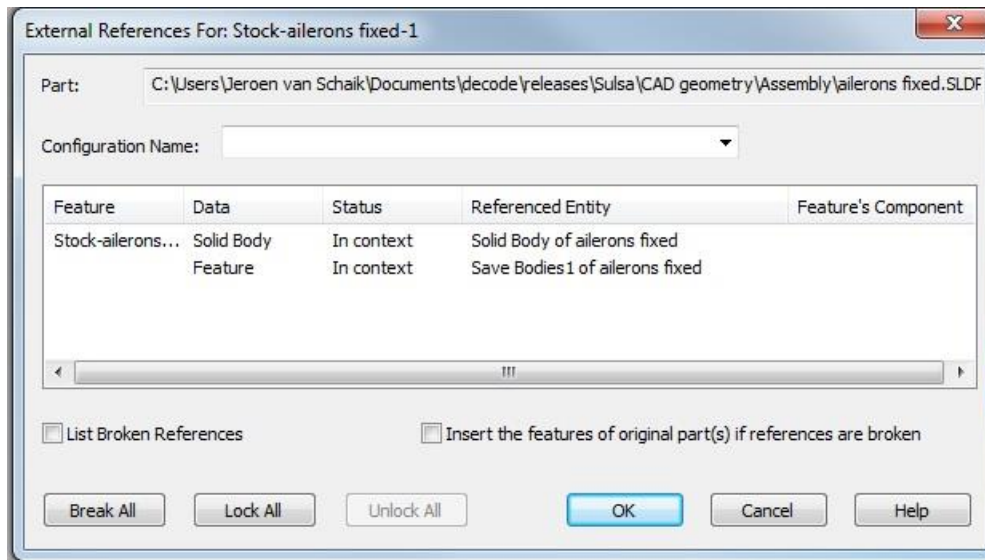


Figure 32 Geometry model structure feedback displaying external references in Solidworks

The utility of these algorithms would come from retrieving existing parts either for model reuse, partial model reuse or to increase the reuse of existing parts in designs. These systems tend to work by analysing the model topology, by analysing the surface or the features of the geometry and creating graphs of the structure of the surfaces such as in (Ma, Huang, and Wang 2009; Bai et al. 2010; Li, Zhang, and Fuh 2010) and comparing these to a library of existing parts to find geometries with a similar structure.

There are several different approaches for creating graphs of the part structure: face adjacency (Ma, Huang et al. 2009) and Feature dependency (Bai, Gao et al. 2010, Li, Zhang et al. 2010).

The feature dependency methods work based purely on feature adjacency and child features, and don't explicitly take into account modelling relations created by features driven by common sketches and constraints.

The use of graphs of the geometry model design intent structure as a tool to help users understand the structure of geometry models is implied by (Li, Zhang, and Fuh 2010) but not investigated further.

A secondary area of research into graphing the structure of geometry models is the exchange of geometry models between different CAD formats for example (Kim, Pratt et al. 2008, Zhang and Luo 2009) . This process works by identifying

or inferring design intent as a graph in origin format, and then transferring the design intent to the target format.

7.1.1 The use of graphs as tools for design.

Graphs and block models are widespread in engineering and process management. Graphs can be used to display the underlying structure of models where the connections between entities are not directly visible, for example in the work of (Kankuzi and Ayalew 2008) on creating graphs of the underlying structure of spreadsheets.

The tracking of the evolution of a design by means of displaying a family tree of design versions from a Product Data Management system is described by (Demian and Fruchter 2009) with the view of facilitating design knowledge reuse.

Keller, Eckert et al. (2009) propose the creation of change propagation graphs from a component dependency matrix as a tool to predict the risk of change propagation during concept design, and establish the least risky order for a gradual design freeze.

See (Lee, Kim et al. 2010) for a design history tracking algorithm for geometry models including the order of application of the features, targeted at speeding up model editing during reuse.

7.2 Requirements for a graphing tool

Section 5.2 describes general requirements for the integration of design rationale functionality in the geometry modelling environment. This section will apply these general requirements to a specific tool to display design intent and design rationale. The following requirements can be identified:

- Display geometry model structure for parts and assemblies.
- Show all the model dependencies, including dependencies between parts and assemblies.
- Be able to retrieve and display design rationale associated with the geometry.

7.3 Practical implementation of a geometry and rationale graphing tool

The tool presented here, 'GRapher', allows the user to view a graph of the complete structure of the geometry model, together with the associated design rationale to explain the design and modelling decisions. This is done by means of a graph showing an integrated view of the geometry model operations, dependencies, parameters, equations and design rationale. These graphs can help the user in the following tasks:

- Allows the tracking of design parameters, input and driven dimensions, externally driven parameters and modelling dependencies through the model, creating an explicit view of the design intent present in the model.
- The inclusion of linked design rationales in the graph can provide detailed information on both the design and modelling decisions.
- The view of model dependencies allows the design re-user to rapidly determine the downstream changes of model edits or partial deletions of features.

The GRapher tool creates geometry structure graphs 'as modelled', rather than as a volume decomposition of a part as machined as in (Shah 1991). The graphing functionality presented here creates graphs of based on geometry model dependencies, rather than feature dependencies. In practice the feature dependency acyclic graph (FDAG), as defined by Li, Zhang et al. (2010), is a subset of the model dependency graph.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

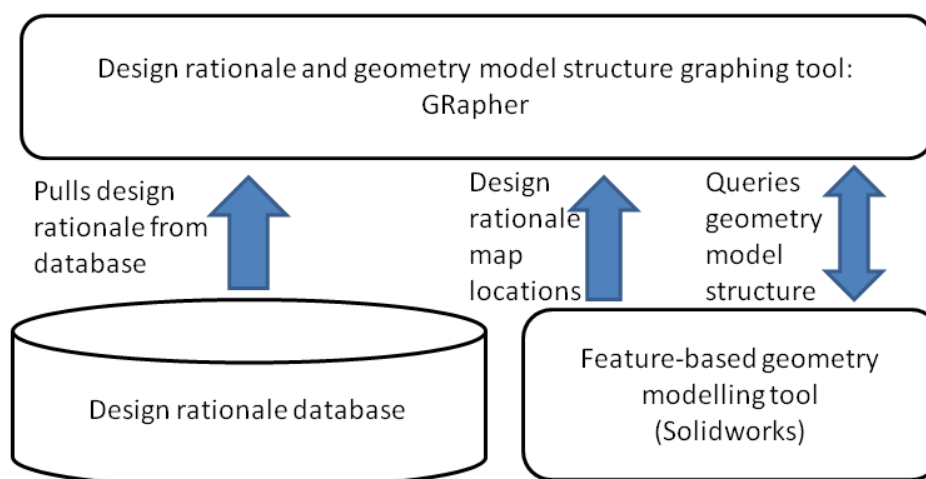


Figure 33 Interactions of the GRapher tool with the geometry model and the design rationale database

The method used to create graphs of geometry models is through the API interface of the chosen geometry modeller, Solidworks. See section 6.3.1 for the rationale for choosing this particular geometry modelling tool. While this particular tool has been implemented for Solidworks, it should be possible to create similar tools for any feature-based geometry modeller that gives API access to the model properties. The .net graph library Quickgraph¹⁰ was used to provide the graph data structure and search functionality as well as serialization.

Figure 33 illustrates the interactions of the GRapher tool with both the design rationale database and the geometry modelling tool. Figure 34 displays a block diagram of the graphing process.

¹⁰ <http://quickgraph.codeplex.com/>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

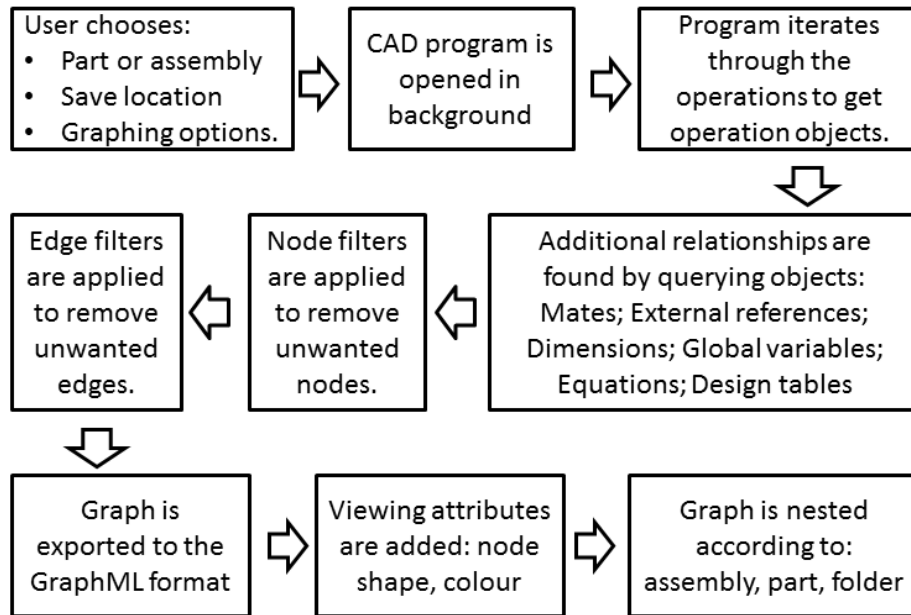


Figure 34 Method used by the geometry graphing tool to create graphs of model structure and design rationale

Prior to starting the graphing process, the user needs make the following selections:

- The location of a Solidworks part or assembly file.
- A name and disk location to save the graph produced.
- A set of options to filter or modify the graph is provided, as shown in Figure 38.
- Optionally, a design rationale database and login details from which the associated rationale can be retrieved.

Once the graphing process starts, the geometry modelling tool is started in the background and the part or assembly is opened. The initial set of nodes is created by following the order of operations and sub-operations in the feature tree. Once the main nodes in the model have been found a number of further operations can be performed to identify further attributes or properties of these features, such as: Mates, Dimensions, External References and Equations, as well as links between nodes in the graph, which indicate the hierarchy or relations of the nodes. Up to the point where the graph is saved the approach is object oriented, so from any node a reference to the object in the geometry model can be found retrieved and any operation or functionality which is made available by the API in the geometry modeller can be used to gather further information on the operations and their links.

7.3.1 Iterating through the operation tree

The first step in the graphing process is a top-down iteration through the modelling-tree, starting at the model level of the part or assembly and following the modelling-tree to get a set of objects for the operations in the model, which provide the core set of nodes for the geometry model graph.

As an object oriented approach is used, therefore an object reference for each item in which is added to the graph is retained, which allows further information and properties to be extracted from the model. In addition to the object, a set of metadata is collected and added to node, which are subsequently serialized with the graph.

7.3.2 Establishing additional connections

Based on the user-selected options the second step in the graphing process is the creation of the secondary structure of the graph. Not all of these relationships are found in an FDAG, and provide an insight to the more advanced associative relationships in the model.

The following options are available for user selection:

- **Get Parents:** retrieves the parent-child relationships
- **Get Mates:** Retrieves assembly mate relationships and attempts to find entities references by the mate relationship
- **Get Equations:** Retrieves equations and attempts to connect equations to the input and output dimensions
- **Get External References:** retrieves external references inside and attempts to connect the referenced entities.
- **Get Design tables:** Checks for dimensions which are controlled by design tables.

Depending on the type of relationships, these steps can involve the creation of additional nodes representing equations and mates, while the relations found by the 'Get Parents' and 'Get External references' are represented as edges. If a dimension is found to belong to a design table, this information is added to the node metadata. The Edges created by the 'Get Parents' option correspond to the parent-child relationships shown in Figure 31, as well as the edges in a feature dependency acyclic graph.

7.3.3 Retrieving design rationale

As the attributes which store the design rationale links are present in the modelling tree they are added to the graph by the initial iteration step of the graphing process. If the rationale retrieval option has been selected, and the design rationale database can be accessed the design rationale is retrieved from the design rationale database. The attribute reference contains the identifier of the design rationale node, which is used to query the MySQL design rationale database. The resulting design rationale nodes and edges are added to the graph, with identical icons as used in compendium to facilitate recognition.

7.3.4 Filtering graphs

Problems in understanding the structure of a geometry model are most likely to occur for more complex parts and assemblies. The GRapher tool is targeted at providing design intent information for complex parts and assemblies. The large number of nodes and edges in the graphs of complex parts or assemblies causes problems in the usability of the graph as a tool for understanding the structure of the design. To reduce the size of the graph filters can be applied based on user selected options to remove nodes which are not relevant.

Li, Zhang, and Fuh (2010) propose a graph filter which removes leaf nodes from the graph for a set number of iterations, to retain only the core structure of the graph. The graphing tool presented here takes a different approach using condition based filters. Once a full set of nodes and edges representing the model has been created filtering operations are executed to remove edges and nodes which are not relevant to the user. For example it can filter out orphaned vertices or remove edges of a certain type between features and sketches. The types of filtering needed will depend on the user's purposes in creating the graph. The filtering options can be accessed through the user options menu in Figure 38.

7.3.5 Graph nesting

In addition to filtering the graph, the creation of nested sub-graphs according to the geometry model structure is an important tool to facilitate the use of large graphs. The user can collapse sub-graphs which are of no interest to reduce the size of the graph laid out at any one time.

The following grouping options are provided:

- Assembly
- Part (model)
- Folder
- Rationale

The graphml format supports nesting graphs (Brandes, Eiglsperger et al. 2002), however the Quickgraph graph library doesn't support the serialization of nested graphs (Quickgraph Serialization). For this reason the nesting of the graph is performed after serialization by editing the xml based on the metadata serialized with the nodes.

The nesting operations are applied from rationale to assembly level in bottom up a nesting order. This method of grouping corresponds to the structure of geometry model parts and assemblies: assemblies containing parts containing folders containing features which may have attached rationale. However, the current implementation of the grouping is non-recursive, and therefore it's unable to nest assemblies containing sub-assemblies.

7.3.6 Serialization, layout and viewing of graphs

The graphs are serialized to an XML based graph format, graphml¹¹ (Brandes, Eiglsperger et al. 2002, Brandes, Eiglsperger et al. 2004), which can be read and edited by a variety of graph editors, such as Gephi¹² and yEd¹³. Additional formatting for the YEd graph viewer, such as node colours, titles, shapes and icons were added by iterating through the graphml file and adding XML attributes for yEd viewing based on node metadata.

¹¹ <http://graphml.graphdrawing.org/>

¹² <https://gephi.org/>

¹³ <http://www.yworks.com>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:y="http://www.yworks.com/xml/graphml">
3   <key for="graphml" id="d0" yfiles.type="resources" />
4   <key for="edge" id="d10" yfiles.type="edgegraphics" />
5   <key for="node" id="d6" yfiles.type="nodegraphics" />
6   <key id="Label" for="node" attr.name="Label" attr.type="string" />
7   <key id="Type" for="node" attr.name="Type" attr.type="string" />
8   <key id="MetaType" for="node" attr.name="MetaType" attr.type="string" />
9   <key id="Group" for="node" attr.name="Group" attr.type="string" />
10  <key id="OwnerModel" for="node" attr.name="OwnerModel" attr.type="string" />
11  <key id="IsInput" for="node" attr.name="IsInput" attr.type="string" />
12  <key id="IsDesignTable" for="node" attr.name="IsDesignTable"
  attr.type="string" />
13  <key id="IsFulyDefinedSketch" for="node" attr.name="IsFulyDefinedSketch"
  attr.type="string" />
14  <key id="HasExternalReference" for="node" attr.name="HasExternalReference"
  attr.type="string" />
15  <key id="DimValue" for="node" attr.name="DimValue" attr.type="string" />
16  <key id="RationaleGroup" for="node" attr.name="RationaleGroup"
  attr.type="string" />
17  <key id="ID" for="node" attr.name="ID" attr.type="string" />
18  <key id="Assembly" for="node" attr.name="Assembly" attr.type="string" />
19  <key id="Label" for="edge" attr.name="Label" attr.type="string" />
20  <graph id="G" edgedefault="directed" parse.nodes="21" parse.edges="50"
  parse.order="nodesfirst" parse.nodeids="free" parse.edgeids="free">

```

Figure 35 Graphml header including definition of geometry metadata

The graphml format explicitly allows extensions with additional attributes to be defined for either nodes or edges (Brandes, Eiglsperger et al. 2002, Brandes, Eiglsperger et al. 2004). Figure 35 shows the header file, which provides the xml definitions for the geometry model and rationale attributes used by GRapher.

Table 7 Content of metadata stored in graph

Attribute name:	Content data type:	Information content:	Used for node types:
Label	String	Name of the entity in Solidworks	All
Type	String	Low level type, e.g. extrusion, sketch, plane	All, where available
Meta-type	String	High level type, e.g. dimension, feature, equation	All

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Group	String	The group or folder to which the entity belongs	All
Ownermodel	String	The part to which the entity belongs	All
IsInput	Boolean	Whether the dimension is driven by an equation or design table.	Dimensions
IsDesignable	Boolean	Whether the dimension is part of a design table.	Dimensions
IsFullydefinedSketch	Boolean	Whether the sketch is explicitly defined.	Sketches
HasExternalReferences	Boolean : Integer	Whether the operation has external references, and if so the number of external references.	All operations
Dimvalue	numerical	The value of the dimension	Dimensions and equations
Rationalegroup	String	The rationale map to which the node belongs	Rationale

Figure 36 provides the graphml definition of an example node, in this case a dimension, including further data defining how the node is displayed in YEd.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

