BRIDGING THE KNOWLEDGE GAP BETWEEN DESIGN, MANUFACTURE AND MEASUREMENT IN THE FIELD OF SURFACE TEXTURE

By

QUNFEN QI

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computing & Engineering

The University of Huddersfield

November 2013

Abstract

Surface texture, a core part of geometrical product specifications and verification (GPS), is embraced by the whole surface manufacture chain from design through manufacture and measurement, and plays a significant role in determining the functional performance of workpieces. The delivery and implementation of surface texture knowledge in GPS, however, is undergoing critical problems in current practice. Surface specification/design systems lag far behind the measurement systems. This is caused by knowledge gaps between design, manufacture and measurement in surface texture exemplifying the necessity of an infrastructure which synergy seamlessly between different stages.

This thesis documents the development of a surface texture knowledge platform called CatSurf to bridge the knowledge gaps. A category theory based knowledge modelling methodology is proposed to underpin the mathematical foundation of the CatSurf. Deploying this methodology, the knowledge modelling for areal and profile surface texture is carried out. The design and implementation of the CatSurf system is developed based on modelling. In addition, the CatSurf system is integrated with Computer Aided Design systems by utilising a Component Object Model (COM) and XML (Extensible Markup Language) based integration methodology.

The integrated CatSurf system provides unambiguous surface texture information for designers and metrologists, and enables metrology assisted design and manufacture to become reality. Currently, it is an executable system with three different modules which can be integrated with CAD systems such as AutoCAD and SolidWorks. A special module is developed for Rolls Royce with a single roughness parameter *Ra* for gas washed surfaces. The system is tested and recognised by various parties such as Rolls Royce, CAx and GPS experts, computing and mechanical engineers and researchers, etc.

Related Publications

1. Qi, Q., Jiang, X. & Scott, P.J. (2013) A Knowledge Platform for Surface Texture in Advanced Manufacturing. *2nd Annual EPSRC Manufacture The Future Conference*, 17th-18th, Sep. 2013, Cranfield, UK
2. Qi, Q., Jiang, X., Scott, P.J. & Lu, W. (2013) Surface texture specification, the more complete the better? *Procedia CIRP*, *10*, 233-237
3. Qi, Q., Scott, P.J., Jiang, X. & Lu, W. (2013) The discussion of uncertainties in surface texture during design, manufacture and measurement. *Paper presented at the 11th international Symposium of Measurement Technology and Intelligent Instruments (ISMTII)*, Aachen, Germany
4. Qi, Q., Scott, P.J., Jiang, X. & Lu, W. (2013) Design and implementation of an integrated surface texture information system for design, manufacture and measurement. Submitted to *Computer - Aided Design* (CAD-D-13-00296, under review).
5. Qi, Q., Scott, P.J., Jiang, X., Lu, W. & Ding, H. (2013) The management of uncertainties in surface texture during design, manufacture and measurement. submitted to *Surface Topography: Metrology and Properties*, article ID: STMP/484610/SPE, under review (This paper was invited to submit to STMP based on ISMTII conference paper)
6. Qi, Q., Jiang, X., Scott, P.J. & Lu, W. (2012) Model for a CAD-assisted designing process with focus on the definition of the surface texture specifications and quality. *Proceedings of the 37th International MATADOR Conference*, Springer, 273-276
7. Qi, Q., Jiang, X. & Scott, P.J. (2012) Knowledge modeling for specifications and verification in areal surface texture. *Precision Engineering, 36*, 322-333
   (This paper is also selected by Engineering Community, and now featured online on Advances in Engineering)
8. Qi, Q., Jiang, X., Liu, X. & Scott, P.J. (2010) An unambiguous expression method of the surface texture. *Measurement*, *43(10)*, 1398-403

9. Qi, Q., Jiang, X., Blunt, L. & Scott, P.J. (2010) Modeling of the concepts in ISO standards for PST. *Paper presented at the Future Technologies in Computing and Engineering: Proceedings of Computing and Engineering Annual Researchers' Conference 2010*: CEARC'10. University of Huddersfield, Huddersfield, UK.172-176
10. Lu, W., Liu, X., Jiang, X., Qi, Q. & Scott, P.J (2010) Categorical data model for cylindricity consistent with geometrical product specifications standard system. *Proc. of SPIE*, *v7544*, DOI:10.1117/12.885276
11. Lu, W., Jiang, X., Qi, Q. & Scott, P.J. (2010) Modeling the integration between specifications and verification for cylindricity based on category theory. *Measurement Science and Technology*, *21*, 115107