```
237     </node>
238     <node id="8">
239         <data key="Label">D2@BottomHoleSketch@Engine bracket v2.Part</data>
240         <data key="Type">swOrdinateDimension</data>
241         <data key="MetaType">Dimension</data>
242         <data key="Group" />
243         <data key="OwnerModel">Engine bracket v2</data>
244         <data key="IsInput">>true</data>
245         <data key="IsDesignTable">False</data>
246         <data key="IsFullyDefinedSketch" />
247         <data key="HasExternalReference" />
248         <data key="DimValue">120</data>
249         <data key="RationaleGroup" />
250         <data key="ID" />
251         <data key="Assembly" />
252         <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
253             <y:ShapeNode>
254                 <y:Geometry height="30" width="420" x="30" y="30" />
255                 <y:Fill color="#FFE44E" transparent="false" />
256                 <y:BorderStyle color="#FFCC00" transparent="false" />
257                 <y:NodeLabel alignment="center" autoSizePolicy="content"
                fontFamily="Dialog" fontSize="14" fontStyle="plain"
                hasBackgroundColor="false" hasLineColor="false"
                modelName="internal" modelPosition="c" textColor="#000000"
                visible="true">D2@BottomHoleSketch@Engine bracket
                v2.Part</y:NodeLabel>
258                 <y:Shape type="parallelogram" />
259             </y:ShapeNode>
260         </data>
261     </node>
```

Figure 36 an example node in the Graphml format

The rationale node icons, which are vital for the visual identification of the Compendium nodes, are hard coded into the Graphml to obtain a self-contained graph.

There is no built in graph viewer or layout functionality included in the geometry model graphing tool to avoid duplicating existing functionality provided by existing graph viewing tools. Graph viewing and layout are done entirely within the YEd graph editing tool, which provides good hierarchical graph layout functions, which correspond well to the inherently hierarchical structure of geometry models. If the YEd graphing tool is installed, the geometry graphing tool provides the option to open the graph in YEd directly.

7.3.7 Colour scheme

The nodes have been colour coded according to their function in the model, and the status of the feature. Table 8 gives an overview of the colour scheme used when the graph is displayed in the yEd graph viewer. Where possible the colours

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

have been matched to those used in Solidworks. For example for design table dimensions, which are pink in the graphical model view, as in Figure 29.

Table 8 GRapher node colours giving status feedback

Entity types:	Display colour:
Parts	Soft green
Features	Cream
Sketches	Lilac
Design rationale	White with icons
Features with external references	Blue
Dimensions	Drab green
Input dimensions	Yellow
Under-defined sketches	Red
Design table dimensions	Pink
Equations	Light blue
Mates	Tangerine
Components	Soft red

7.3.8 User interface

The user interface for GRapher is implemented as a standalone program, which runs the CAD suite, Solidworks, in the background when required. The starting page of the user interface of the GRapher tool is displayed in Figure 37; it allows selection of a Solidworks file, with the option to save the graph under the same

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

filename as the source part, though with the .graphml extension. A similar page exists for the selection of a disk location to which to save the graph.

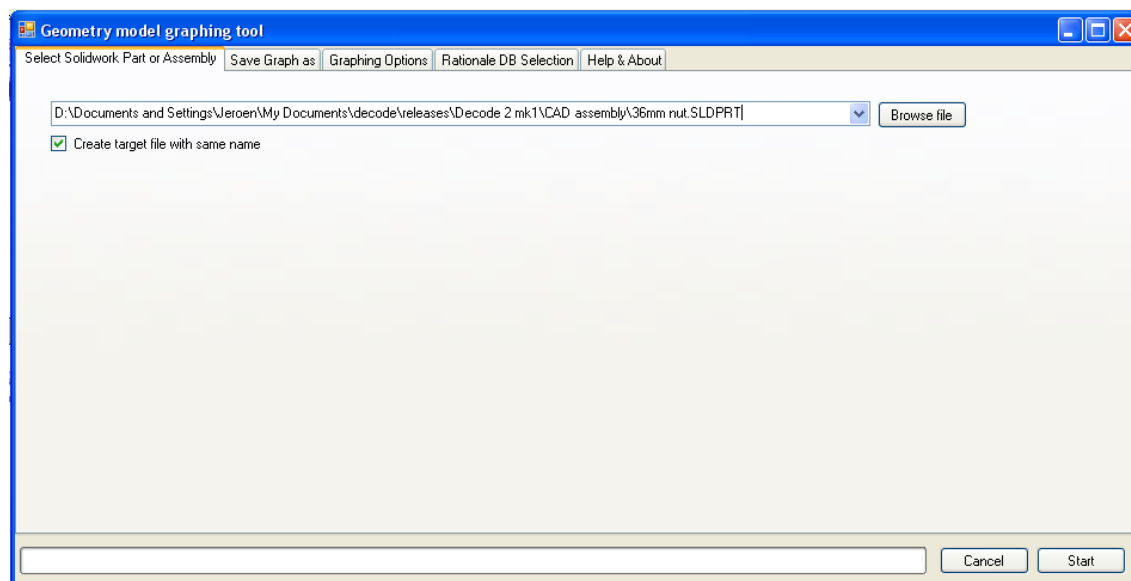


Figure 37 Grapher user interface, file selection menu

The third step is the selection of options to customize the creation of the graph to the user requirements. Figure 38 gives an overview of the options available. A standard set of options which give good al-round results is pre-selected for the user, which can be modified depending on particular interests in creating the graph, for example if the user is interested in equations, but not in 'mate' relationships, the user can un-tick the 'Get mates' box, while leaving the 'Get equations' box ticked.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Items to be added to the graph

Select or Deselect all

- Get Parents
- Get Mates
- Get Equations
- Get External References
- Get Design Tables

Childless nodes to remove

Select or Deselect all

- OriginProfileFeature
- RefPlane
- Reference

Specific nodes to remove

Select or Deselect all

- Front Plane
- Top Plane
- Right Plane
- Origin
- Equation Manager

Edges to remove

Select or Deselect all

- Secondary Dim
- Has Equation
- Has Feature

Graph nesting options

Select or Deselect all

- RationaleGroup
- Group
- OwnerModel
- Assembly

Further options

Select or Deselect all

- Remove Feature To SketchEdge
- Remove Orphan Components
- Remove Orphan Vertex
- Remove Features without dimension
- Mark input Dimensions
- Mark underdefined sketches
- Mark external references

Figure 38 Options presented to the user to allow the customization of the graph created depending on points of interest in the model.

The menu in Figure 39 allows the selection of a design rationale database in which the linked design rationale is stored. A help menu is also provided to aid users in the use of the GRapher tool.

Once the user is satisfied with all the options, the graphing process can be initiated by pressing the 'start' button, after which a progress bar on the bottom of the user interface indicates the progress.

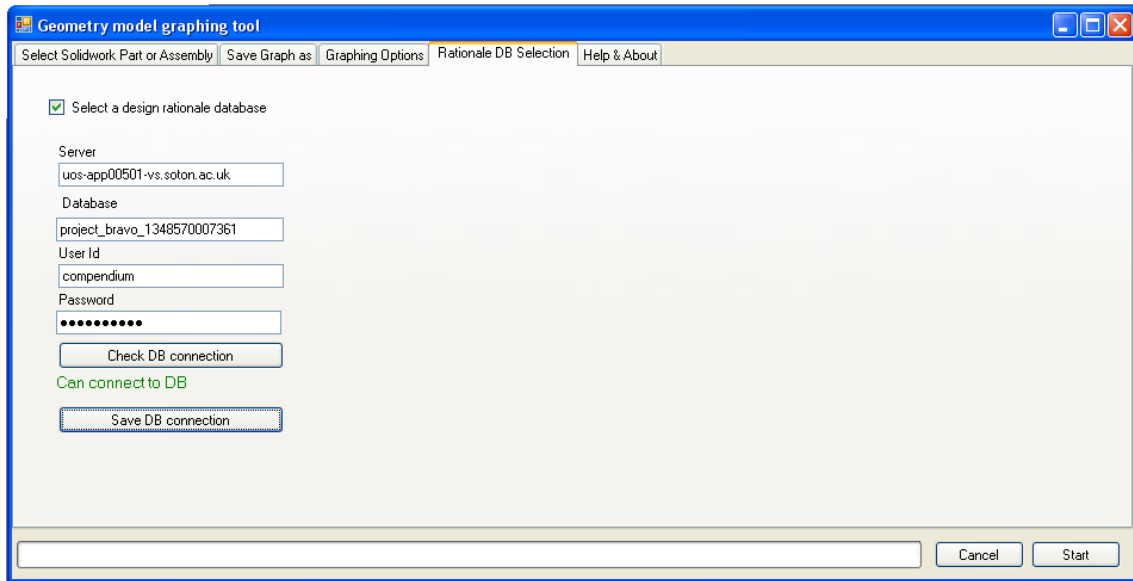


Figure 39 Rationale database selection menu

7.4 Features of graphs produced by the GRapher tool

A few example graphs are next shown to elucidate the contribution that design structure graphs can provide to the understanding of geometry models, and how linked design rationale is retrieved.

7.4.1 Feature relationships

The graph of the part shown in Figure 29, and its associated design rationale shown in Figure 30 is the graphs shown in Figure 40. Elements present in every model, such as the origin, and orientation planes have been filtered out using the options shown in Figure 38.

While the model shown in this graph is a simple due to space constraints, the approach yields useful graphs for larger models and assemblies. Figure 44 shows the graph of a large assembly. While individual nodes are not readable at this size, when used in a graph viewer the zooming facility allow rapid switching from a detailed view of individual parameters to a global overview. In this case the particular point of interest is the way in which the equations and parameters in the assembly drive the features in the parts.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

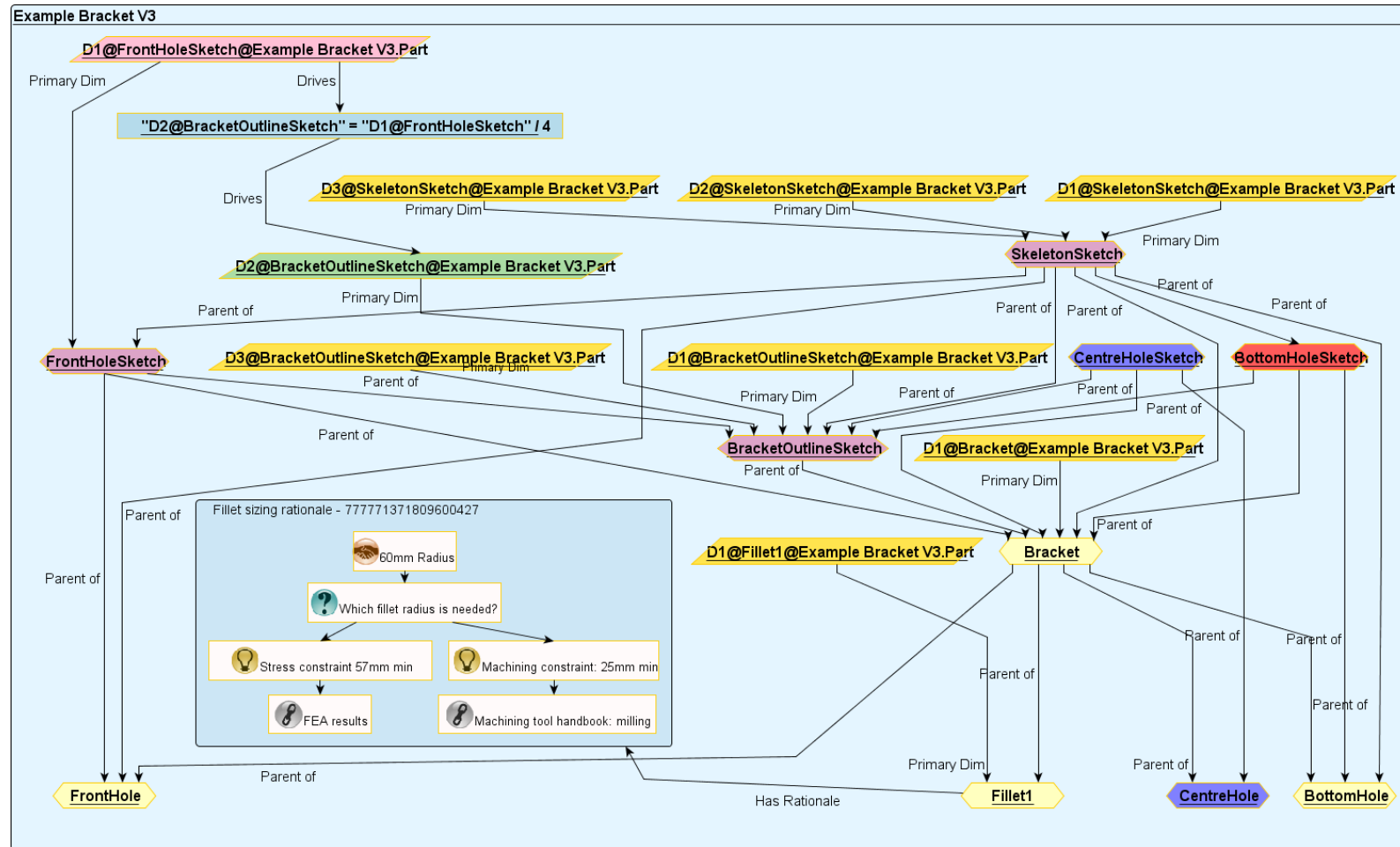


Figure 40 the basic geometry model structure for the part shown in Figure 29, including the design rationale map linked to the model as shown in Figure 30.

7.4.2 Assembly relationships

This section discusses the model relationships which exist in assemblies and which create modelling connections between parts or between parts and the assembly. Three types of such connections can be distinguished:

- Mates, which define the respective geometrical placement of parts.
- External references, which define relationships by modelling operations.
- Equations, which define relationships between parameters.

To create an understanding of such relationships in assemblies, this can be difficult to track in the geometry model,

GRapher attempts to trace these links to the relevant operation or parameter elsewhere in the model.

The following example shows how assembly mates are displayed in graph format.

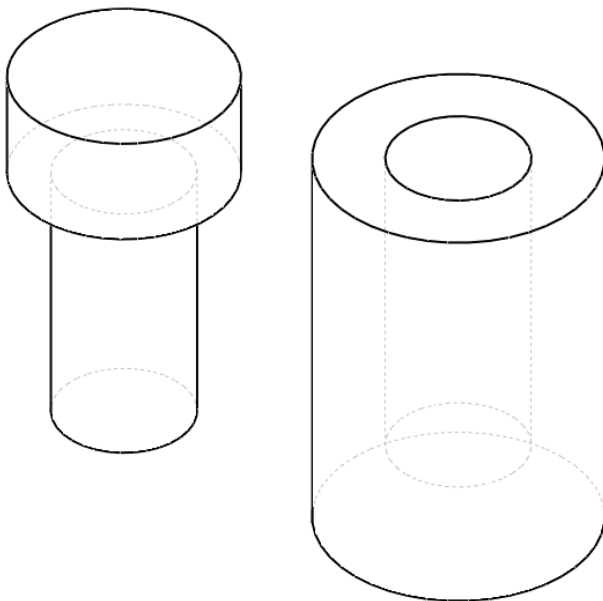


Figure 41 the constituent parts of a simple assembly prior to adding mate constraints. Based on the example in (Shah and Rogers 1993).

Figure 42 shows the resulting assembly created from the parts in Figure 41, by means of adding two assembly relationships. Figure 43 shows the graph of both parts in the assembly.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

The presence of mate relationships between entities in the model frequently implies that further relationships exist between the features of the parts, the Grapher tool can trace mate relationships back to the feature in the geometry model and create a link between the mate entity and the features to which the mate refers.

As the mate relationships are between different features on both parts, each part has two features which are connected to a mate relationship node. As can be observed in the operations tree in Figure 42, the assembly contains two mate relationships, one concentric, and one coincident. These two mate relationships are displayed in the mates folder of the graph in Figure 43.

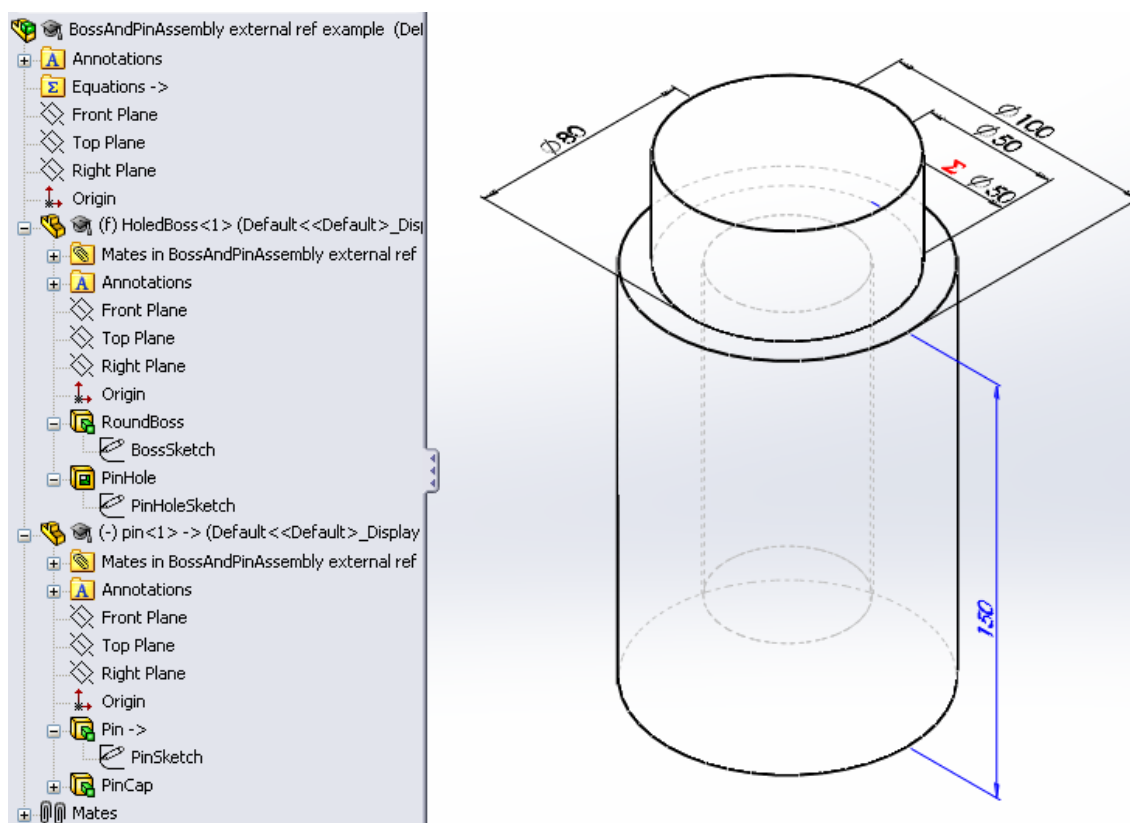


Figure 42 an assembly of the parts in Figure 41 including mate relationships, an equation and an external reference.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

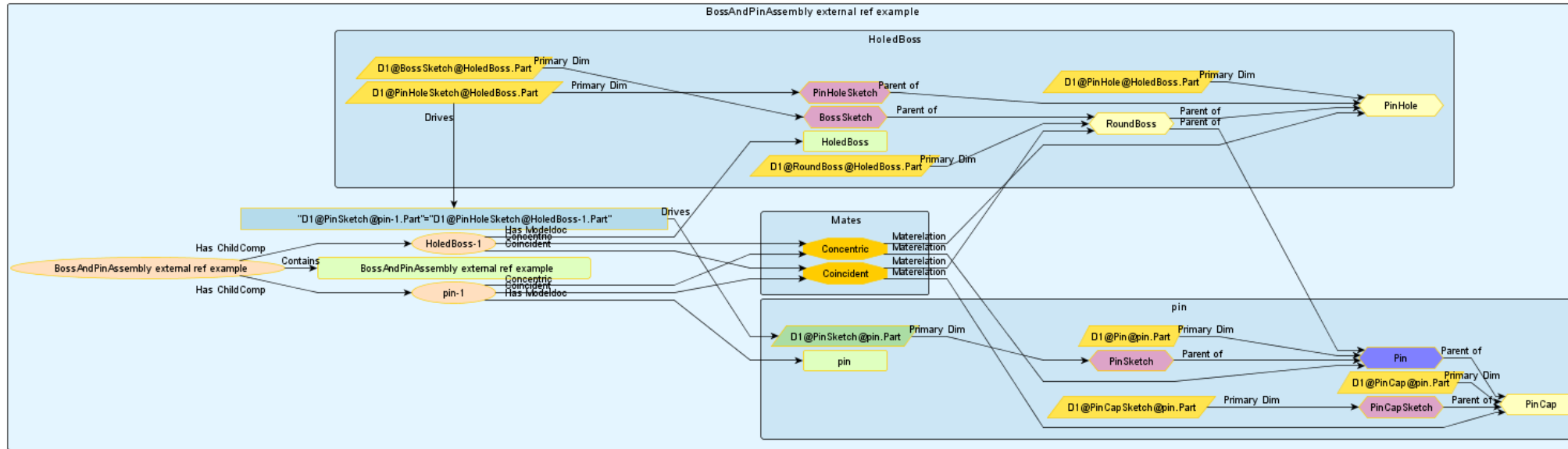


Figure 43 The graph resulting from the assembly in Figure 42 including the mate relations between features, an equation and an external reference.

7.4.3 Creating geometry graphs for large parts and assemblies

The graphs created for assemblies and parts containing a high number of features, sketches and dimensions are correspondingly large. However, as the graphs are ordered into sub-graphs by assembly, part and folder, it is possible to collapse those parts of the graph which are not needed.

Figure 44 shows the graph of a large assembly, showing the inter part equations and design table dimensions. The individual parts are contained within boxes, making the flow of information between the assembly and the parts, or between the parts clear. The hierarchy of the assembly is laid out from left to right.

As the graph viewer allows rapid zooming of the graph, a user can go rapidly from an assembly-wide overview of the structure, to a detailed view of the relationships.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

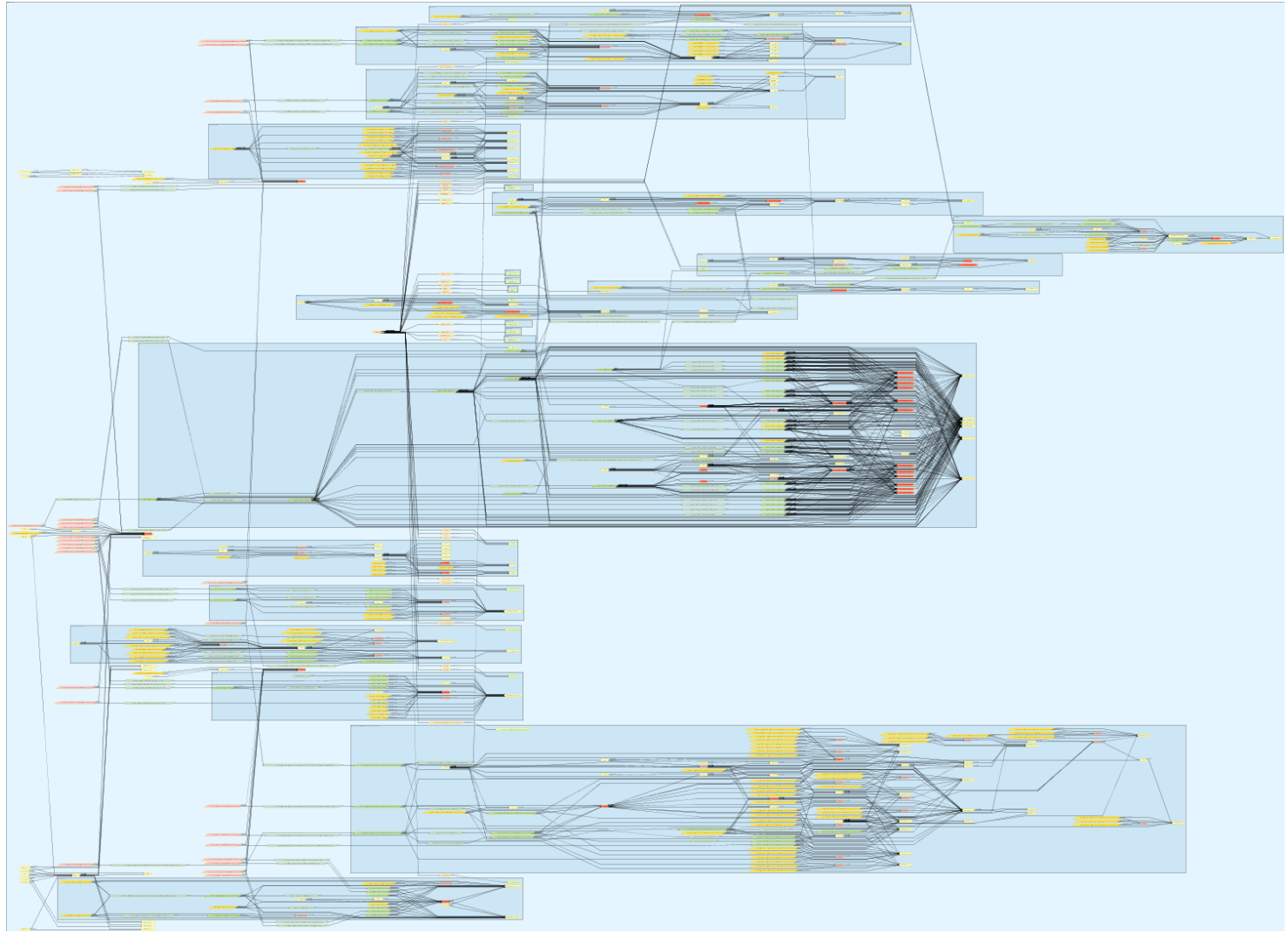


Figure 44 Geometry model graph of a large assembly.

7.5 Applications of the graphs produced

A graph of the combined geometry model structure and design rationale can serve as a form of design documentation, documenting both the design intent including the rationale for the part or assembly. Once the graphs have been created they exist separately from both the design rationale databases and the geometry models and can therefore alleviate any fears of lack of future access to either the geometry model or the rationale database.

A second use of the graphs is to aid the understanding of the geometry model structure for persons who have no intimate knowledge of the model. Cases where this can occur include design reuse, collaborative editing, and design review. The hierarchical structure of the graph allows the user to quickly see if a group of features, which, together, create a function of the design, has downstream dependencies. This can help understanding change propagation within the model.

The model structure graphs can provide information on change propagation within the geometry model, which can benefit geometry model editors requiring an exhaustive understanding of design intent before making model changes. In addition if the geometry model includes inter-part relationships, either through external references or mates between parts, it can provide design change propagation information. As such it can help inform the creation of change propagation modelling matrices proposed by (Clarkson, Simons et al. 2004, Koh, Caldwell et al. 2013).

One particular use of the geometry structure graphs has been the diagnosis of parametric design problems during the marking of student parametric modelling assignments: upon the diagnosis of the model failing to respond to parametric inputs, the graphs allow the marker to trace the parameter from the design table controlling the design and then through the assembly and to the features which should be controlled by the dimension. Figure 44 shows such a graph. This is especially useful considering the high number of designs to be marked, allowing little time to familiarize with the design and the high variation in implementation of the parametric functionality between students.

7.5.1 Typical usage scenario

A typical application of the graphing software is to understand the structure of the geometry model, which consists of a dual structure that reflects both design decisions of the part as well as the modelling decisions. The need to understand unfamiliar geometry models is greater for complex geometry models, as these have more modelling elements, such as parameters, operations and constraints. The incentive to reuse complex geometry models is high as the amount of time and effort invested in creating the model makes them harder and costlier to replace.

This typical usage scenario will consider the following case: a user is presented with a model of an assembly of parts with high level parametric functionality driven by a design table placed in a spreadsheet. However, when trying to make the required changes the user finds that changes to the parameters in the spreadsheet do not produce the desired changes in the model. This particular case is poorly supported by the 'parent-child' type functionality commonly provided in geometry modelling tools, as this functionality only gives information on the relationships between operations, rather than parameters.

The geometry model audit tool discussed in chapter 8 would be able to give the user a first indication of the global type of parametric relationships in the parts, but would not enable the user to trace the path of the parameters as they flow through the assembly.

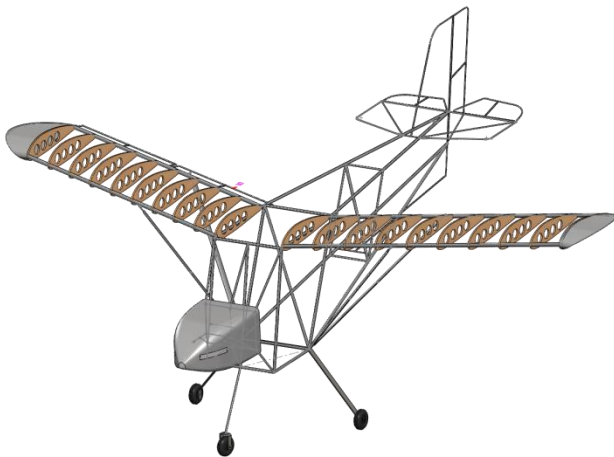


Figure 45 An example assembly to which parametric modifications have to be made.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

An example case, represented by the rough parametric assembly of a microlight aircraft is shown in Figure 42. In this case the user knows the name of the parameter in the design table that should drive the geometry, and the name of the operation which does not function as expected.

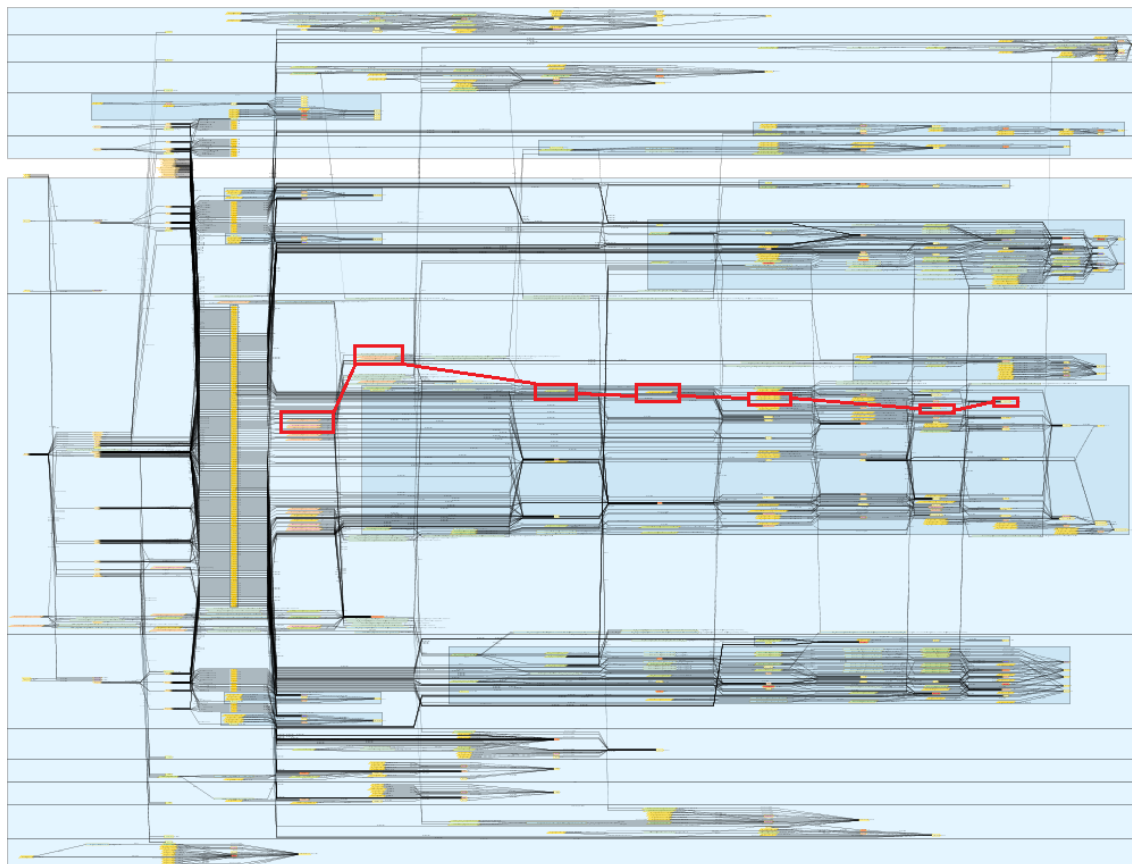


Figure 46 The graph created by the GRapher tool for the assembly in Figure 42 showing the chain of parameter blocks which pass from the design table dimension on the left, to the operation producing the geometry on the right.

The user searches the graph for the non-functional operation and through the secondary dimensions finds the faulty equation. As the parameter is passed through multiple equations, global dimensions and dimensions which are in two different assemblies and a part it is difficult to track the parameter through the model in the normal user interface of the geometry modelling tool.

As the user can view the neighbourhood of the selected node on the graph in a sidebar of the yEd graph viewer, tracking a dimension through a complex graph is relatively easy.

7.5.2 System usage recommendations

The user-friendliness of the graphs created by the tool depends for a large part on how well the user has followed geometry modelling best practices. To obtain the full functionality of the graphing tool it is recommended for the geometry model creator to:

- Give names to features and sketches, particularly for those that are highly connected in the model.
- Group operations in folders of no more than roughly 30 features.
- Group operations in folders by function or sub-function.

However, as these guidelines correspond closely to good geometry modelling practice, they should not create a significant additional workload for geometry model editors.

7.6 Summary

This chapter describes the creation of a tool to create graphs of geometry models and their associated design rationale. The methods used to create these graphs are discussed, including some example graphs created from selected geometry models to illustrate the graphing process and provide a background for a discussion on the utility of such graphs during the design process.

8. Audit of geometry indicating the presence of linked design rationale

This chapter describes the design and implementation of a tool, GReAT (Geometry & Rationale Audit Tool), to display a coloured overlay on geometry indicating several geometry model properties, as well as the presence of linked design rationale.

This tool is particularly targeted at design reviewers and re-users who need to rapidly create an understanding of the model structure, the type of inputs to the model and the presence of design rationale.

8.1 Related works

The audit of design rationales for logical consistency has a long history; Conklin and Burgess Yakemovic (1991) propose both syntactic and semantic checks of design rationale, with further work by Bracewell, Wallace et al. (2009) for the highly structured DRed design rationale format. The system proposed by (Burge and Brown 2004) proposes a system which performs an integrated audit of design rationale and associated software by means of syntactic and semantic inferencing, performs automated checking of design rationale associated with software code in the IDE.

The use of colour coded indicators to give feedback on design status can be found in the design status indicators in the form of traffic lights (Clarkson and Hamilton 2000), while the colour of a DRed node gives feedback on its status (Bracewell, Wallace et al. 2009). DRed charts are used in conjunction with CAD drawings of the final design as a support for design review and audit meetings (Bracewell, Wallace et al. 2009).

The use of augmented views of geometry for design reviews is proposed by (Uva, Cristiano et al. 2010), where the use of an augmented reality system to provide product information, such as stress analysis results and design annotation over the engineering drawings and 3d models of the product.

8.2 Requirements for a geometry and design rationale audit tool

During a design review of a geometry model the reviewers, who per the recommendations of Parnas and Weiss (1987) should be outsiders to the particular design process but experts in the field, will need to quickly familiarize themselves with this particular geometry.

Two particular questions can be identified when a reviewer is trying to understand the design: whether or not design rationale exists for a specific feature or area of the geometry, and what drives the geometry in question; external references, design tables etc. Design rationale capture is especially critical for the input parameters in the geometry design as these are clearly design decisions, and frequently drive other dimensions further in the design (Chan 2007). The presence of geometry which is created from sketches which are not fully defined is a clear indication of either sloppy design thinking or geometry modelling and is not acceptable.

The following requirements can be identified for a tool to aid design reviews by a visual feedback of geometry model status:

- Provide visual feedback on the presence of linked design rationale for areas of the geometry.
- Allow quantitative filtering of design rationale to check substantial rationale has been captured.
- Provide feedback on the status of areas of the geometry such as input dimensions, external references, under-defined sketches and externally driven dimensions, as these can provide information on quality and influences on this area of the geometry.

8.3 Implementation

The design audit tool GReAT uses the GRapher geometry model design intent graphing tool discussed in chapter 7 to create a graph of the model structure. Then algorithms, included in Appendix 1: Algorithms for geometry model audit, are applied depending on user selected options to find the nodes that have the selected types of relationships. Then the geometry entities related to nodes are retrieved, which are then used to colour those parts of the geometry which are affected.

8.3.1 Geometry model and database interactions

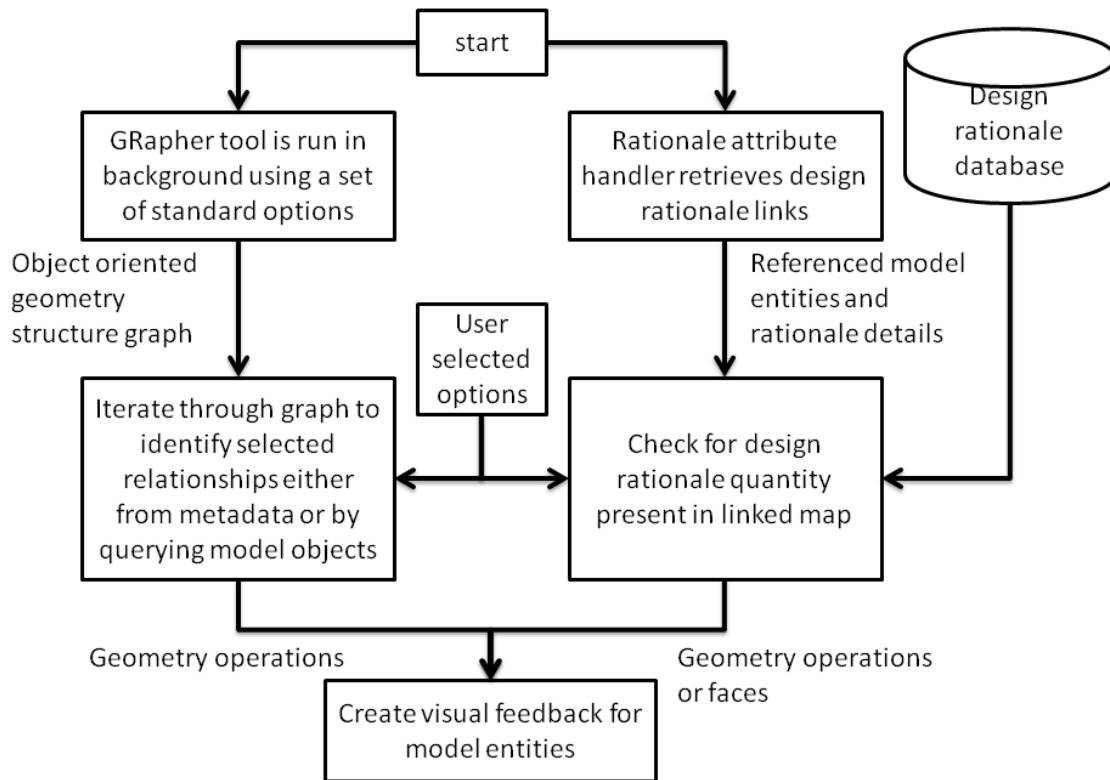


Figure 47 Structure of the geometry audit tool

In parallel to this all design rationale attributes previously created with the RatLink tool are retrieved from the modelling tree, and if a minimum number of design rationale nodes has been specified, the design rationale database is checked for the quantity of nodes present in the linked design rational map. Finally all geometry entities are coloured according to their relationship, in reverse order of priority:

1. External references
2. Design table dimensions
3. Input dimensions
4. Under defined sketches
5. Design rationale

The reverse order of priority ensures that in the case of a geometry entity having properties which come under more than one selected option type, visual feedback shown is for the most significant property.

8.3.2 User interface

The user interface is contained in one menu, which can be accessed through the ‘show rationale’ button on the add-in toolbar shown in Figure 26.

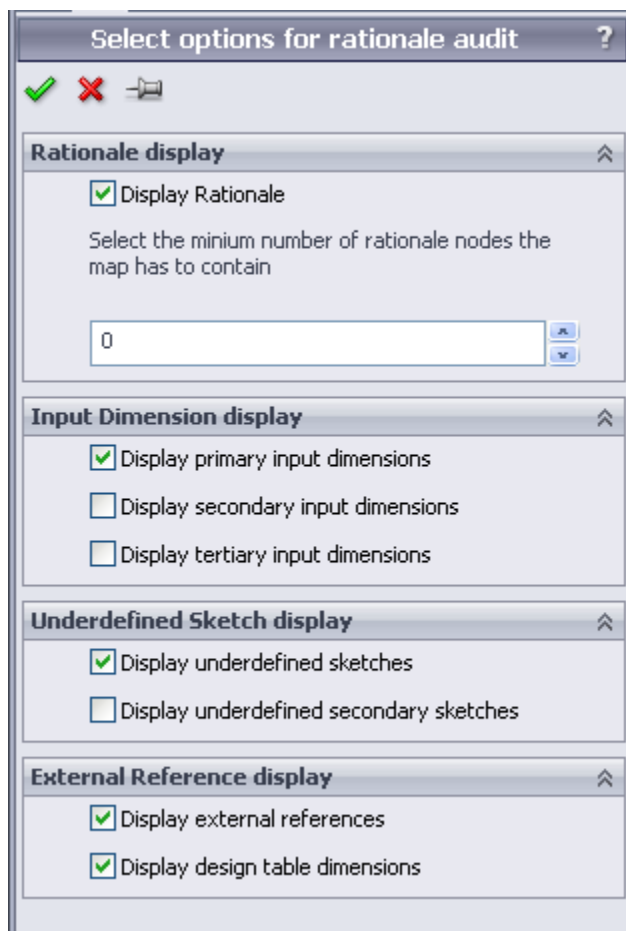


Figure 48 The design audit menu shown integrated in Solidworks. The option selected here result in the geometry seen in Figure 49.

Figure 48 shows the user interface menu, which contains the options which can be used to select which types of relationships are displayed. Table 9 shows the functionality of each of the options.

Table 9 options of the design audit functionality.

option	Functionality
Display rationale	Display the presence of linked design rationale.
Minimum number of	Sets the minimum number of design rationale nodes the

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

rationale nodes	linked map has to contain for the display of the operation as having linked design rationale to be activated.
Primary input dimensions	Display the presence of input dimensions created during the definition of the operation.
Secondary input dimensions	Displays the presence of input dimensions in profile sketches.
Tertiary input dimensions	Displays the presence of input dimensions in other entities used to define the operation, such as skeleton sketches and planes.
Under-defined sketches	Displays the presence of profile sketches which are not fully defined.
Secondary under-defined sketches	Displays the presence of other under-defined sketches, such as skeleton sketches, used to define the operation.
External references	Displays the presence of operations driven by external references.
Design table dimensions	Displays the presence of operations containing dimensions driven by design tables.

8.3.1 Colour scheme

The colour scheme chosen is similar to the colour scheme used for the GRapher tool discussed in 7.3.7.

Table 10 shows the display colours used. The colours are highly transparent, so that the properties of the features which are not directly visible from the outside of the model can be seen without sectioning the geometry.

Table 10 Design status feedback colours.

Features with:	Display colour:
No attributes	Translucent
Design rationale	Green
External references	Blue
Input dimensions	Yellow
Under-defined sketches	Red
Design table dimensions	Pink

8.4 Examples of the visual feedback of produced by the GReAT tool

To demonstrate the feedback this tool can provide to a design reviewer or re-user trying to understand the geometry model, the result of the geometry audit for a simple and complex geometry will be shown and discussed.

Figure 49 shows the result of a geometry audit on a simple geometry. In the operations tree on the left we can see the attribute attached to the 'Fillet1' feature, providing an additional indication of the presence of design rationale. The colour 'buttons' to the right of the feature tree give further feedback on the status of the corresponding feature.

The geometries shown in Figure 49 and Figure 50 show the results of a geometry audit of a simple and a complex model. By means of options, as shown in Figure 48 the user can choose the type of audit to perform.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

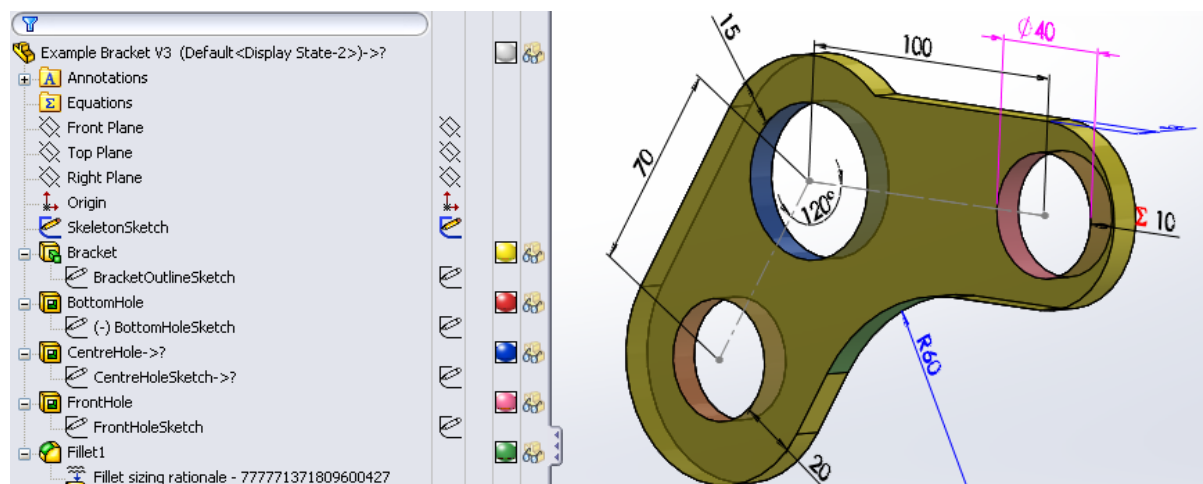


Figure 49 The example geometry from Figure 29 after design rationale and geometry audit.

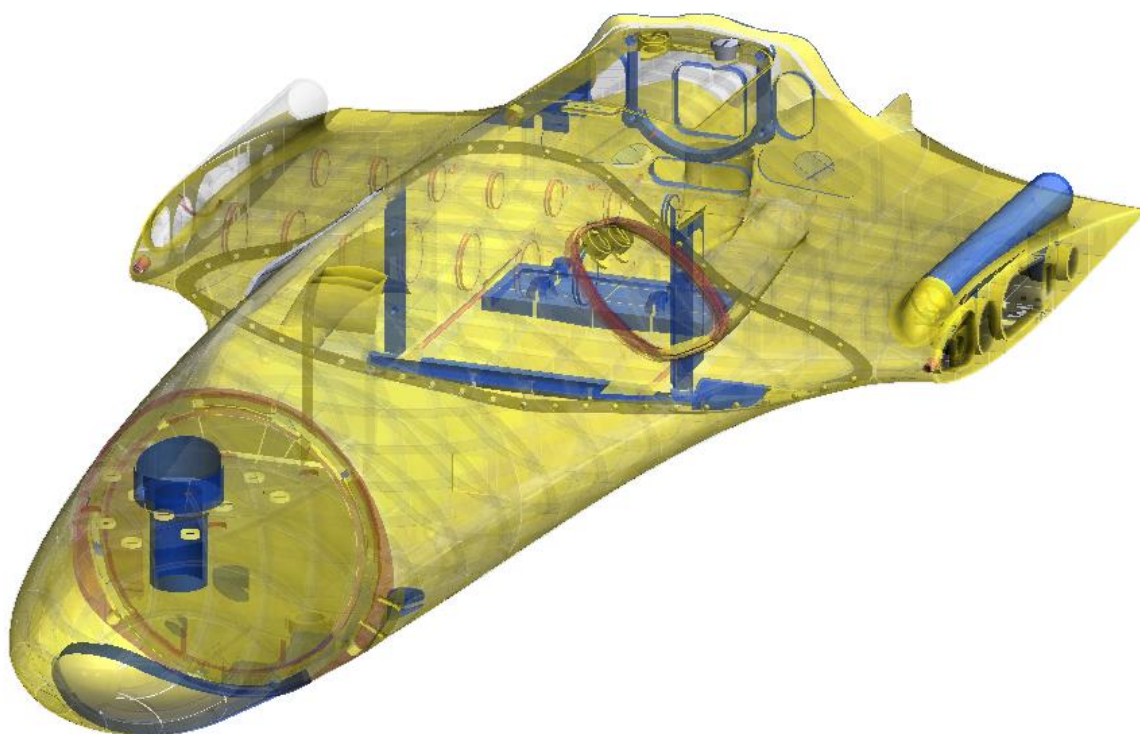


Figure 50 The audit tool applied to a complex piece of geometry, an UAV fuselage

By adding rationale explaining input dimensions, the geometry modeller can explain design decisions, and this will be presented in the audit as a green colour. The audit result for a complex geometry model displayed in Figure 50

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

shows a clear absence of features coloured green. This indicates that the model currently doesn't have any linked rationale. Based on the information shown here, the user should make sure to fully define the under-defined sketches, check the correctness of the external references and add rationale for the design decisions.

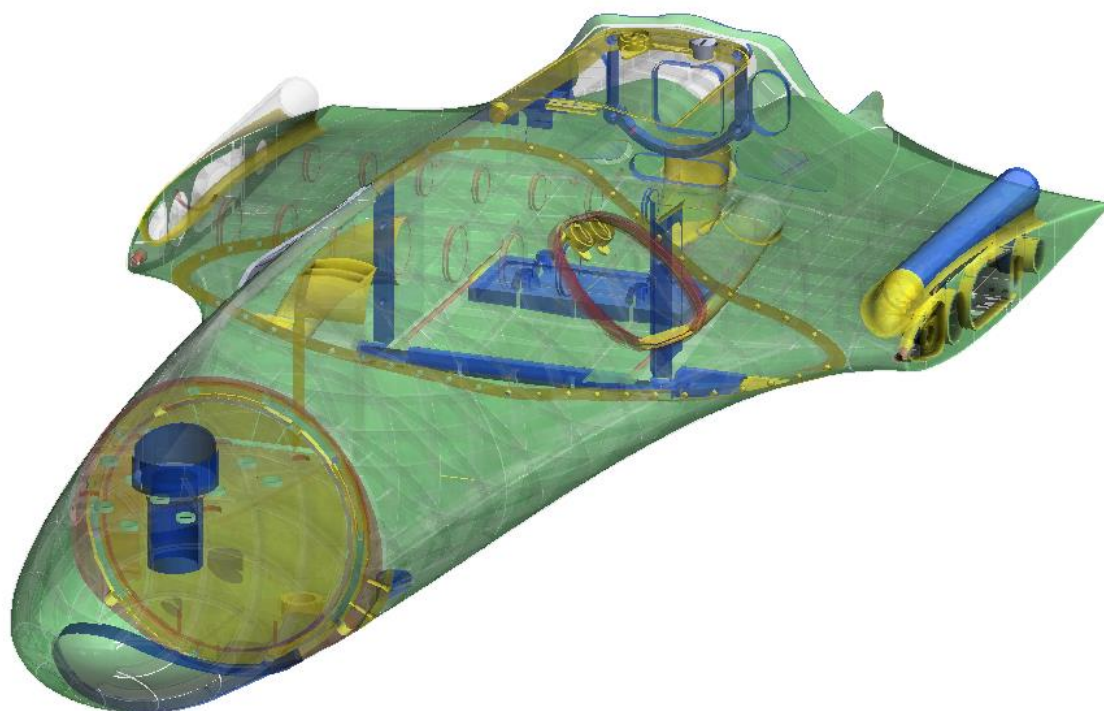


Figure 51 Fuselage design showing result of design audit after rationale has been added for the skin thickness.

In Figure 51 design rationale for the skin thickness of the part shown in Figure 50 has been added to the model. Even though many subsequent features have added to the geometry since the skin thickness decision, the audit function remains capable of tracing this design decision into the finalized geometry. This graphical indication of the design properties allow the reviewer to focus on potential trouble hotspots, and graphically see whether they have linked rationale, speeding up the review process.

8.5 Applications of the geometry audit functionality of GReAT

8.5.1 During design reviews

The purpose of a design review is to make sure that the design fulfils all of the requirements, and that the design decisions made can be justified. In this case a design review of the geometry is considered.

The justification for a design decision can come in many forms, for example: calculations, spreadsheets or design rationale. Without links between the design rationale and the geometry, it is difficult to tell if design rationale has been captured for a particular geometry decision, without searching through a large number of design rationale maps.

The first contribution of the geometry audit tool to the design review process would be to clearly indicate the presence of linked design rationale. The further filtering function can help indicate only the parts of the geometry which have a linked rationale map containing at least a set minimum number of nodes. The geometry in Figure 50, presents a case where no linked design rationale is present, while in Figure 51 and Figure 49 linked design rationale is present for one area of the geometry.

The current prototype version of the tool does not directly allow the reviewer to directly access the design rationale itself from the geometry modelling tool by double clicking on the link, but using the unique identifying number of the linked design rationale map present in the design rationale link, the linked design rationale map can be quickly found in Compendium by searching for the identifying number.

It might not always be practical to capture design rationale for each feature in the geometry, but for instance if a reviewer considers a particular feature of the geometry to be risky or poorly considered, it can speed up the design review considerably if the reviewer knows that there is design rationale for this feature and where to find it.

The second contribution of the GReAT tool is to assist parametric design by indicating the source of the dimensions of the area of the geometry.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

If the part contains red areas, which indicate the presence of geometry based on underdefined sketches, the part clearly needs correction, as this implies the geometry is not fully defined.

The presence of external references indicates the relationships of the part with other elements of the design, implemented as relationships to elements in the assembly (Top down design) or lower level parts in the assembly (Bottom up design). Only driven external reference relationships are indicated by the audit tool. Design table dimensions fulfil a similar function in the design, in that they allow external control over the parameters in a part or assembly. In the case of bottom up design this should not be considered to be a cause for concern, but in the case of unexpected or undesired references the visual feedback can provide a useful indication of their presence. On a practical level the implication of finding either external references or design table dimensions during the geometry model audit is that they are appropriate and up to date; e.g. is the part external reference linked to the latest version of the external part.

The presence of input dimensions indicates design decisions. In principle all design decisions should have some form of design decision rationale supporting them. If a part of the geometry is shown as being based on input dimensions, but not as having attached design rationale, this should serve as a reminder that design rationale should be captured for the design decisions behind these parameters.

It can occur that a part of the geometry remains translucent, which indicates that it doesn't match to any of the audit criteria selected. This particularly occurs for operations which are not based on dimensions, such as 'mirror' operations with respect to one of the origin planes (e.g. the Top, Front or Right plane). This is no cause for particular concern, and no action is required.

8.5.2 As a tool to assist editing of parametric geometry models

While the tool is primarily targeted at design reviews, the colour feedback provided can also assist the editing of unfamiliar geometry models by indicating which areas of the model are underdefined or driven by design tables and external references. In the case of areas of the geometry driven by design tables the user can proceed directly to the table in question and see whether the

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

necessary edits can be made there, without having to look through all of the dimensions which define this feature in the operations or sketches.

8.6 Summary

The design rationale audit tool presented in this chapter provides a visual feedback indicating which features and faces in the geometry model have attached design rationale. Furthermore, it indicates which areas of the geometry are driven by external references or design table dimensions, indicates which features are based on sketches which are not fully defined.

9. Evaluation: field testing of the proposed design rationale linking toolset.

This chapter describes a field trial of the prototype design rationale linking toolset presented in the previous chapters. The main aim of this study was to test whether there is an increase in design rationale capture during geometry design as a result of the ability to link geometry to design rationale.

The main goals of this study are to establish whether:

- The proposed method of linking design rationale functions in a real world collaborative design environment.
- The ability to link design rationale to geometry increases design rationale capture during geometry design.

The general purpose and goals of a Descriptive 2 study is discussed in section 1.5.4. To recapitulate: its purpose is to evaluate how and to what extent the changes to the design process contribute to the success criteria established at the start of the project. The success criteria defined in section 1.5.1 is: “Improve the gathering of in-context design decision information during computer aided design and geometry modelling”. The metric to evaluate the success criteria is the quantity of design rationale nodes captured during the design phases, as discussed in section 4.2.

9.1 Descriptive 2 study: a thought experiment

An ideal experiment would consist of a test and a control group performing an identical design task, using an identical toolset, except for the additions to the toolset being tested.

The baseline toolset would constitute a CAD package, such as Solidworks and a design rationale editor, such as Compendium. Benchmark values for the success criteria should be obtained from a control group; who perform an identical design task with the baseline toolset. The test group would have access to the baseline toolset, as well as the additional tools under test.

The toolset presented in the previous chapters is targeted at geometry design of complex aeronautical engineering products. To test the toolset a user group is required which could benefit from the tool use while performing a design task.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

An ideal design task for the evaluation of the success criteria would encompass the following elements:

- A sufficiently complex design task to benefit from design rationale capture.
- A design task which requires substantial collaboration between designers.
- Need for a full completion of all design stages up to and including the geometry design.
- Substantial usage of geometry modelling during the design process.
- The ability to reuse linked design rationale from previous design work.

An alternative to such a large experiment would be to test key functionalities in small controlled experiments using a control group. An example of such an experiment on the impact of annotations of geometry models on model editing can be found in (Alducin-Quintero, Contero et al. 2011). In the case of the design tools presented in this thesis the use of small experiments would permit validation of parts of the framework, for example a small, controlled, experiment to evaluate the effect of the presence of geometry model structure and design rationale graphs on the speed of design editing. While a small, controlled experiment would permit the validation of individual elements of the toolset, it would be difficult to evaluate the impact over a longer design process or on collaborative design work.

To recapitulate, an idealized experiment would have the following features:

- A test and a control group performing an identical design exercise with and without the changes to the design tools.
- Test populations of sufficiently large size to obtain statistically significant results.
- A sufficiently realistic design exercise, including collaborative design activities.

9.2 Study setup

A group of 77 undergraduate aeronautical engineering students performing a group design project was identified as a possible test population for the following reasons:

- Seven groups performing an identical design exercise in parallel using the same toolset
- A sufficiently large, long and complex project for design rationale capture
- Absence of confidential or commercially sensitive information.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

- The grading and assessment both of individual and group performance gives an additional metric for design quality.

However, the design project describe above is a graded design exercise; therefore it was considered unethical to perform a control group study, as the use of the toolset could give an advantage or disadvantage to the groups having access to it. Despite the disadvantage of this particular test population, the large population, long design task duration, need for geometry design and collaborative nature of the design project were considered to outweigh the inconveniences. The high cost of organizing an intensive, long duration, controlled experiment would be prohibitive, while smaller experiment would make it difficult to see any effects related to collaborative work and the building of knowledge of the design space during the design process.

To test the performance of the research prototype tool, seven groups of student-designers performing a simulated but complex design process were invited to use the tool during the design process. The design process of this design project was not complete, in the sense that the designs created during this project are normally taken only to an embodiment design level, and are not taken into production. However, one of the previous year's projects has progressed into further design stages, including wind tunnel testing and prototyping, with the intention to eventually produce the design commercially.

9.2.1 Data collection

Three main sources for information on the usage of the toolset and general design rationale capture were used during the field testing:

- The analysis of the design rationale database, using the software developed by (Gorissen 2011), with some additional software to measure the usage of the design rationale linking tools.
- The grades of the design teams and peer reviews of design team members.
- Observation of design teams during design reviews, including limited interview questions.

9.2.2 Design assignment

The assignment was quite free and simply formulated: “Design a two-seat **Small Light Aeroplane (SLA or microlight)**, as per CAP 482, British Civil Airworthiness Requirements - Section S”

The students were evaluated based on two outputs: firstly a design report detailing the design process and design and secondly a presentation of the design by the design team, presenting the design and key performance data. The presentation and design were critiqued by a group of Microlight aircraft industry experts. The groups were assessed on presentation clarity, design realism, certification issues and marketability of the aircraft. A third, unmarked, project output was an X-plane¹⁴ model of the aircraft for evaluation in a physics based flight simulator by a professional test pilot to provide feedback to the teams regarding the controllability and performance of their design. The grades were adjusted for individual performance based on a peer assessment by the design team of the individual’s contribution.

9.2.3 Tool availability

The toolset presented in the previous chapters was deployed to university desktop computer workstations and made available to download for installation on the students’ personal computers. Compendium was also available on university workstations, and instructions were provided on how to download Compendium for home use. The login details for the Compendium databases were given out 7 days after the start of the project, after an instructional lecture on the use of Compendium and the design rationale linking toolset. Email support and one-to-one assistance regarding the setup or use of the toolset was available to the designers.

9.2.4 User group

The designers were mainly third year aerospace engineering students. The designers were divided into seven teams of between 10 and 12 designers, with a total of 77 students. Groups were balanced so that the average grade for each group of the student’s previous courses was similar. Some groups experienced

¹⁴ see <http://www.x-plane.com/>

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

dropouts, but final group grades were corrected for group size. The grading of the students was based on team performance with a peer assessed correction for individual effort. The user group can be qualified as consisting of novice designers.

9.2.5 Time schedule and work hours available

Each student was expected to work 160-190 hours on this project, leading to an availability of between 1600-2280 man hours for the project per team, depending on team size. The length of the project was 120 days from assignment to final presentation. The students also had other courses, and the design process was interrupted by exam and vacation periods. Two interim design progress reviews were held 21 and 42 days into the project. The student performance was not assessed at these moments. The design reviews were used to give feedback on the design presented as well as the design progression. Specifically those teams which had not selected a design concept and started geometry design by the second review were strongly encouraged to do so rapidly.

9.3 Study results

Both the 64 bit and the 32 bit version of the design rationale linking tool were downloaded 57 times for use on private computers, with users choosing to download both installers at the same time, and some users downloading the tool more than once. No problems were encountered regarding the use of the design rationale capture toolset during a collaborative design process.

9.3.1 Design rationale capture

Each of the groups captured a substantial amount of design rationale with a difference of a factor of 2.5 between the lowest and the highest group. Figure 52 shows the design rationale capture during the project per team.

While there is a considerable variation in capture between teams, several recurrent patterns can be observed: All teams captured a substantial amount of their design rationale during the initial stages of the concept design. Rationale capture lay dormant during breaks in the design process, such as holidays and

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

exam periods. The majority of teams had a resurgence of rationale capture towards the end of the project.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

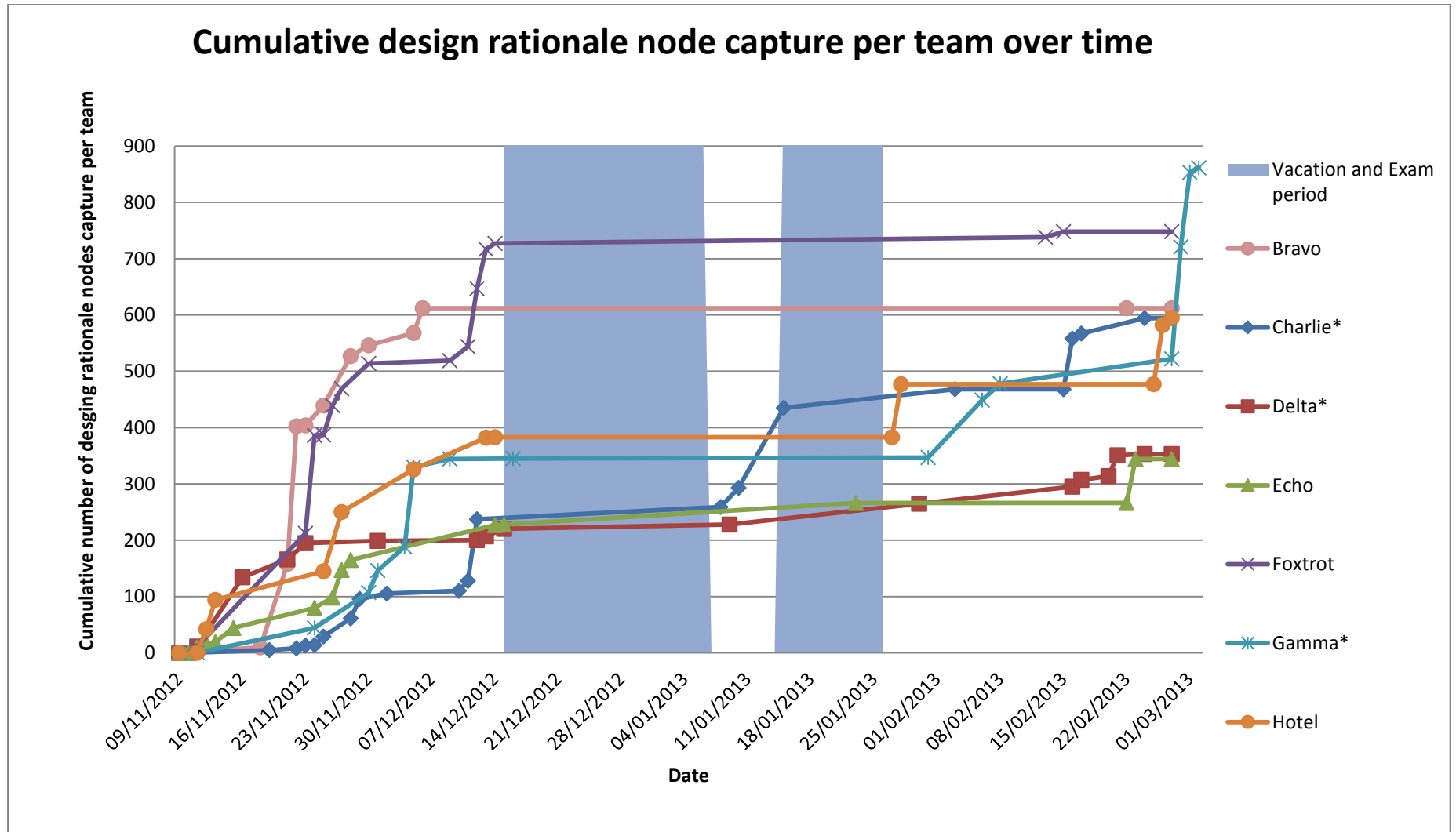


Figure 52 A graph showing the cumulative design rationale node capture per team over time. The groups marked with an asterisk used the design rationale to geometry linking tool.

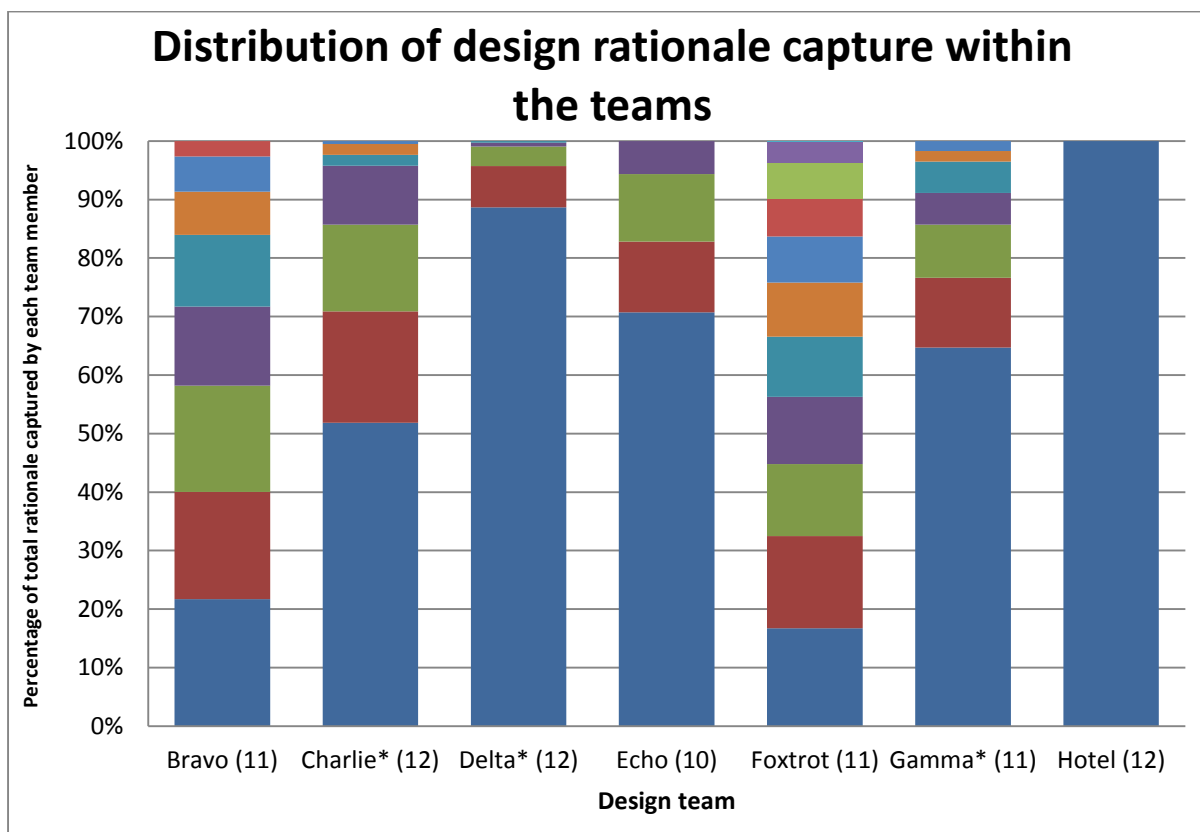


Figure 53 Distribution of design rationale capture per team. Asterisks indicate use of design rationale linking tool, numbers in brackets the number of team members.

The graph in Figure 53 shows the distribution of design rationale capture within the teams. There is quite a large variation between the teams, with team Hotel, where all the design rationale is captured by one person, whereas in another, team Foxtrot the most rationale captured by any one member is 17%. It should be noted that the ‘Hotel’ group indicated during the design review that they had created paper based rationales and then one team member had the task of digitizing them. Persons who capture large amounts of design rationale could be qualified as super-users or technology champions. Various measurements could be used to estimate or define whether a team contains a technology champion for design rationale capture. In this case five out of seven teams had an identifiable champion, defined in this case as persons who each captured over 200 nodes of rationale and between 50% and 100% of the total design rationale per team. See Appendix 5 for a further analysis of the design rationale capture and peer review scores of the population and the super users.

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

9.3.1 Use of design rationale linking tool

The design rationale linking tool was used by three out of the seven design teams. All three teams which used the design rationale linking tool had a technology champion. Of the four teams not using linking to geometry models two teams captured no or very little (less than 2% of total) rationale after concept design. In the case of the five teams which had rationale capture champions, all champions were reviewed favourably in peer reviews, with four out of five being the most favourably reviewed or shared most favourably reviewed team member in the general peer assessment by the design group, indicating that those who captured the most design rationale were also the most active in the design process, and are recognized for their work. As the peer reviews are based on the entire design process, rather than just the design rationale capture it can safely be assumed that those doing the most design rationale capture have also performed an above average amount of the other design tasks.