Acknowledgements

First and foremost I offer my sincere gratitude to my supervisor, Professor Xiangqian (Jane) Jiang who conduct this research work to success, for recognising me as her research student and for leading me in the way of surface metrology. It is my great privilege to have such dedicated guide and mentor who provides dynamic encouragement and all the guidance to me over the time of my academic years. There is no doubt that without her efforts the task would not be achieved.

I would like to devote my profound gratitude to Professor Paul J Scott who has helped fine-tune many of the ideas expressed in this thesis, and has been a valuable source of encouragement. His positive outlook and confidence in my research inspired me and gave me confidence. I have been extremely lucky to have a co-supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I would also like to thank Professor Liam Blunt for his support throughout the duration of this project. Thanks are due to Dr. Paul Bills for proof reading this thesis and his help and advice over years. I would also like to thank Dr. Feng Gao, Dr. Tunkun Li and Dr. Wenhan Zeng for their advice and help over years. The past and present members of Centre for Precision Technology (CPT), the new EPSRC Centre for Innovative Manufacturing in Advanced Metrology, have also been a great help, a family, and their friendship has enhanced the quality of my working life and hence created an atmosphere conducive to research.

My immense gratitude also goes out to my family (mum, dad, Hui, Xia, Jie, Jun, Xuan, Yong and Yi), who may not always have understood but who have always been understanding, for their continued support and encouragement.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AFM	Atomic force microscopy
APA	Any Process Allowed
API	Application Programming Interface
AST	Areal Surface Texture
BAC	Bearing Area Curve
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAT	Computer Aided Tolerancing
CAx	Computer Aided Technologies
COM	Component Object Model
DBMS	Database Management System
DMMs	Design, Manufacture and Measurement in Surface texture
GPS	Geometrical Product Specifications
ISO	International Standards Organisation
IT	International Tolerance (Grades)
MRR	Material Removal Required Process
NIST	National Institute for Standards and Technology
NMR	No Material Removed Process
NPL	National Physical Laboratory
PUMA	Procedure for Uncertainty MAnagement
PST	Profile Surface Texture
PTB	Physikalisch-Technische Bundesanstalt
SEM	Scanning Electron Microscope
STM	Scanning Tunnelling Microscope
VOT	Variance Orientation Transform
XML	Extensible Markup Language

1. Introduction

1.1 Background

The trend in global manufacturing, along with the emergence of computer-aided technologies (CAx), urges a rigorous and systematic common language to characterise geometrical products throughout the product supply chain. An international technical language, called Geometrical Product Specifications and Verification (GPS[1]), has created a synergy for design, manufacture and measurement. It uses rigorous mathematical definitions of geometric specifications to map that of verification, intending to save design modification and manufacture time, and to reduce scrap material in manufacture and measurement cost. Comprehensive implementations of the GPS-language globally, will promote future manufacturing moving to a knowledge driven economic environment, where design, manufacture and measurement are integrated into a single engineering process that enables 'right first time' every time fabrication of customised products (National Physical Laboratory [NPL], 2012). Such evolutions will force product technical specification and verification to be much more precise and with a clearer implementation methodology.

Over the last decades, continuing efforts have been directed toward understanding fundamental concepts and models in the GPS system, as well as developing optimised tolerance models and applications for the system. However as yet the GPS is largely a document based system which covers several kinds of geometric characteristics (such as size, distance, form, surface texture, etc.) and its implementation is viewed as highly complex, requiring high levels of understanding.

[1] GPS is also commonly used acronym for 'Global Positioning System'. However, the distinction between the two should be clear from the context.