Table 11 Use of design rationale linking tool

group	Number of linked, non-deleted rationale maps	Instances of linked maps	Unique nodes contained in linked maps	Edges to and from linked rationale maps	Total unique nodes in maps containing a rationale link
Charlie	11	11	0	2	388
Delta	15	15	0	25	41
Gamma	36	45	284	17	113

The three teams which used rationale linking all obtained above average grades for their group design project, and were also the two highest graded groups. The groups using design rationale linking toolset tended to capture on average 4.9% more design rationale during the whole project. When corrected for group

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

size at the end of the project the groups using design rationale linking captured 4.8% more design rationale.

9.3.2 Did the use of the design rationale toolset correlate with higher design rationale capture during the geometry design phase?

The groups using the design rationale linking tool, RatLink, did capture far more design rationale late in the design process, with on average 53% of the total rationale captured during the second half of the design process, versus 18% for the teams not using design rationale linking.

The groups using the rationale linking tool captured significantly more rationale ($\alpha = 0.025$) during the second half of the design process. See Appendix 4 for the data and calculation. See section 9.4.1 for a discussion of the validity of this result.

Only one of the groups placed design rationale within the linked maps, which was the intended practice for design rationales created during geometry design.

9.3.3 Use of design rationale feedback functionality

The user statistics sent back by the geometry and design rationale graphing tool, GGrapher, show that the use of geometry and design rationale graphing toolset was low, with limited usage by only four individual users. In total the graphing functionality was used 14 times, with rationale being included 10 times. However it is possible that the graphing tool was used more frequently offline, in which case usage statistics wouldn't have been received.

No data was recorded for the use of the visual design rationale presence feedback function, so no assessment can be made of the use of this functionality.

The low usage of these tools conforms to expectations, considering that they are targeted at secondary design users, not the original designers.

9.3.4 Feedback from users

Some of the users felt that the lack of ability to view the design rationale directly from the geometry modeller by means of double clicking on the design rationale

attribute made the design rationale linking less user-friendly. “Compendium worked well for us as a design tool, as the whole team could see and add to the rationale, but the Solidworks design rationale linking was of limited use as you couldn’t click on the link and see the rationale.” See the future works section for a proposed solution to this issue.

9.4 Discussion

The pattern of design rationale capture of high capture during concept design and late evaluation and documentation stages is consistent with earlier studies such as in chapter 4; however it is interrupted by holidays and exam periods during which the design rationale capture was very low.

The study shows that linking geometry models to design rationales is possible, and that those groups who used the tool performed well in the design process, obtaining above average grades. Those groups that had a technology champion for design rationale capture were also more likely to link their design rationales to the geometry models. There is also a strong correlation between high rationale capture by an individual and positive evaluation during peer reviews. The centralised storage of the design rationale was viewed positively by the users, as it allowed the whole team to view the design rationale and the options which had been investigated.

The following sections will discuss the interpretation of the results of this field trial.

9.4.1 Self-selection bias

Three out of seven design teams used the design rationale linking tool provided. As the teams decided themselves whether to use the design rationale linking tool, it is not possible to establish a direct relationship between team performance and use of the design rationale linking tool.

It is likely that the teams whose designs were progressing well had more spare capacity to capture design rationale throughout the design process. Furthermore, there is the possibility of an inverse causal relationship, where groups capturing design rationale during the second half of the design process

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

are more likely to try and use tools for design rationale capture during this stage, while groups which are not capturing design rationale during the second half of the design process would have little incentive to use the tools proposed.

A control group study, in which only some teams would have access to the tool was considered, but not carried out as it would be unethical to give a part of the students more help in the form of tools and support. However there is a strong correlation between the use of the design rationale tool and design rationale capture in the second half of the design process, which shows that design rationale linking is at the very least a promising area of investigation.

9.4.2 Possible reasons for low tool use

While the use of Compendium by all the design teams was within the range of expectations, the use of the design rationale linking tool was lower than anticipated, despite repeated encouragement to the designers. Several reasons can be identified for this:

- The lack of incentive for the teams to use the design rationale linking toolset.
- Time pressure on teams to meet deadlines.
- Groups did not perform full detail design, and spent relatively little time during geometry design.
- Designers were relatively inexperienced at designing complete aircraft, and therefore were already experiencing a steep learning curve during the time when the novel rationale linking tool was made available.
- Limits of reality of the study: design outputs in the form of report and presentation rather than a design ready for manufacture.
- Late transition from concept design observed in the teams leaves little time for detail design.
- The capture of design rationales was not a formal goal for the design teams.
- No requirement to do a rationale audit before or during design reviews.

Of the possible reasons given above the lack of incentive for design teams to use the toolset is a recurrent issue in design rationale capture; see section 3.2.1 and 3.6.3 for a discussion. The design teams were had no incentive to use the toolset other than the utility of the tool during the design process itself. The use of the design rationale audit tool, GREAT, on geometry produced during the project could have provided the users with an additional, external, incentive to use the RatLink tool.

The high workload on the designers in the later design stages could have contributed to low capture as design rationale capture or linking was not a formal requirement and therefore may have been seen as a non-essential task.

9.5 Conclusion

This chapter describes the field testing of the toolset proposed as a result of the prescriptive I study in chapter 5 and implemented in chapters 6, 7 and 8.

The results show that although the design rationale linking tool has worked well in a collaborative environment, the tool uptake was low, with only three out of seven teams making use of the tool.

Although the use of the toolset only correlated with slightly higher design rationale capture overall, the teams which did use the design rationale linking tool captured significantly more design rationale during the geometry design stage, although these results may be biased due to self-selection of tool use.

This study illustrates the difficulties which can be encountered when conducting test of design support tools in real design situations, rather than controlled experiments. For these reasons the clarity of the results of the study could be improved by a more controlled experiment, for example by randomly selecting half the groups to have access to the full toolset, and the other half only to a design rationale editor and geometry modelling tool without the annotation functionality.

10. Conclusions

The work presented in this thesis investigates the opportunities for, and the consequences of, annotating geometry models with links to design rationale. Furthermore, the work presented in this thesis attempts to fulfil the goals set out by (Lee 1997, Regli, Hu et al. 2000, Bracewell, Ahmed et al. 2004, Chan 2007, McKay, Kundu et al. 2009) of a design rationale system which is integrated in the geometry design tool and allows the capture of an integrated design information space.

The necessity of the integration of design rationale capture tools and methods into the geometry design environment has been discussed frequently in literature. However few practical implementations of such functionality are known to exist, making it impossible to test the hypothesis that such tools would improve design rationale capture during geometry design. The work presented in this thesis is an attempt to establish and implement a practical method to create links from entities in the geometry model to the relevant design rationale and further use these links in the design process.

10.1 Review of objectives

The objectives which were outlined in section 1.3 will be reviewed, and their outcomes briefly summarized:

***Objective 1:** To review the existing literature on design rationale, geometry modelling, computer aided design, geometry annotation and knowledge management to create and understanding of the issues concerning design rationale capture during geometry design.*

Chapters 2 and 3 review the state of the art and find that despite the significant efforts in the geometry annotation community there have been few examples of the integration of design rationale capture functionality into feature-based geometry modellers, and that the creation of such a functionality would enable design rationale to be captured as and when the design geometry is edited.

***Objective 2:** To quantitatively study the capture of design rationale using the current toolset to establish a baseline pattern of design rationale capture using a state of the art toolset during a mechanical engineering design process.*

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

Chapter 4 presents the result of a quantitative study of design rationale capture during geometry design, which shows a low capture of design rationale during the geometry design phase. This confirms the observations in the existing literature of high design rationale capture during concept design, but low design rationale capture during geometry editing.

Objective 3: *Based on the understanding of design rationale capture during the mechanical engineering design process as a result of objectives 1 and 2, to propose new approaches to aid capture and further integrate design rationale into the mechanical engineering design process.*

The inability to link entities in the geometry model to design rationale is identified as a possible cause of the low design rationale capture during geometry design. Chapter 5 describes proposal of a system to enable design rationale capture during geometry design by integrating a linking functionality into a feature based geometry modelling tool, which creates design rationale links by means of attaching attributes to geometry entities.

Objective 4: *To implement and integrate a new tool into the existing industrial toolset, according to the methods and requirements proposed for objective 3.*

Chapter 6, 7 and 8 describe the implementation of three novel tools to integrate design rationale capture into the geometry design process, and to use linked design rationales, together with structure graphs of the geometry model as a means of displaying both the design rationale and the design intent of the design. Analysis of linked design rationale and graphs of the geometry model allows a feedback indicating design rationale presence as well as geometry model status to be displayed as a graphical overlay of the geometry model.

Objective 5: *To evaluate the effectiveness of the new methods and tools during field testing.*

Chapter 9 describes field testing of the proposed toolset. Although use of the toolset correlated with significantly higher design rationale capture during geometry design, reservations have to be made regarding causal nature of this relationship.

10.2 Contributions

This work has presented a method which allows the annotation of ‘feature-based’ geometry models with links to new or existing design rationale maps, as well as further tools and methods to use such annotations during the design process. While considerable work has previously been done on the annotation of geometry models in general, as discussed in section 2.4, and the annotation of geometry models with design rationales in particular, as discussed in section 3.5, the first principal contribution of this work to the field of geometry annotation is a method for the annotation of ‘feature-based’ geometry models with references to a location in an existing set of design rationale maps. This method has been implemented in a prototype tool called RatLink (RATionale Link).

The second principal contribution is a prototype tool called GRapher (Geometry and Rationale Grapher, which displays the structure of a geometry model as a graph, including the operations, dimensions, sketches, equations as mates. It also collects any linked rationale maps based on the annotations made with the RatLink tool, and includes these in the map. The GRapher tool makes the design decisions which are implicit in the geometry model explicitly viewable, by providing a clear and complete overview of the part or assembly as it is modelled, and thereby give users a detailed understanding of parametric and modelling dependencies. As the functional structure of the part should be closely linked to the way in which the geometry model is created, the graph of the structure of the geometry model can also provide additional information on the functional structure of the part as designed. Finally the inclusion of design rationale in the graph allows the existing design rationales to be retrieved and used when changes have to be made to the design or the geometry model, as a result of design modifications or design reuse.

The third principal contribution is a prototype tool called GReAT (Geometry & Rationale Audit Tool). This tool creates a visual overlay on the geometry models which gives information on the presence and quantity of design rationale linked to an area of the model, as well information and quality indicators based on properties and relationships of the geometry model. The visual feedback based on the graphs created by GRapher. The target audience of this tool includes

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

design reviewers and design re-users who need to rapidly understand the parts and the sources of its decisions.

The principal contributions of the work presented in this thesis can be summarized as follows:

1. A quantitative study of design rationale capture during a complete mechanical engineering design process which supports the pre-existing observations that design rationale capture during geometry design is low.
2. The creation of a tool, RatLink, to annotate 'feature-based' geometry models with links to design rationale, allowing design rationale to be captured for precise areas of the geometry.
3. The creation of a novel tool, GRapher, to create combined graphs of the structure of geometry models and the associated design rationale.
4. The creation of a novel tool, GReAT, to display the presence of design rationale as a coloured overlay on the geometry, as well as indicators of the status of the geometry.

10.3 Recommendations for future work

While a causal link cannot be established between linked design rationale capture and team performance and quantity of design rationale captured during geometry design, the correlation between use of the design rationale linking tool and the increased capture of design rationale during the geometry design phase shown in this paper thesis further investigation. A more complete validation of the design rationale integration into the geometry design process could be achieved by an industrial case study evaluating the toolset during a full design process, including a control group which would not have access to design rationale linking tools. An alternative means validating the design tools presented in this thesis would be the use of small controlled experiments as discussed in section 9.1.

As requested by the users, further integration of both design rationale viewing and editing tools in the geometry modeller to allow for a seamless editing and reviewing of the design rationales linked to the geometries would be desirable, although it would create repeated functionality with the existing design rationale editing software.

The combination of a design rationale format which is oriented to making and formalizing design decisions, such as the DRed format, with database stored design rationales which allow multiple users and tools to interact simultaneously

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

with the design rationale, such as Compendium, would allow further audit functionality to be implemented. For example the extension of the visual design rationale feedback tool to display the results of a design rationale audit system as described by Bracewell, Gourtovaia et al. (2009) would give visual feedback on design rationale status in addition to quantity. Such an integration of design rationale audit functions with the visual feedback function discussed in chapter 8 would enable to see how changes in rationale propagate to model geometry: for instance if an issue has an accepted answer which is later rejected the change in design rationale picked up by a design rationale audit can be displayed on the relevant geometry areas.

A Rolls-Royce funded PhD research project has been established to further implement design rationale linking and develop the concepts presented here and apply them in an industrial context.

10.4 Summary

A review of the literature on geometry model annotation indicates that although annotation is often motivated by a desire to explain design decision, there are relatively few attempts to create annotations based on design rationale. The literature on design rationale indicates that despite the large number of tools and methods to capture and use design rationale, the use of design rationale in the mechanical engineering industry is generally low and further indications that in design projects where design rationale is used, design rationale capture during geometry design is low. This thesis presents a study of design rationale capture during the design of a light UAV which confirms the low design rationale capture during geometry design.

The work presented in this thesis is based on the premise that there is a substantial quantity of rich design information present in 'feature-based' geometry models which is not easily accessible to the user, and seeks provide a means to extract this information from the model and present it to the user in a novel way.

To aid design rationale capture and use during geometry design a toolset consisting of three main functionalities is proposed: Firstly a tool named

DESIGN RATIONALE CAPTURE DURING GEOMETRY DESIGN

RatLink, to annotate operations or faces in the geometry model with links to the design rationale. Secondly a tool named GReAT, to display a visual overlay on the geometry model indicating the presence of the design rationale, and the status of the geometry. Thirdly a tool named GRapher, to create graphs of the structure of the geometry model and the design rationale attached to the geometry. The tools presented allow designers and design users to link elements of feature-based geometry models to design rationales, review the input dimensions and design rationales and view an integrated graph of the design rationale and geometry model structure.

A study involving undergraduate engineering students was set up to evaluate the functionality of the prototype toolset during a design project, which seeks to approach real design conditions. The toolset was rolled out to university desktop workstations, was made available to a group of 77 undergraduate aeronautical engineering students performing a group design project of designing a small microlight aircraft in seven groups.

The uptake of the design rationale linking tool was lower than anticipated, with only three out of seven teams making use of the tool. While the usage of the rationale linking tool was arguably low during the design process, its usage correlated with slightly higher design rationale capture overall, and a statistically significant increase of design rationale capture during the geometry design stage. Although several reservations have to be made regarding self-selection bias and a possible inverse causal relationship between design rationale capture and tool use. The groups using the design rationale linking tool had above average grades, and individuals capturing design rationales were well regarded in peer reviews.

Appendices

Appendix 1: Algorithms for geometry model audit

The following C# code is used to determine which features, dimensions, sketches and planes can be qualified as containing either:

- Input dimensions
- Underdefined sketches
- External references
- Design table dimensions

```

/// <summary>
/// searches the graph for features containing input dimensions
/// </summary>
/// <returns></returns>
public Feature[] InputDimensionFeatures(bool primaryDim, bool secondaryDim, bool tertiaryDim)
{
    List<Feature> featureList = new List<Feature>();

    foreach (SWObjVertex vertex in MyObjGraph.Vertices)
    {
        if (vertex as DimVertex != null)
        {
            DimVertex MyVertex = (DimVertex)vertex;

            // this functionality exists already, so we might as well use it.
            if (Convert.ToBoolean(MyVertex.IsInputDim) == true)
            {
                IEnumerable<SWObjEdge> MyOutEdges;
                MyObjGraph.TryGetOutEdges(vertex, out MyOutEdges);

                // check the indegree to see whether this isn't a driven dimension.
                int indegree = MyObjGraph.InDegree(vertex);
                if (indegree == 0)
                {
                    foreach (SWObjEdge e in MyOutEdges)
                    {
                        if ((primaryDim & e.EdgeType == "Primary Dim") | (secondaryDim & e.EdgeType ==
"Secondary Dim")) // should really replace all this with an enumerable.
                        {
                            SWObjVertex target = e.Target;

                            if (target as FeatVertex != null)
                            {
                                FeatVertex myvert = (FeatVertex)target;
                                Feature myFeat = myvert.MyFeature;
                                object specificFeat = myFeat.GetSpecificFeature2();
                                if (specificFeat == null)
                                {
                                    featureList.Add(myFeat);
                                }
                            }
                        }
                    }
                }
                else if (tertiaryDim & (e.EdgeType == "Has Feature"))

```

Appendix 1

```
        {
            SWObjVertex target = e.Target;
            if (target as FeatVertex != null)
            {
                FeatVertex myvert = (FeatVertex)target;
                Feature myFeat = myvert.MyFeature;
                object specificFeat = myFeat.GetSpecificFeature2();
                if (specificFeat == null)
                {
                    featureList.Add(myFeat);
                }
            }
        }
    }
}

return featureList.ToArray();
}

/// <summary>
/// finds features defined based on underdefined sketches
/// </summary>
/// <returns></returns>
public Feature[] UnderdefinedSketchFeatures(bool primaryUnderDef, bool secondaryUnderDef)
{
    List<Feature> featureList = new List<Feature>();
    foreach (SWObjVertex vertex in MyObjGraph.Vertices)
    {
        // is there any point passing through the feature?
        if (vertex as FeatVertex != null)
        {
            FeatVertex myvert = (FeatVertex)vertex;
            Feature myFeat = myvert.MyFeature;
            object specificFeat = myFeat.GetSpecificFeature2();
            if (specificFeat == null) // we have a solid feature...
            {
                IEnumerable<SWObjEdge> MyEdges;

                // because the features have a sketch, rather than the other way round
                MyObjGraph.TryGetOutEdges(vertex, out MyEdges);
                foreach (SWObjEdge E in MyEdges)
                {
                    // now let's do only the direct owners of underdefined sketches.
                    if ((E.EdgeType == "Has Feature" & (E.Target as FeatVertex != null)) &
primaryUnderDef)
                    {
                        FeatVertex featVertex = (FeatVertex)E.Target;
                        if (featVertex.Type == "ProfileFeature") // sketch
                        {
                            if (featVertex.IsFulyDefinedSketch == "false")
                            {
                                featureList.Add(myFeat);
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}

MyObjGraph.TryGetInEdges(vertex, out MyEdges);
foreach (SWObjEdge E in MyEdges)
{
    // the wider search
    if ((E.EdgeType == "Parent of" & (E.Source as FeatVertex) != null) &
secondaryUnderdef)
    {
        FeatVertex featVertex = (FeatVertex)E.Source;

        if (featVertex.Type == "ProfileFeature")
        {
            if (featVertex.IsFulyDefinedSketch == "false")
            {
                featureList.Add(myFeat);
            }
        }
    }
}
}
}
}

return featureList.ToArray();
}

/// <summary>
/// finds features which have external references
/// </summary>
/// <returns></returns>
public Feature[] ExternalrefFeatures()
{
    List<Feature> featureList = new List<Feature>();

    foreach (SWObjVertex vertex in MyObjGraph.Vertices)
    {
        if (vertex as FeatVertex != null)
        {
            FeatVertex myvert = (FeatVertex)vertex;
            Feature myFeat = myvert.MyFeature;

            if (myFeat.ListExternalFileReferencesCount() > 0)
            {
                object specificFeat = myFeat.GetSpecificFeature2();
                // check to see if it's a bodyfeature or surfacefeature and otherwise look for sketches
and such

                if (specificFeat == null)
                {
                    featureList.Add(myFeat);
                }
                else
                {
                    IEnumerable<SWObjEdge> MyEdges;

```

Appendix 1

```
MyObjGraph.TryGetOutEdges(vertex, out MyEdges);

foreach (SWObjEdge E in MyEdges)
{
    if ( E.EdgeType == "Has Feature" & (E.Target as FeatVertex != null))
    {
        if (E.Target.Type == "ProfileFeature") // sketch
        {
            if (E.Source.MetaType == "Feature")
            {
                Feature MyFeature = (Feature)E.Target.MyObject;
                Object SpecificFeature = MyFeature.GetSpecificFeature2();
                // check if it's a solid or surface feature...
                if (SpecificFeature == null)
                {
                    featureList.Add(MyFeature);
                }
            }
        }
    }
}

return featureList.ToArray();
}

/// <summary>
/// finds features which have dimensions driven by design tables
/// </summary>
/// <returns></returns>
public Feature[] DesignTableDimFeatures()
{
    List<Feature> featureList = new List<Feature>();

    foreach (SWObjVertex vertex in MyObjGraph.Vertices)
    {
        if (vertex as FeatVertex != null)
        {
            FeatVertex myvert = (FeatVertex)vertex;
            Feature myFeat = myvert.MyFeature;
            DisplayDimension displayDim = (DisplayDimension)myFeat.GetFirstDisplayDimension();
            // '0' works for everything except chamfers.
            while (displayDim != null)
            {
                Dimension currentDim = displayDim.GetDimension2(0);
                if (currentDim != null)
                {
                    if (currentDim.IsDesignTableDimension() == true)
                    {
                        object specificFeat = myFeat.GetSpecificFeature2();
                        // check to see if it's a bodyfeature or surfacefeature and otherwise look for
                        sketches and such
                        if (specificFeat == null)
                        {
```

```

        featureList.Add(myFeat);
    }
    else
    {
        // ok so if it's a sketch or a plane with external references we should tag
those as well.

        IEnumerable<SWObjEdge> MyEdges;
        MyObjGraph.TryGetOutEdges(vertex, out MyEdges);

        foreach (SWObjEdge E in MyEdges)
        {
            if (E.EdgeType == "Has Feature" & (E.Target as FeatVertex != null))
            {
                if (E.Target.Type == "ProfileFeature") // sketch
                {
                    if (E.Source.MetaType == "Feature")
                    {
                        Feature MyFeature = (Feature)E.Target.MyObject;
                        Object SpecificFeature = MyFeature.GetSpecificFeature2();
                        // check if it's a solid or surface feature...
                        if (SpecificFeature == null)
                        {
                            featureList.Add(MyFeature);
                        }
                    }
                }
            }
        }
    }
}
displayDim = myFeat.GetNextDisplayDimension(displayDim);
if (displayDim != null)
{
    currentDim = displayDim.GetDimension2(0);
}
}
}
return featureList.ToArray();
}
}

```


Appendix 2: Filtering geometry model graphs

The following C# code filters are applied based on user-selected options, during the graphing process to remove nodes from the graph which have little interest to the user. The functions are delegates, which are applied during an iteration over all the nodes in the graph.

```

/// <summary>
/// Removes all items of a certain type from the graph if they don't have children in the
graph.
/// </summary>
/// <param name="MyObjGraph"></param>
/// <returns></returns>
public SWObjGraph RemoveChildless(SWObjGraph MyObjGraph, SWObjVertex vertex, string Type,
string Name)
{
    if (vertex.Type == Type)
    {
        if (MyObjGraph.OutDegree(vertex) == 0)
        {
            MyObjGraph.RemoveVertex(vertex);
        }
    }
    return MyObjGraph;
}

/// <summary>
/// removes all edges of a certain type
/// </summary>
/// <param name="MyObjGraph"></param>
/// <param name="Type"></param>
/// <returns></returns>
public SWObjGraph RemoveEdges(SWObjGraph MyObjGraph, SWObjEdge Edge, string Type)
{
    if (Edge.EdgeType == Type)
    {
        MyObjGraph.RemoveEdge(Edge);
    }

    return MyObjGraph;
}

/// <summary>
/// removes all edges of a type if the source is a feature and the target is a sketch
/// </summary>
/// <param name="MyObjGraph"></param>
/// <param name="Type"></param>
/// <returns></returns>
public SWObjGraph RemoveFeatureToSketchEdge(SWObjGraph MyObjGraph, SWObjEdge Edge, string
Type)
{
    if (Edge.EdgeType == Type)
    {
        if (Edge.Target.Type == "ProfileFeature") // sketch
        {
            if (Edge.Source.MetaType == "Feature")
            {
                MyObjGraph.RemoveEdge(Edge);
            }
        }
    }
    return MyObjGraph;
}

/// <summary>
/// Removes Component vertices which only refer to Models
/// </summary>
/// <param name="MyObjGraph"></param>
/// <returns></returns>
public SWObjGraph RemoveOrphanComp(SWObjGraph MyObjGraph, SWObjVertex vertex)

```

Appendix 3

```
{
    if ((vertex as CompVertex) != null)
    {
        if (MyObjGraph.InDegree(vertex) == 0)
        {
            if (MyObjGraph.OutDegree(vertex) == 1)
            {
                SWObjEdge myedge = MyObjGraph.OutEdge(vertex, 0);
                SWObjVertex myvertex = myedge.GetOtherVertex(vertex);
                if ((myvertex as ModVertex) != null)
                {
                    MyObjGraph.RemoveVertex(vertex);
                }
            }
        }
    }
    return MyObjGraph;
}

/// <summary>
/// Removes vertices which are not connected to any other vertices in the graph
/// </summary>
/// <param name="MyObjGraph"></param>
/// <returns></returns>
public SWObjGraph RemoveOrphanVertex(SWObjGraph MyObjGraph, SWObjVertex vertex)
{
    if (MyObjGraph.OutDegree(vertex) == 0 && MyObjGraph.InDegree(vertex) == 0)
    {
        MyObjGraph.RemoveVertex(vertex);
    }
    return MyObjGraph;
}

/// <summary>
/// Removes features which do not contain any dimensions
/// </summary>
/// <param name="MyObjGraph"></param>
/// <returns></returns>
public SWObjGraph RemoveNonDimFeat(SWObjGraph MyObjGraph, SWObjVertex vertex)
{
    if ((vertex as FeatVertex) != null)
    {
        SWObjVertex OtherVertex = null;
        bool hasdim = false;
        IEnumerable<SWObjEdge> edgecollection = MyObjGraph.InEdges(vertex);
        foreach (SWObjEdge Edge in edgecollection)
        {
            OtherVertex = Edge.GetOtherVertex(vertex);
            if ((OtherVertex as DimVertex) != null)
            {
                hasdim = true;
            }
        }
        if (hasdim == false)
        {
            MyObjGraph.RemoveVertex(vertex);
        }
    }
    return MyObjGraph;
}

/// <summary>
/// Removes nodes with a specific name and type
/// </summary>
/// <param name="MyObjGraph"></param>
/// <returns></returns>
public SWObjGraph RemoveSpecificNode(SWObjGraph MyObjGraph, SWObjVertex vertex, string
Type, string Name )
{
    if (vertex.Label == Name && vertex.Type == Type)
    {
        MyObjGraph.RemoveVertex(vertex);
    }
    return MyObjGraph;
}
}
```

Appendix 3: graphml file for the graph of the structure of a simple geometry model, an engine bracket.

```

<?xml version="1.0" encoding="utf-8"?>
<graphml
xmlns:y="http://www.yworks.com/xml/graphml"
xmlns="http://graphml.graphdrawing.org/xmlns"
  <key for="graphml" id="d0" yfiles.type="resources" />
  <key for="edge" id="d10" yfiles.type="edgegraphics" />
  <key for="node" id="d6" yfiles.type="nodegraphics" />
  <key id="Label" for="node" attr.name="Label" attr.type="string" />
  <key id="Type" for="node" attr.name="Type" attr.type="string" />
  <key id="MetaType" for="node" attr.name="MetaType" attr.type="string" />
  <key id="Group" for="node" attr.name="Group" attr.type="string" />
  <key id="OwnerModel" for="node" attr.name="OwnerModel" attr.type="string" />
  <key id="IsInput" for="node" attr.name="IsInput" attr.type="string" />
  <key id="IsDesignTable" for="node" attr.name="IsDesignTable" attr.type="string" />
  <key id="IsFulyDefinedSketch" for="node" attr.name="IsFulyDefinedSketch"
attr.type="string" />
  <key id="HasExternalReference" for="node" attr.name="HasExternalReference"
attr.type="string" />
  <key id="DimValue" for="node" attr.name="DimValue" attr.type="string" />
  <key id="RationaleGroup" for="node" attr.name="RationaleGroup" attr.type="string" />
  <key id="ID" for="node" attr.name="ID" attr.type="string" />
  <key id="Assembly" for="node" attr.name="Assembly" attr.type="string" />
  <key id="Label" for="edge" attr.name="Label" attr.type="string" />
  <graph id="G" edgedefault="directed" parse.nodes="23" parse.edges="47"
parse.order="nodesfirst" parse.nodeids="free" parse.edgeids="free">
  <node id="Example Bracket V3" yfiles.foldertype="group">
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="Assembly"></data>
    <data key="Group"></data>
    <data key="d6">
      <y:ProxyAutoBoundsNode>
        <y:Realizers active="0">
          <y:GroupNode>
            <y:Geometry height="100" width="100" x="30" y="30" />
            <y:Fill color="#CAECFF84" transparent="false" />
            <y:Shape type="roundrectangle" />
            <y:State Closed="True" innerGraphDisplayEnabled="False" />
            <y:NodeLabel
fontFamily="Dialog" fontSize="15" fontStyle="plain" hasBackgroundColor="#99CCFF"
hasLineColor="false" modelName="internal" modelPosition="t" textColor="#000000"
visible="true" width="38.693359375">Example Bracket V3</y:NodeLabel>
          </y:GroupNode>
          <y:GroupNode>
            <y:Geometry height="150" width="150" x="30" y="30" />
            <y:Fill color="#CAECFF84" transparent="false" />
            <y:Shape type="roundrectangle" />
            <y:State Closed="False" innerGraphDisplayEnabled="True" />
            <y:NodeLabel
fontFamily="Dialog" fontSize="15" fontStyle="plain" hasBackgroundColor="#99CCFF"
hasLineColor="false" modelName="internal" modelPosition="t" textColor="#000000"
visible="true" width="38.693359375">Example Bracket V3</y:NodeLabel>
          </y:GroupNode>
        </y:Realizers>
      </y:ProxyAutoBoundsNode>
    </data>
  <graph edgedefault="directed" id="Example Bracket V3">
    <node id="777771371809600427" yfiles.foldertype="group">
      <data key="OwnerModel">Example Bracket V3</data>
      <data key="Assembly"></data>
      <data key="Group"></data>
      <data key="d6">
        <y:ProxyAutoBoundsNode>
          <y:Realizers active="0">
            <y:GroupNode>
              <y:Geometry height="100" width="100" x="30" y="30" />
              <y:Fill color="#CAECFF84" transparent="false" />
              <y:Shape type="roundrectangle" />
              <y:State Closed="True" innerGraphDisplayEnabled="False" />
              <y:NodeLabel
fontFamily="Dialog" fontSize="15" fontStyle="plain" hasBackgroundColor="#99CCFF"

```

Appendix 3

```
hasLineColor="false"      modelName="internal"      modelPosition="t"      textColor="#000000"
visible="true" width="38.693359375">777771371809600427</y:NodeLabel>
</y:GroupNode>
  <y:GroupNode>
    <y:Geometry height="150" width="150" x="30" y="30" />
    <y:Fill color="#CAECFF84" transparent="false" />
    <y:Shape type="roundrectangle" />
    <y:State Closed="False" innerGraphDisplayEnabled="True" />
    <y:NodeLabel alignment="center" autoSizePolicy="node_width"
fontFamily="Dialog"      fontSize="15"      fontStyle="plain"      hasBackgroundColor="#99CCFF"
hasLineColor="false"      modelName="internal"      modelPosition="t"      textColor="#000000"
visible="true" width="38.693359375">777771371809600427</y:NodeLabel>
  </y:GroupNode>
</y:Realizers>
</y:ProxyAutoBoundsNode>
</data>
<graph edgedefault="directed" id="777771371809600427">
  <node id="21">
    <data key="Label">Fillet sizing rationale - 777771371809600427</data>
    <data key="Type">FeatureRat</data>
    <data key="MetaType">Attribute</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue" />
    <data key="RationaleGroup">777771371809600427</data>
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
      <y:ShapeNode>
        <y:Geometry height="30" width="440" x="30" y="30" />
        <y:Fill color="#FFFAFA" transparent="false" />
        <y:BorderStyle color="#FFCC00" transparent="false" />
        <y:NodeLabel alignment="center" autoSizePolicy="content"
fontFamily="Dialog"      fontSize="14"      fontStyle="plain"      hasBackgroundColor="false"
hasLineColor="false"      modelName="internal"      modelPosition="c"      textColor="#000000"
visible="true">Fillet sizing rationale - 777771371809600427</y:NodeLabel>
        </y:ShapeNode>
      </data>
    </node>
  </graph>
</node>
<node id="21">
  <data key="Label">Fillet sizing rationale - 777771371809600427</data>
  <data key="Type">FeatureRat</data>
  <data key="MetaType">Attribute</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue" />
  <data key="RationaleGroup">777771371809600427</data>
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="440" x="30" y="30" />
      <y:Fill color="#FFFAFA" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14"      fontStyle="plain"      hasBackgroundColor="false"      hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Fillet sizing
rationale - 777771371809600427</y:NodeLabel>
      </y:ShapeNode>
    </data>
  </node>
<node id="0">
  <data key="Label">Front Plane</data>
  <data key="Type">RefPlane</data>
  <data key="MetaType">Feature</data>
```

```

<data key="Group" />
<data key="OwnerModel">Example Bracket V3</data>
<data key="IsInput" />
<data key="IsDesignTable" />
<data key="IsFullyDefinedSketch" />
<data key="HasExternalReference">>false</data>
<data key="DimValue" />
<data key="RationaleGroup" />
<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="120" x="30" y="30" />
    <y:Fill color="#FFFFFF" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Front
Plane</y:NodeLabel>
    <y:Shape type="hexagon" />
  </y:ShapeNode>
</data>
</node>
<node id="1">
  <data key="Label">Origin</data>
  <data key="Type">OriginProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFFF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" visible="true">Origin</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="2">
  <data key="Label">SkeletonSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">>true</data>
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="140" x="30" y="30" />
      <y:Fill color="#DDA4C8" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" visible="true">SkeletonSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>

```

Appendix 3

```
</y:ShapeNode>
</data>
</node>
<node id="3">
  <data key="Label">D1@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">100</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="410" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="4">
  <data key="Label">D2@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">70</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="410" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D2@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="5">
  <data key="Label">D3@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swOrdinateDimension</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">120</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="410" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
```

```

        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D3@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
        <y:Shape type="parallelogram" />
    </y:ShapeNode>
</data>
</node>
<node id="6">
    <data key="Label">CentreHoleSketch</data>
    <data key="Type">ProfileFeature</data>
    <data key="MetaType">Feature</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch">true</data>
    <data key="HasExternalReference">true: 1</data>
    <data key="DimValue" />
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="160" x="30" y="30" />
            <y:Fill color="#8080FF" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">CentreHoleSketch</y:NodeLabel>
            <y:Shape type="hexagon" />
        </y:ShapeNode>
    </data>
</node>
<node id="7">
    <data key="Label">FrontHoleSketch</data>
    <data key="Type">ProfileFeature</data>
    <data key="MetaType">Feature</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch">true</data>
    <data key="HasExternalReference">>false</data>
    <data key="DimValue" />
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="150" x="30" y="30" />
            <y:Fill color="#DDA4C8" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">FrontHoleSketch</y:NodeLabel>
            <y:Shape type="hexagon" />
        </y:ShapeNode>
    </data>
</node>
<node id="8">
    <data key="Label">D1@FrontHoleSketch@Example Bracket V3.Part</data>
    <data key="Type">swDimensionTypeUnknown</data>
    <data key="MetaType">Dimension</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput">true</data>
    <data key="IsDesignTable">True</data>
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue">40</data>
    <data key="RationaleGroup" />

```


Appendix 3

```

<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="420" x="30" y="30" />
    <y:Fill color="#ffc0d6" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@FrontHoleSketch@Example Bracket V3.Part</y:NodeLabel>
    <y:Shape type="parallelogram" />
  </y:ShapeNode>
</data>
</node>
<node id="9">
  <data key="Label">BottomHoleSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">false</data>
  <data key="HasExternalReference">false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="160" x="30" y="30" />
      <y:Fill color="#ff5a5a" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BottomHoleSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="10">
  <data key="Label">BracketOutlineSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">true</data>
  <data key="HasExternalReference">false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="200" x="30" y="30" />
      <y:Fill color="#DDA4C8" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BracketOutlineSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="11">
  <data key="Label">D1@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">true</data>
  <data key="IsDesignTable">False</data>

```

```

<data key="IsFulyDefinedSketch" />
<data key="HasExternalReference" />
<data key="DimValue">20</data>
<data key="RationaleGroup" />
<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="470" x="30" y="30" />
    <y:Fill color="#FFE44E" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
    <y:Shape type="parallelogram" />
  </y:ShapeNode>
</data>
</node>
<node id="12">
  <data key="Label">D2@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">false</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFulyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">10</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="470" x="30" y="30" />
      <y:Fill color="#ABDDA4" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D2@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="13">
  <data key="Label">D3@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFulyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">15</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="470" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D3@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="14">

```

Appendix 3

```

<data key="Label">Bracket</data>
<data key="Type">Extrusion</data>
<data key="MetaType">Feature</data>
<data key="Group" />
<data key="OwnerModel">Example Bracket V3</data>
<data key="IsInput" />
<data key="IsDesignTable" />
<data key="IsFullyDefinedSketch" />
<data key="HasExternalReference">>false</data>
<data key="DimValue" />
<data key="RationaleGroup" />
<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="120" x="30" y="30" />
    <y:Fill color="#FFFFFF" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">Bracket</y:NodeLabel>
    <y:Shape type="hexagon" />
  </y:ShapeNode>
</data>
</node>
<node id="15">
  <data key="Label">D1@Bracket@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">10</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="340" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@Bracket@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="16">
  <data key="Label">BottomHole</data>
  <data key="Type">ICE</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFFF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BottomHole</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>

```

```

    </y:ShapeNode>
  </data>
</node>
<node id="17">
  <data key="Label">CentreHole</data>
  <data key="Type">ICE</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">true: 1</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#8080FF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">CentreHole</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="18">
  <data key="Label">FrontHole</data>
  <data key="Type">ICE</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFBF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">FrontHole</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="19">
  <data key="Label">Fillet1</data>
  <data key="Type">Fillet</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />

```

Appendix 3

```

        <y:Fill color="#FFFFFF" transparent="false" />
        <y:BorderStyle color="#FFCC00" transparent="false" />
        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14"      fontStyle="plain"      hasBackgroundColor="false"      hasLineColor="false"
modelName="internal"      modelPosition="c"      textColor="#000000"
visible="true">Fillet1</y:NodeLabel>
        <y:Shape type="hexagon" />
    </y:ShapeNode>
</data>
</node>
<node id="20">
    <data key="Label">D1@Fillet1@Example Bracket V3.Part</data>
    <data key="Type">swDimensionTypeUnknown</data>
    <data key="MetaType">Dimension</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput">>true</data>
    <data key="IsDesignTable">False</data>
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue">60</data>
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="340" x="30" y="30" />
            <y:Fill color="#FFE44E" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14"      fontStyle="plain"      hasBackgroundColor="false"      hasLineColor="false"
modelName="internal"      modelPosition="c"      textColor="#000000"
visible="true">D1@Fillet1@Example Bracket V3.Part</y:NodeLabel>
            <y:Shape type="parallelogram" />
        </y:ShapeNode>
    </data>
</node>
<node id="21">
    <data key="Label">Fillet sizing rationale - 777771371809600427</data>
    <data key="Type">FeatureRat</data>
    <data key="MetaType">Attribute</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue" />
    <data key="RationaleGroup">777771371809600427</data>
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="440" x="30" y="30" />
            <y:Fill color="#FFFAFA" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14"      fontStyle="plain"      hasBackgroundColor="false"      hasLineColor="false"
modelName="internal"      modelPosition="c"      textColor="#000000"      visible="true">Fillet sizing
rationale - 777771371809600427</y:NodeLabel>
            <y:Shape type="rectangle" />
        </y:ShapeNode>
    </data>
</node>
<node id="22">
    <data key="Label">"D2@BracketOutlineSketch" = "D1@FrontHoleSketch" / 4</data>
    <data key="Type" />
    <data key="MetaType">Equation</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue">0</data>
    <data key="RationaleGroup" />
    <data key="ID" />

```

```

<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="520" x="30" y="30" />
    <y:Fill color="#B5DBEB" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">"D2@BracketOutlineSketch" = "D1@FrontHoleSketch" / 4</y:NodeLabel>
    <y:Shape type="rectangle" />
  </y:ShapeNode>
</data>
</node>
</graph>
</node>
<node id="77771371809600427" yfiles.foldertype="group">
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="Assembly"></data>
  <data key="Group"></data>
  <data key="d6">
    <y:ProxyAutoBoundsNode>
      <y:Realizers active="0">
        <y:GroupNode>
          <y:Geometry height="100" width="100" x="30" y="30" />
          <y:Fill color="#CAEFFF84" transparent="false" />
          <y:Shape type="roundrectangle" />
          <y:State Closed="True" innerGraphDisplayEnabled="False" />
          <y:NodeLabel alignment="center" autoSizePolicy="node_width"
fontFamily="Dialog" fontSize="15" fontStyle="plain" hasBackgroundColor="#99CCFF"
hasLineColor="false" modelName="internal" modelPosition="t" textColor="#000000"
visible="true" width="38.693359375">77771371809600427</y:NodeLabel>
        </y:GroupNode>
        <y:GroupNode>
          <y:Geometry height="150" width="150" x="30" y="30" />
          <y:Fill color="#CAEFFF84" transparent="false" />
          <y:Shape type="roundrectangle" />
          <y:State Closed="False" innerGraphDisplayEnabled="True" />
          <y:NodeLabel alignment="center" autoSizePolicy="node_width"
fontFamily="Dialog" fontSize="15" fontStyle="plain" hasBackgroundColor="#99CCFF"
hasLineColor="false" modelName="internal" modelPosition="t" textColor="#000000"
visible="true" width="38.693359375">77771371809600427</y:NodeLabel>
        </y:GroupNode>
      </y:Realizers>
    </y:ProxyAutoBoundsNode>
  </data>
</graph edgedefault="directed" id="77771371809600427">
  <node id="21">
    <data key="Label">Fillet sizing rationale - 77771371809600427</data>
    <data key="Type">FeatureRat</data>
    <data key="MetaType">Attribute</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference" />
    <data key="DimValue" />
    <data key="RationaleGroup">77771371809600427</data>
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
      <y:ShapeNode>
        <y:Geometry height="30" width="440" x="30" y="30" />
        <y:Fill color="#FFFAFA" transparent="false" />
        <y:BorderStyle color="#FFCC00" transparent="false" />
        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Fillet sizing
rationale - 77771371809600427</y:NodeLabel>
        <y:Shape type="rectangle" />
      </y:ShapeNode>
    </data>
  </node>

```

Appendix 3

```

</graph>
</node>
<node id="0">
  <data key="Label">Front Plane</data>
  <data key="Type">RefPlane</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFBF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Front
Plane</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="1">
  <data key="Label">Origin</data>
  <data key="Type">OriginProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFBF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Origin</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="2">
  <data key="Label">SkeletonSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">>true</data>
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="140" x="30" y="30" />
      <y:Fill color="#DDA4C8" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"

```

```

modelName="internal"                modelPosition="c"                textColor="#000000"
visible="true">SkeletonSketch</y:NodeLabel>
  <y:Shape type="hexagon" />
</y:ShapeNode>
</data>
</node>
<node id="3">
  <data key="Label">D1@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">100</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="410" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal"                modelPosition="c"                textColor="#000000"
visible="true">D1@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="4">
  <data key="Label">D2@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">70</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="410" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal"                modelPosition="c"                textColor="#000000"
visible="true">D2@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="5">
  <data key="Label">D3@SkeletonSketch@Example Bracket V3.Part</data>
  <data key="Type">swOrdinateDimension</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">120</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />

```


Appendix 3

```
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="410" x="30" y="30" />
    <y:Fill color="#FFE44E" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D3@SkeletonSketch@Example Bracket V3.Part</y:NodeLabel>
    <y:Shape type="parallelogram" />
  </y:ShapeNode>
</data>
</node>
<node id="6">
  <data key="Label">CentreHoleSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">true</data>
  <data key="HasExternalReference">true: 1</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="160" x="30" y="30" />
      <y:Fill color="#8080FF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">CentreHoleSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="7">
  <data key="Label">FrontHoleSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">true</data>
  <data key="HasExternalReference">false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="150" x="30" y="30" />
      <y:Fill color="#DDA4C8" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">FrontHoleSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="8">
  <data key="Label">D1@FrontHoleSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">true</data>
  <data key="IsDesignTable">True</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
```

```

<data key="DimValue">40</data>
<data key="RationaleGroup" />
<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="420" x="30" y="30" />
    <y:Fill color="#ffc0d6" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@FrontHoleSketch@Example Bracket V3.Part</y:NodeLabel>
    <y:Shape type="parallelogram" />
  </y:ShapeNode>
</data>
</node>
<node id="9">
  <data key="Label">BottomHoleSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">false</data>
  <data key="HasExternalReference">false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="160" x="30" y="30" />
      <y:Fill color="#ff5a5a" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BottomHoleSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="10">
  <data key="Label">BracketOutlineSketch</data>
  <data key="Type">ProfileFeature</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch">true</data>
  <data key="HasExternalReference">false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="200" x="30" y="30" />
      <y:Fill color="#DDA4C8" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BracketOutlineSketch</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="11">
  <data key="Label">D1@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>

```

Appendix 3

```
<data key="MetaType">Dimension</data>
<data key="Group" />
<data key="OwnerModel">Example Bracket V3</data>
<data key="IsInput">true</data>
<data key="IsDesignTable">False</data>
<data key="IsFullyDefinedSketch" />
<data key="HasExternalReference" />
<data key="DimValue">20</data>
<data key="RationaleGroup" />
<data key="ID" />
<data key="Assembly" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="470" x="30" y="30" />
    <y:Fill color="#FFE44E" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
    <y:Shape type="parallelogram" />
  </y:ShapeNode>
</data>
</node>
<node id="12">
  <data key="Label">D2@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">false</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">10</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="470" x="30" y="30" />
      <y:Fill color="#ABDDA4" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D2@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="13">
  <data key="Label">D3@BracketOutlineSketch@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">15</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="470" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D3@BracketOutlineSketch@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
```

```

</node>
<node id="14">
  <data key="Label">Bracket</data>
  <data key="Type">Extrusion</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFFF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c"
visible="true">Bracket</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="15">
  <data key="Label">D1@Bracket@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">10</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="340" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c"
visible="true">D1@Bracket@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="16">
  <data key="Label">BottomHole</data>
  <data key="Type">ICE</data>
  <data key="MetaType">Feature</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference">>false</data>
  <data key="DimValue" />
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFFF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />

```

Appendix 3

```

        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">BottomHole</y:NodeLabel>
        <y:Shape type="hexagon" />
    </y:ShapeNode>
</data>
</node>
<node id="17">
    <data key="Label">CentreHole</data>
    <data key="Type">ICE</data>
    <data key="MetaType">Feature</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference">true: 1</data>
    <data key="DimValue" />
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="120" x="30" y="30" />
            <y:Fill color="#8080FF" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">CentreHole</y:NodeLabel>
            <y:Shape type="hexagon" />
        </y:ShapeNode>
    </data>
</node>
<node id="18">
    <data key="Label">FrontHole</data>
    <data key="Type">ICE</data>
    <data key="MetaType">Feature</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference">>false</data>
    <data key="DimValue" />
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:ShapeNode>
            <y:Geometry height="30" width="120" x="30" y="30" />
            <y:Fill color="#FFF0FF" transparent="false" />
            <y:BorderStyle color="#FFCC00" transparent="false" />
            <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">FrontHole</y:NodeLabel>
            <y:Shape type="hexagon" />
        </y:ShapeNode>
    </data>
</node>
<node id="19">
    <data key="Label">Fillet1</data>
    <data key="Type">Fillet</data>
    <data key="MetaType">Feature</data>
    <data key="Group" />
    <data key="OwnerModel">Example Bracket V3</data>
    <data key="IsInput" />
    <data key="IsDesignTable" />
    <data key="IsFullyDefinedSketch" />
    <data key="HasExternalReference">>false</data>
    <data key="DimValue" />
    <data key="RationaleGroup" />
    <data key="ID" />
    <data key="Assembly" />
    <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">

```

```

    <y:ShapeNode>
      <y:Geometry height="30" width="120" x="30" y="30" />
      <y:Fill color="#FFFFFF" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">Fillet1</y:NodeLabel>
      <y:Shape type="hexagon" />
    </y:ShapeNode>
  </data>
</node>
<node id="20">
  <data key="Label">D1@Fillet1@Example Bracket V3.Part</data>
  <data key="Type">swDimensionTypeUnknown</data>
  <data key="MetaType">Dimension</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput">>true</data>
  <data key="IsDesignTable">False</data>
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue">60</data>
  <data key="RationaleGroup" />
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="340" x="30" y="30" />
      <y:Fill color="#FFE44E" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D1@Fillet1@Example Bracket V3.Part</y:NodeLabel>
      <y:Shape type="parallelogram" />
    </y:ShapeNode>
  </data>
</node>
<node id="21">
  <data key="Label">Fillet sizing rationale - 777771371809600427</data>
  <data key="Type">FeatureRat</data>
  <data key="MetaType">Attribute</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />
  <data key="IsFullyDefinedSketch" />
  <data key="HasExternalReference" />
  <data key="DimValue" />
  <data key="RationaleGroup">777771371809600427</data>
  <data key="ID" />
  <data key="Assembly" />
  <data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:ShapeNode>
      <y:Geometry height="30" width="440" x="30" y="30" />
      <y:Fill color="#FFFafa" transparent="false" />
      <y:BorderStyle color="#FFCC00" transparent="false" />
      <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000" visible="true">Fillet sizing
rationale - 777771371809600427</y:NodeLabel>
      <y:Shape type="rectangle" />
    </y:ShapeNode>
  </data>
</node>
<node id="22">
  <data key="Label">"D2@BracketOutlineSketch" = "D1@FrontHoleSketch" / 4</data>
  <data key="Type" />
  <data key="MetaType">Equation</data>
  <data key="Group" />
  <data key="OwnerModel">Example Bracket V3</data>
  <data key="IsInput" />
  <data key="IsDesignTable" />

```

Appendix 3

```
<data key="IsFullyDefinedSketch" />
<data key="HasExternalReference" />
<data key="DimValue">0</data>
<data key="RationaleGroup" />
<data key="ID" />
<data key="d6" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:ShapeNode>
    <y:Geometry height="30" width="520" x="30" y="30" />
    <y:Fill color="#B5DBEB" transparent="false" />
    <y:BorderStyle color="#FFCC00" transparent="false" />
    <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="14" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
modelName="internal" modelPosition="c" textColor="#000000"
visible="true">D2@BracketOutlineSketch" = "D1@FrontHoleSketch" / 4</y:NodeLabel>
    <y:Shape type="rectangle" />
  </y:ShapeNode>
</data>
</node>
<edge id="0" source="0" target="1">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="1" source="0" target="2">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="2" source="0" target="6">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="3" source="0" target="7">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="4" source="0" target="9">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="5" source="0" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
```

```

</edge>
<edge id="6" source="0" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="7" source="0" target="16">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="8" source="0" target="17">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="9" source="0" target="18">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="10" source="1" target="2">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="11" source="1" target="6">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="12" source="1" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="13" source="1" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">

```


Appendix 3

```
<y:PolyLineEdge>
  <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
  <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
  <y:Arrows source="none" target="standard" />
</y:PolyLineEdge>
</data>
</edge>
<edge id="14" source="1" target="17">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="15" source="2" target="7">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="16" source="2" target="9">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="17" source="2" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="18" source="2" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="19" source="2" target="16">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="20" source="2" target="18">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
```

```

<edge id="21" source="3" target="2">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="22" source="4" target="2">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="23" source="5" target="2">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="24" source="6" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="25" source="6" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="26" source="6" target="17">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="27" source="7" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="28" source="7" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>

```

Appendix 3

```
<y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
<y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
<y:Arrows source="none" target="standard" />
</y:PolyLineEdge>
</data>
</edge>
<edge id="29" source="7" target="18">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="30" source="8" target="7">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="31" source="8" target="22">
  <data key="Label">Drives</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Drives</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="32" source="9" target="10">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="33" source="9" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="34" source="9" target="16">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="35" source="10" target="14">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="36" source="11" target="10">
```

```

<data key="Label">Primary Dim</data>
<data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
  <y:PolyLineEdge>
    <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
    <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
    <y:Arrows source="none" target="standard" />
  </y:PolyLineEdge>
</data>
</edge>
<edge id="37" source="12" target="10">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="38" source="13" target="10">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="39" source="14" target="16">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="40" source="14" target="17">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="41" source="14" target="18">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="42" source="14" target="19">
  <data key="Label">Parent of</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
      <y:EdgeLabel alignment="center" fontSize="14">Parent of</y:EdgeLabel>
      <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
  </data>
</edge>
<edge id="43" source="15" target="14">
  <data key="Label">Primary Dim</data>
  <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />

```

Appendix 3

```
        <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
        <y:Arrows source="none" target="standard" />
    </y:PolyLineEdge>
</data>
</edge>
<edge id="44" source="19" target="21">
    <data key="Label">Has Attribute</data>
    <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:PolyLineEdge>
            <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
            <y:EdgeLabel alignment="center" fontSize="14">Has Attribute</y:EdgeLabel>
            <y:Arrows source="none" target="standard" />
        </y:PolyLineEdge>
    </data>
</edge>
<edge id="45" source="20" target="19">
    <data key="Label">Primary Dim</data>
    <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:PolyLineEdge>
            <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
            <y:EdgeLabel alignment="center" fontSize="14">Primary Dim</y:EdgeLabel>
            <y:Arrows source="none" target="standard" />
        </y:PolyLineEdge>
    </data>
</edge>
<edge id="46" source="22" target="12">
    <data key="Label">Drives</data>
    <data key="d10" xmlns:y="http://www.yworks.com/xml/graphml">
        <y:PolyLineEdge>
            <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0" />
            <y:EdgeLabel alignment="center" fontSize="14">Drives</y:EdgeLabel>
            <y:Arrows source="none" target="standard" />
        </y:PolyLineEdge>
    </data>
</edge>
</graph>
<data key="d0">
</data>
</graphml>
```

Appendix 4: Statistical significance calculation for test result.

This appendix provides the method and calculation used to determine the statistical significance of the correlation of design rationale tool use with increased design rationale capture during the second half of the design exercise.

H_0 , the null hypothesis, is that the group using the RatLink tool did not capture more design rationale during the geometry design phase. H_1 , the alternative hypothesis, that the groups using the design rationale linking tool, RatLink captured more design rationale during the geometry design phase of the design exercise, which is assumed to coincide with the second half of the design exercise. The selected confidence level for a one-tailed test is $\alpha = 0.025$.

Table 12 Design rationale nodes captured during the second half of the design exercise.

* denotes groups using Ratlink tool.	Bravo	Charlie*	Delta*	Echo	Foxtrot	Gamma*	Hotel
Total design rationale nodes captured	612	594	353	344	748	861	595
Of which captured during the second half	0	357	133	116	21	516	212
Percentage of total nodes	0.00%	60.10%	37.68%	33.72%	2.81%	59.93%	35.63%

Given that in this case there are two groups of unequal size, there are two types of statistical test which could be used: Student's t-test with unequal sample size and equal variance and Welch's t-test (Welch 1947) with unequal sample size and unequal variance. Given that, although both groups originally came from the same population, they have self-selected to use the tool, rather than being assigned to either group, the assumption necessary for Student's t-test that both

Appendix 5

groups have equal variance cannot be retained. For this reason Welch's t-test is more appropriate.

The standard deviation used is the population standard deviation, as the whole population is sampled.

Equation 1 Population standard deviation

$$\sigma_i = \sqrt{\frac{\sum(X_j - \mu_i)^2}{n_i}}$$

Where:

μ_i is the population mean for population i

σ_i is the population standard deviation for population i

X_j is the value of sample j in population i

n_i is the number of samples in population i

Table 13 Standard deviation and mean for groups using or not using the design rationale capture tool

	Groups using design rationale linking tool, RatLink	Groups not using the design rationale linking tool
Sample size, n	3	4
Mean design rationale nodes captured during the second half of the design exercise, \bar{X}_i	335.33	87.25
Population standard deviation, σ	157.11	84.25

Equation 2 Welch's t-test for unequal variances

$$t = \frac{\mu_1 + \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Using Welch's t-test for samples with unequal variances, and the data in Table 13, we obtain a t-value of 2.48.

Equation 3 Welch-Satterwaithe equation giving the effective degrees of freedom ν according to the chi-squared distribution.

$$\nu = \frac{\left(\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\right)^2}{\frac{\left(\frac{\sigma_1^2}{n_1}\right)^2}{(n_1 - 1)} + \frac{\left(\frac{\sigma_2^2}{n_2}\right)^2}{(n_2 - 1)}}$$

When the Welch-Satterwaithe equation is applied to the data in Table 13, the degrees of freedom ν is 25.99, which when rounded to 26 gives the significance levels in Table 14.

Table 14 Critical t-values for a degree of freedom of 26

Two tailed significance level	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.025$
One tailed significance level	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.005$
Critical t-value for DOF = 26	1.31	1.71	2.06	2.78

Therefore the resulting t-value of 2.48 leads to the conclusion that the alternative hypothesis cannot be rejected with a significance level of $\alpha = 0.025$. This means that the groups using the design rationale linking tool captured significantly more design rationale during the second half of the design exercise than the groups not using the design rationale linking tool.

Appendix 5: individual peer review score versus design rationale captured

As discussed in section 9.3.1, a small subset of the total population participating in the experiment could be classed as a being a super-user or technology champion. As a part of the grading process at the end of the design exercise the participants were asked to perform a peer review of the other teammates in their group. These peer review scores were then normalized and used to redistribute marks between team members in an effort to reflect the extent to which each individual had contributed to the design process.

The chart in Figure 54 plots the normalized peer review score against the number of design rationale nodes captured by each individual.

It is interesting to remark that all the super users have been give above average ratings in the peer review scores.

While there is a wide range of both design rationale capture and peer review scores there is a mild correlation between design rationale capture and higher peer review score when the whole population is considered, there is no such correlation when the population without the super users is considered.

Only in the case of the super-users is there a correlation between high design rationale capture and above average peer review scores.

Based on the available date it cannot be determined whether the correlation found between high design rationale capture and above average peer review score implies a causal relation, or whether the super users were generally highly active persons in the design process, making an above average contribution to the team's efforts, but also captured substantial amounts of design rationale. It is however an interesting point which could merit further study.

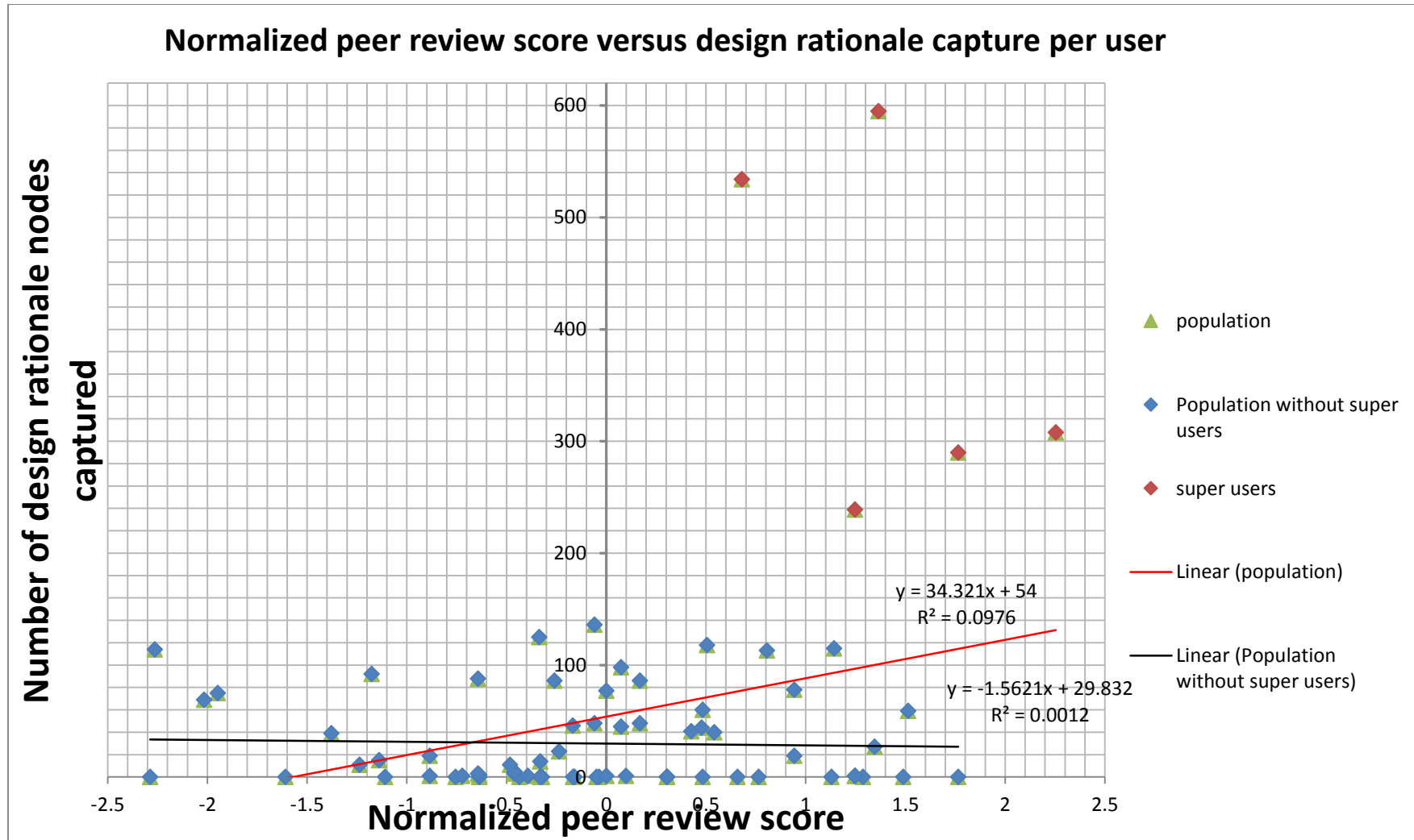


Figure 54 overall design rationale capture versus normalized peer review score for all users

List of References

- Ahmed, S., L. T. M. Blessing and K. M. Wallace (1999). The relationships between data, information and knowledge based on a preliminary study of engineering designers. Proc. ASME Design Theory and Methodology Conference, DETC99/DTM-8754.
- Ahmed, S. and K. M. Wallace (2004). "Identifying and supporting knowledge needs of novice designers within the aerospace industry." Journal of Engineering Design **15**: 475-492.
- Ahmed, S. (2005). "Encouraging reuse of design knowledge: a method to index knowledge." Design Studies **26**(6): 565-592.
- Ahmed, S. and K. Wallace (2006). Reusing design knowledge. Advances in Design, Springer: 75-86.
- Alducin-Quintero, G., M. Contero, J. Martin-Gutiérrez, D. Guerra-Zubiaga and M. Johnson (2011). Productivity Improvement by Using Social-Annotations about Design Intent in CAD Modelling Process. Online Communities and Social Computing, Springer.
- Armstrong, G., R. H. Bracewell and M. Gourtovaia (2012). Rationale Development and Evaluation Tool.
- Aubry, S., I. Thouvenin, D. Lenne and J. Olive (2007). A knowledge model to read 3D annotations on a virtual mock-up for collaborative design. Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on.
- Auricchio, M. and R. H. Bracewell (2009). Engineering Design by Integrated Diagrams. Proceedings of the 17th International Conference on Engineering Design (ICED'09).
- Auricchio, M., R. H. Bracewell and K. M. Wallace (2010). "Understanding how the information requests of aerospace designers influence information seeking behaviour." Journal of Engineering Design **21**: 707-730.
- Auricchio, M., N. L. Eng, J. C. Ortíz, P. R. N. Childs and R. H. Bracewell (2011). On the functions of products. Proceedings of the 18th International Conference on Engineering Design (ICED'11).
- Auricchio, M., R. H. Bracewell and G. Armstrong (2012). The Function Analysis Diagram. Chicago, Illinois, USA: International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. ASME.
- Auricchio, M. and R. Bracewell (2013a). "The Function Analysis Diagram: intended benefits and co-existence with other functional models." Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Special Issue on Functional Descriptions in Engineering **27**(3): 2-2.
- Auricchio, M. and R. H. Bracewell (2013b). "Capturing an integrated design information space with a diagram-based approach." Journal of Engineering Design: 1-32.
- Bai, J., S. Gao, W. Tang, Y. Liu and S. Guo (2010). "Design reuse oriented partial retrieval of CAD models." Computer-Aided Design **42**: 1069-1084.
- Ball, A., L. Ding and M. Patel (2008). "An approach to accessing product data across system and software revisions." Advanced Engineering Informatics **22**(2): 222-235.

Bibliography

- Ball, A., M. Patel, C. A. McMahon, S. Green, P. J. Clarkson and S. Culley (2008). "A grand challenge: immortal information and through-life knowledge management (KIM)." International Journal of Digital Curation **1**: 53-59.
- Ball, L. J., N. J. Lambell, T. C. Ormerod, S. Slavin and J. A. Mariani (2001). "Representing design rationale to support innovative reuse: a minimalist approach." Automation in construction **10**: 633-674.
- Blessing, L. T. M. and A. Chakrabarti (2009). "DRM, a design research methodology."
- Boell, S. (2013, 12 march 2013). "CompendiumNG: Dev community takes the codebase forward." Retrieved 21 October, 2013, from <http://compendiuminstitute.net/news/rostra/news.php@r=55&t=2&id=54.htm>.
- Boujut, J.-F. (2003). "User-defined annotations: artefacts for co-ordination and shared understanding in design teams." Journal of Engineering Design **14**: 409-419.
- Boujut, J.-F. and J. Dugdale (2006). "Design of a 3D annotation tool for supporting evaluation activities in engineering design." Cooperative Systems Design, COOP **6**: 1-8.
- Bracewell, R. H., K. Shea, P. M. Langdon, L. T. M. Blessing and P. J. Clarkson (2001). A methodology for computational design tool research. Proceedings of ICED01, Glasgow, Scotland.
- Bracewell, R. H. and K. M. Wallace (2003). A Tool for Capturing Design Rationale. Proceedings of the 14th International Conference on Engineering Design (ICED03), Stockholm, Sweden.
- Bracewell, R. H., S. Ahmed and K. M. Wallace (2004). Dred and design folders: a way of capturing, storing and passing on, knowledge generated durign design projects. Design and Engineering Technical Conference (DETC '04), Salt lake city, Utah, USA, ASME.
- Bracewell, R. H. and K. M. Wallace (2006). Introducing the capture of argumentation-based design rationale into industrial practise. DCC'06 Design Rationale Workshop Notes.
- Bracewell, R. H., M. Gourtovaia, K. M. Wallace and P. J. Clarkson (2007). Extending Design Rationale to Capture An Integrated Design Information Space. Proceedings of the 16th International Conference on Engineering Design (ICED07), Paris, France.
- Bracewell, R. H., M. Gourtovaia, M. Moss, D. Knott, K. M. Wallace and P. J. Clarkson (2009). DRED 2.0: A Method and Tool for Capture and Communication of Design Knowledge Deliberated in the Creation of Technical Products. Proceedings of the 17th International Conference on Engineering Design (ICED'09), Stanford, CA, USA.
- Bracewell, R. H., K. Wallace, M. Moss and D. Knott (2009). "Capturing design rationale." Computer-Aided Design **41**: 173-186.
- Brandes, U., M. Eiglsperger, I. Herman, M. Himsolt and M. S. Marshall (2002). GraphML progress report structural layer proposal. Graph Drawing, Springer.
- Brandes, U., M. Eiglsperger, J. Lerner and C. Pich (2004). "Graph markup language (GraphML)." Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications). Chapman & Hall/CRC.
- Bras, B. A. and F. Mistree (1991). "Designing Design Pocesesses in Decision-Based Concurrent Engineering." SAE Transactions, Journal of Materials and Manufacturing **100**: 451-458.

List of References

- Brunsmann, J. and W. Wilkes (2009). "Enabling product design reuse by long-term preservation of engineering knowledge." International Journal of Digital Curation **4**: 17-28.
- Buckingham Shum, S. J. (1996). Design argumentation as design rationale. The encyclopedia of computer science and technology. NY, Marcel Dekker. **35**: 95-128.
- Buckingham Shum, S. J., A. M. Selvin, M. Sierhuis, J. Conklin, C. Haley and B. Nuseibeh (2006). Hypermedia support for argumentation based rationale: 15 years from gIBIS and QOC. Rationale management in software engineering. A. Dutoit, R. McCall, I. Mistrik and B. Paech, Springer: 111-132.
- Burge, J. E. and D. C. Brown (2004). An integrated approach for software design checking using design rationale. 1st International Conference on Design Computing and Cognition (DCC'04).
- Burge, J. E. (2005). Software Engineering Using design RATIONale. PhD. PhD thesis, Worcester Polytechnic Institute.
- Burge, J. E. and D. C. Brown (2008). "Software Engineering Using RATIONale." Journal of Systems and Software **81**: 395-413.
- Burgess Yakemovic, K. C. and J. E. Conklin (1990). Report on a development project use of an issue-based information system. Proceedings of the 1990 ACM conference on Computer-supported cooperative work.
- Busby, J. S. (1999). "The Problem with Design Reuse: An Investigation into Outcomes and Antecedents." Journal of Engineering Design **10**: 277-296.
- Buur, J. and M. Myrup Andreasen (1989). "Design models in mechatronic product development." Design Studies **10**(3): 155-162.
- Capilla, R. (2009). Embedded design rationale in software architecture. Joint Working IEEE/IFIP Conference on Software Architecture, 2009 European Conference on Software Architecture. WICSA/ECSA 2009.
- Chan, C.-C. (2007). Capturing design rationale within a CAD environment. PhD. PhD thesis, Hong Kong Polytechnic University.
- Clarkson, P. J. and J. R. Hamilton (2000). "'Signposting', a parameter-driven task-based model of the design process." Research in Engineering Design **12**: 18-38.
- Clarkson, P. J., C. Simons and C. Eckert (2004). "Predicting change propagation in complex design." Journal of Mechanical Design **126**: 788.
- Conklin, J. E. and M. L. Begeman (1987). gIBIS: A hypertext tool for team design deliberation. Proceedings of the ACM conference on Hypertext.
- Conklin, J. E. and M. L. Begeman (1988). "gIBIS: A hypertext tool for exploratory policy discussion." ACM Transactions on Information Systems (TOIS) **6**: 303-331.
- Conklin, J. E. and K. C. Burgess Yakemovic (1991). "A process-oriented approach to design rationale." Human-Computer Interaction **6**: 357-391.
- Conklin, J. E., A. Selvin, S. J. Buckingham Shum and M. Sierhuis (2001). Facilitated hypertext for collective sensemaking: 15 years on from gIBIS. Proceedings of the 12th ACM conference on Hypertext and Hypermedia, New York, NY, USA, ACM.

Bibliography

Conway, A. P. and W. J. Ion (2013). "Enhancing the design dialogue: an architecture to document engineering design activities." Journal of Engineering Design **24**: 140-164.

Davies, D. (2008). Representation of multiple engineering viewpoints in Computer Aided Design through computer-interpretable descriptive markup Doctor of Philosophy (PhD), University of Bath.

Demian, P. and R. Fruchter (2006). "An ethnographic study of design knowledge reuse in the architecture, engineering, and construction industry." Research in Engineering Design **16**(4): 184-195.

Demian, P. and R. Fruchter (2009). "Effective visualization of design versions: visual storytelling for design reuse." Research in Engineering Design **4**: 193-204.

Ding, L., J. Matthews, C. A. McMahon and G. Mullineux (2007). An Extended Product Model for Constraint-Based Redesign Applications. Proceedings of the 16th International Conference on Engineering Design (ICED07).

Ding, L., D. Davies and C. A. McMahon (2009). "The integration of lightweight representation and annotation for collaborative design representation." Research in Engineering Design **20**: 185-200.

Ding, L., J. Matthews, C. A. McMahon and G. Mullineux (2009). "An information support approach for machine design & building companies." Concurrent Engineering **17**: 103-109.

Ding, L. and S. Liu (2010). "Markup in Engineering Design: A Discourse." Future Internet **2**: 74-95.

Ding, L., A. Ball, M. Patel and G. Mullineux (2011). "Lightweight product lifecycle information management for small enterprises." International Journal of Product Lifecycle Management **5**: 21-36.

Diwakaran, R. P. and M. D. Johnson (2012). "Analyzing the effect of alternative goals and model attributes on CAD model creation and alteration." Computer-Aided Design **44**: 343-353.

Du, J., S. Jing and J. Liu (2012). "Creating shared design thinking process for collaborative design." Journal of Network and Computer Applications **35**: 111-120.

Eckert, C., M. Stacey and P. Clarkson (2004). The Lure of the Measurable in Design Research. Proceedings of Design 2004, 8th International Design Conference.

Eng, N., M. Aurisicchio, R. Bracewell and G. Armstrong (2012). Mapping for Design Decision Support in Industry. Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2012 Chicago, IL, USA.

Eng, N. L., R. H. Bracewell and P. J. Clarkson (2009). Concept diagramming software for engineering design support: a review and synthesis of studies. ASME 2009 International design technical conference & computers and information in engineering conference, San Diego, CA, USA, ASME.

Eng, N. L., E. Marfisi and M. Aurisicchio (2011). Adapting aerospace design rationale mapping to civil engineering: a preliminary study. Proceedings of the 18th International Conference on Engineering Design (ICED'11).

List of References

- Ferraro, M., D. Gorissen, J. P. Scanlan, K. A. J., E. Quaranta, B. Schumann, J. van Schaik and M. i Gisbert (2012). Toward Value-Driven Design of a Small, Low-Cost UAV. Proceedings of 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference.
- Fischer, G., A. C. Lemke, R. McCall and A. I. Morch (1991). "Making argumentation serve design." Human-Computer Interaction **6**: 393-419.
- Gorissen, D. (2011, 2011). "compendiumstats." Retrieved 6 september 2013, from <https://github.com/dgorissen/compendiumstats>.
- Gorissen, D., E. Quaranta, M. Ferraro, K. Takeda, S. J. P. and A. J. Keane (2012). Architecting a Decision Environment for Complex Design Evaluation. Proceedings of 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Github.
- Gorissen, D., E. Quaranta, M. Ferraro, B. Schumann, J. van Schaik, A. Keane and J. Scanlan (2013). "A Decision Environment for Complex Design Evaluation." AIAA Journal. Under review.
- Hisarciklilar, O. and J.-F. Boujut (2007). An annotation based approach to support design communication. Proceedings of the 16th International Conference on Engineering Design (ICED07).
- Hisarciklilar, O. (2008). Formes et structures des annotations sémantiques pour supporter la communication en conception collaborative asynchrone. PhD, Institut National Polytechnique de Grenoble-INPG.
- Hisarciklilar, O. and J.-F. Boujut (2009). "An annotation model to reduce ambiguity in design communication." Research in Engineering Design **20**: 171-184.
- Hisarciklilar, O., R. Rasoulifar, J.-F. Boujut, G. Thomann and F. Villeneuve (2009). User-Designer Collaboration in the Design Process of Surgical Instruments: New Aspects for Annotation as a Communication Tool. Proceedings of the 17th International Conference on Engineering Design (ICED'09).
- Hooey, B. L. and D. C. Foyle (2007). Requirements for a Design Rationale Capture tool to support NASA's Complex systems. International workshop on managing knowledge for space missions, Pasadena, CA, USA.
- Huet, G., S. J. Culley, C. A. McMahon and C. Fortin (2007). "Making sense of engineering design review activities." Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM **21**: 243-266.
- Huet, G., C. A. McMahon, F. Sellini, S. J. Culley and C. Fortin (2007). "Knowledge loss in design reviews." Advances in Integrated Design and Manufacturing in Mechanical Engineering II -: 277-291.
- Huet, G., H. McAlpine, R. Camarero, S. J. Culley, T. Leblanc and C. Fortin (2009). The Management of Digital Sketches through PLM solutions. Proceedings of the 17th International Conference on Engineering Design (ICED'09).
- ISO (1994). "10303: 1994-Industrial Automation Systems and Integration-Product Data Representation and Exchange." The ISO web site is at <http://www.iso.ch/cate/cat.html-search> on 10303.

Bibliography

- Iyer, G. R. and J. J. Mills (2006). "Design Intent in 2D CAD: Definition and Survey." Computer-Aided Design and Applications **3**: 259-267.
- Johnson, M. D. and R. P. Diwakaran (2011). "An educational exercise examining the role of model attributes on the creation and alteration of CAD models." Computers & Education **57**(2): 1749-1761.
- Kankuzi, B. and Y. Ayalew (2008). An end-user oriented graph-based visualization for spreadsheets. Proceedings of the 4th international workshop on End-user software engineering, New York, NY, USA, ACM.
- Keller, R., C. M. Eckert and P. J. Clarkson (2009). "Using an engineering change methodology to support conceptual design." Journal of Engineering Design **20**: 571-587.
- Keraron, Y., A. Bernard and B. Bachimont (2009). "Annotations to improve the using and the updating of digital technical publications." Research in Engineering Design **20**: 157-170.
- Kim, J., M. J. Pratt, R. G. Iyer and R. D. Sriram (2008). "Standardized data exchange of CAD models with design intent." Computer-Aided Design **40**: 760-777.
- Kim, S., R. H. Bracewell and K. M. Wallace (2007). Improving design reuse using context. Proceedings of the 16th International Conference on Engineering Design (ICED07).
- Klein, M. (1993). "Capturing design rationale in concurrent engineering teams." Computer **26**(1): 39-47.
- Klein, M. (1997). Capturing geometry rationale for collaborative design. Enabling Technologies: Infrastructure for Collaborative Enterprises, 1997., Proceedings Sixth IEEE workshops on, IEEE.
- Koh, E. C. Y., N. H. M. Caldwell and P. J. Clarkson (2013). "A technique to assess the changeability of complex engineering systems." Journal of Engineering Design **24**(7): 477-498.
- Kunz, W. and H. W. J. Rittel (1970). Issues as elements of information systems. Berkeley, Center for planning and development research.
- Kurland, R. (2008) "Solid Edge with synchronous technology changes solid modeling."
- Lee, H., J. Kim and A. Banerjee (2010). "Collaborative intelligent CAD framework incorporating design history tracking algorithm." Computer-Aided Design **42**: 1125-1142.
- Lee, J. (1990). SIBYL: a tool for managing group design rationale. Proceedings of the 1990 ACM conference on Computer-supported cooperative work, New York, NY, USA, ACM.
- Lee, J. and K.-Y. Lai (1991). "What's in design rationale?" Human-Computer Interaction **6**(3-4): 251-280.
- Lee, J. and K.-Y. Lai (1992). A Comparative Analysis of Design Rationale Representations.
- Lee, J. (1997). "Design rationale systems: Understanding the Issues." IEEE intelligent systems **12**: 78-84.
- Lenne, D., I. Thouvenin and S. Aubry (2009). "Supporting design with 3D-annotations in a collaborative virtual environment." Research in Engineering Design **20**: 149-155.

List of References

- Li, C., C. McMahon and L. Newnes (2011). Progress with OntoCAD: A Standardised Ontological Annotation Approach to CAD Systems. International Conference on Product Lifecycle Management (PLM 2011), Eindhoven, Netherlands.
- Li, C. (2012). Ontology-Driven Semantic Annotations for Multiple Engineering Viewpoints in Computer Aided Design. PhD thesis, University of Bath.
- Li, M., Y. F. Zhang and J. Y. H. Fuh (2010). "Retrieving reusable 3D CAD models using knowledge-driven dependency graph partitioning." Computer-Aided Design and Applications **7**: 417-430.
- Lowe, A., C. A. McMahon and S. Culley (2004). "Characterising the requirements of engineering information systems." International Journal of Information Management **24**: 401-422.
- Lundin, M., S. Sandberg and M. Näsström (2010). Knowledge retention and reuse: using CAD models as carriers of knowledge in product development. ASME international design engineering technical conferences. August.
- Ma, L., Z. Huang and Y. Wang (2009). Common design structure discovery from CAD models. Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09. 11th IEEE International Conference on.
- MacLean, A. and D. McKerlie (1995). "Design space analysis and use-representations." Scenario-based design: envisioning work and technology in system development: 183-207.
- MacLean, A. A., R. M. Young and T. P. Moran (1989a). Design rationale: the argument behind the artifact. Proceedings of the SIGCHI conference on Human factors in computing systems: Wings for the mind. New York, NY, USA: 247-252.
- MacLean, A. A., R. M. Young and T. P. Moran (1989b). Design rationale: the argument behind the artifact. Proceedings of the SIGCHI conference on Human factors in computing systems: Wings for the mind, New York, NY, USA.
- MacLean, A. A., R. M. Young, V. M. E. Bellotti and T. P. Moran (1991). "Questions, options, and criteria: Elements of design space analysis." Human-Computer Interaction **6**: 201-250.
- MacLean, A. A., V. M. Bellotti and S. J. Buckingham Shum (1993). Developing the design space with design space analysis. Computers, Communication and Usability: Design issues, research and methods for integrated services. P. F. Byerly, P. J. Barnard and J. May. Amsterdam, Elsevier: 197-219.
- Magnuson, S. (2012). "After 50 Years, There Is No End in Sight for the Chinook." National Defense Magazine **April**: -.
- Maher, K. (2013) "Does the CAD world need another geometry kernel?" GraphicSpeak.
- Marples, D. L. (1961). "The Decisions of Engineering Design." Engineering Management, IRE Transactions on **EM-8**: 55-71.
- McKay, A., S. Kundu, A. de Pennington and P. G. Dawson (2009). An Integrated Product, Process and Rationale Model for the Provision of Through-Life Information in Product Service Systems. Proceedings of the 17th International Conference on Engineering Design (ICED'09).
- McMahon, C. A., A. Lowe and S. Culley (2004). "Knowledge management in engineering design: personalization and codification." Journal of Engineering Design **15**: 307-325.

Bibliography

- McMahon, C. A., M. Giess and S. Culley (2005). "Information management for through life product support: the curation of digital engineering data." International Journal of Product Lifecycle Management **1**: 26-42.
- Mix, K. J., G. C. Jensen and J. Ryskamp (2010). "Automated Design Rationale Capture within the CAx Environment." Computer-Aided Design and Applications **7**: 361-375.
- Myers, K. L., N. B. Zumel and P. Garcia (2000). "Acquiring Design rationale automatically." Artificial intelligence for engineering design **14**: 1-45.
- Nagel, R. N., W. W. Braithwaite and P. R. Kennicott (1980). Initial graphics exchange specification IGES, version 1.0, National Bureau of Standards.
- Ovsiannikov, I. A., M. A. Arbib and T. H. McNeill (1999). "Annotation technology." International Journal of Human-Computer Studies **50**: 329-362.
- Pahl, G., W. Beitz, H. J. Schulz and U. Jarecki (2007). "Engineering design: a systematic approach."
- Parnas, D. L. and D. M. Weiss (1985). Active design reviews: principles and practices.
- Parnas, D. L. and D. M. Weiss (1987). "Active design reviews: principles and practices." Journal of Systems and Software **7**(4): 259-265.
- Patel, M., A. Ball and L. Ding (2009). "Strategies for the curation of CAD engineering models." International Journal of Digital Curation **4**: 84-97.
- Pavkovic, N., N. Bojetic, I. Vadla and D. Rohde (2010). Embedding Design Rationale Capturing in PLM Systems - a case study with IBIS-based diagrams. Proceedings of the 11th International Design Conference DESIGN 2010.
- Potts, C. and G. Bruns (1988). Recording the reasons for design decisions. Proceedings of the 10th international conference on Software engineering, Los Alamitos, CA, USA, IEEE Computer Society Press.
- Pratt, M. J., B. D. Anderson and T. Ranger (2005). "Towards the standardized exchange of parameterized feature-based CAD models." Computer-Aided Design **37**(12): 1251-1265.
- Pugh, S. (1990). Total Design, Pearons Education.
- Quickgraph. (Serialization). "Serialization." Retrieved 11-10-2013, 2013, from <http://quickgraph.codeplex.com/SourceControl/latest#3.0/sources/QuickGraph.Serialization/GraphQLSerializer.cs>.
- Reeves, B. and F. Shipman (1992). Making it Easy for Designers to Provide Design Rationale. Working Notes, AAAI-92 Workshop on Design Rationale Capture and Use.
- Regli, W. C., X. Hu, M. Atwood and W. Sun (2000). "A survey of design rationale systems: approaches, representation, capture and retrieval." Engineering with computers **16**: 209-235.
- Ritchie, J. M., R. C. Sung, H. Rea, T. Lim, J. R. Corney and I. Howley (2008). The use of non-intrusive user logging to capture engineering rationale, knowledge and intent during the product life cycle. Management of Engineering & Technology, 2008. PICMET 2008. Portland International Conference on.

List of References

- Rittel, H. W. and M. M. Webber (1973). "Dilemmas in a General Theory of Planning." Policy Sciences **4**: 155-169.
- Robertson, B. F. and D. F. Radcliffe (2009). "Impact of CAD tools on creative problem solving in engineering design." Computer-Aided Design **41**: 136-146.
- Robinson, G., J. M. Ritchie, P. N. Day and R. G. Dewar (2007). "System design and user evaluation of Co-Star: An immersive stereoscopic system for cable harness design." Computer-Aided Design **39**(4): 245-257.
- Roldan, M. L., S. Gonnet and H. Leone (2010). "TracED: a tool for capturing and tracing engineering design processes." Advances in Engineering Software **41**: 1087-1109.
- Salehi-Douzloo, V. (2012). An Integrated Approach to Parametric Associative Design for Powertrain Components on the Automotive Industry. PhD thesis, University of Bath.
- Salehi, V. and C. A. McMahon (2009). Action research into the use of parametric associative CAD systems in an industrial context. Proceedings of the 17th International Conference on Engineering Design (ICED'09).
- Salehi, V. and C. A. McMahon (2011). "Development and Application of an Integrated Approach for Parametric Associative CAD Design in an Industrial Context." Computer-Aided Design and Applications **8**: 225-236.
- Salustri, F. A., J. S. Weerasinghe, R. H. Bracewell and N. L. Eng (2007). "Visualising early engineering design information with diagrams." Journal of Design Research **6**: 190-217.
- Sandberg, S. and M. Näsström (2007). A proposed method to preserve knowledge and information by use of knowledge enabled engineering. ASME international design engineering technical conference. September.
- Sandberg, S., M. Lundin, M. Näsström, L.-E. Lindgren and D. Berglund (2013). "Supporting engineering decisions through contextual, model-oriented communication and knowledge-based engineering in simulation-driven product development: an automotive case study." Journal of Engineering Design **24**: 45-63.
- Schumann, B., J. P. Scanlan and K. Takeda (2011). A Generic Operational Simulation for Early Design Civil Unmanned Aerial Vehicles. SIMUL2011: The Third International Conference on Advances in System Simulation, Barcelona, Spain, IARIA.
- Selvin, A., S. Buckingham Shum, M. Seirhuis, J. Conklin, B. Zimmerman, C. Palus, W. Drath, D. Horth, J. Domingue and E. Motta (2001). Compendium: Making meetings into knowledge events. Knowledge Technologies 2001, Texas, USA.
- Shah, J. J. (1991). "Assessment of features technology." Computer-Aided Design **23**(5): 331-343.
- Shah, J. J. and M. T. Rogers (1993). "Assembly modeling as an extension of feature-based design." Research in Engineering Design **5**(3-4): 218-237.
- Shahin, M., P. Liang and M. R. Khayyambashi (2010). Rationale visualization of software architectural design decision using compendium. Proceedings of the 2010 ACM Symposium on Applied Computing, ACM.
- Siemens Product Lifecycle Management Software Inc (2008). Parasolid XT Format Reference.

Bibliography

- Snyder, C. R., C. A. Snyder and C. S. Sankar (1998). "Use of information technologies in the process of building the boeing 777." Journal of Information Technology Management **9**: 31-42.
- Sóbestor, A. and A. J. Keane (2006). Multidisciplinary design optimization of uav airframes. 47th AIAA Structures, Structural Dynamics and Material Conference.
- SpaceClaim Corporation. (2013). "Why Spaceclaim?" Retrieved 17 October 2013, from <http://www.spaceclaim.com/en/default.aspx>.
- Sung, R. C., J. M. Ritchie, G. Robinson, P. N. Day, J. R. Corney and T. Lim (2009). "Automated design process modelling and analysis using immersive virtual reality." Computer-Aided Design **41**: 1082-1094.
- Sung, R. C., J. M. Ritchie, H. J. Rea and J. Corney (2011). "Automated design knowledge capture and representation in single-user CAD environments." Journal of Engineering Design **22**: 487-503.
- Surendra, A., M. Ferraro, B. Schumann, J. van Schaik, J. J. Daniels, D. Gorissen, S. J. P. and A. J. Keane (2012). "The Challenges of using Value-Driven Design for practical design of UAVs." Journal of Aerospace Operations **Volume 1**: 377-386.
- Szykman, S., R. D. Sriram and W. C. Regli (2001). "The role of knowledge in next-generation product development systems." Journal of Computing and Information Science in Engineering **1**: 3-11.
- Toulmin, S. E. (1958). the Uses of Argument, Cambridge University Press.
- Ullman, D. G. (2002). "Toward the ideal mechanical engineering design support system." Research in Engineering Design **13**: 55-64.
- Urbanic, R. J. and W. ElMaraghy (2009). "A design recovery framework for mechanical components." Journal of Engineering Design **20**: 195-215.
- Uva, A. E., S. Cristiano, M. Fiorentino and G. Monno (2010). "Distributed design review using tangible augmented technical drawings." Computer-Aided Design **42**: 364-372.
- van Schaik, J., J. P. Scanlan, A. J. Keane, K. Takeda and D. Gorissen (2011). Improving design rationale capture during embodiment design. Proceedings of the 18th International Conference on Engineering Design (ICED'11).
- Welch, B. L. (1947). "The generalization of student's' problem when several different population variances are involved." Biometrika **34**(1/2): 28-35.
- Zdrahal, Z., P. Mullholland, M. Valasek and A. Bernardi (2007). "Worlds and transformations: supporting the sharing and reuse of design knowledge." International journal of human computer studies **65**: 953-982.
- Zhang, Y. and X. Luo (2009). Design Intent Information Exchange of Feature-Based CAD Models. Computer Science and Information Engineering, 2009 WRI World Congress on.