The implementations of some geometric characteristics were hindered. One example is surface texture[2], one of the most complicated geometrical specification and verification systems in GPS. It is relevant for the whole surface manufacture chain from design through manufacture and qualification, and plays a significant role in determining the functional performances of a workpiece, e.g. friction, wear and lubrication. In recent years, the characterisation of surface texture has experienced a paradigm shift from profile to areal thanks to the rapidly development of advanced measurement instruments and information technology (Jiang et al., 2007a; 2007b). Surface design, manufacturing and metrology are however disconnected, becoming a very complicated and ambiguous system, especially since the necessary skills/expertise are often not available in global supply-chains, SMEs and multi-country manufacturing.

One of the essential reasons for this disconnect is the complexity of surface texture knowledge in GPS. Currently, there are 29 GPS published standards for profile and areal surface texture, a set of new standards, including ISO 25178 series, are being issued. Those paper-based documents which contain a wealth of information under the GPS matrix structure[3] have been recognised as being too complicated to be comprehended and implemented without an effective implementation methodology. Furthermore, some of the definitions in these standards still leave a room for several different interpretations (Leach & Harris, 2002; Scott, 2006). Misunderstanding caused by the ambiguities and imperfections can result in significant information loss between design, manufacture and measurement, especially when there is vast quantities of information for exchange.

These issues necessitate a deeper understanding of the underlying reasons (to be discussed in the next chapter), as well as a comprehensive implementation of surface texture in design, manufacture and measurement. The development of support systems and integrating them with CAx is one of the most efficient ways to allow partners collaborating effectively in creating innovative products.

This project was to develop a surface texture information system to bridge the knowledge gap between Design, Manufacture and Measurement in surface texture

[2] A deep discussion about the definition and characterisation of surface texture will be carried out in Chapter 2.

[3] GPS matrix will be detailed in Chapter 2.

(abbreviated as DMMs). A prototype of the CatSurf system was designed and developed. Currently, it is an executable system with three different modules which can be integrated with various Computer-Aided Design (CAD) systems such as AutoCAD and SolidWorks. A special module was developed for Rolls Royce with a single roughness parameter *Ra* for gas washed surfaces. The system was tested and recognised by various parties such as Rolls Royce, CAx and GPS experts, computing and mechanical engineers and researchers, etc.

1.2 Aims and objectives

The aim of this work is to facilitate engineers using updated GPS standards to design, manufacture and measure surface texture for fast, flexible and cost-saving manufacturing, by creating an integrated surface texture knowledge platform. To achieve this aim, the objectives of this project are classified as follows:

- *Understanding of knowledge gaps*: a deep understanding of knowledge gaps in different stage of DMMs will be carried out.
- *Knowledge modelling Methodology*: a methodology to model and manipulate the manifold and complex knowledge in surface texture will be developed.
- *Knowledge modelling for areal and profile surface texture*: the knowledge model for areal and profile surface texture will be developed utilising the knowledge modelling methodology.
- *Design and development of the CatSurf system*: The CatSurf system which includes one database and three modules each of five parts will be designed and developed.
- *Integration method between CatSurf and CAD systems*: It will develop a method to integrate CatSurf into a CAD system. Two test cases will be undertaken to implement the integration method.

The main objective of this work is to provide unambiguous surface texture information for designers and metrologists. Hence this thesis covers the knowledge of specification and verification in the design and measurement. This project does not cover details about manufacturing guidance such as process planning. Investigating the correlation between particular functional requirements and surface texture is also beyond the scope of this thesis.

1.3 Overview of the study

A brief description of the work undertaken is as follows:

- Chapter 2 presents a review of the surface texture characterisation and the current state of GPS together with the analysis of the knowledge gap between DMMs.
- Chapter 3 presents the knowledge modelling methodology for the CatSurf system.
- Chapters 4 and 5 develop the knowledge model for profile and areal surface texture respectively based on the methodology.
- Chapter 6 presents the design and development of the CatSurf system based on the knowledge model.
- Chapter 7 demonstrates methodology and implementation of the integration between CatSurf and CAD systems.
- Chapter 8 is a conclusion of the work presented in this thesis and recommendations for the future work are presented.

2. Knowledge gap analysis and literature review

This chapter analyses the knowledge gaps between DMMs. The objectives are to develop a better understanding of the knowledge gaps, to identify the potential research work and to clarify the scope of the work to be undertaken, and to carry out a literature review. These gaps are analysed with reference to their location between different phases. The underlying reasons for each knowledge gap are discussed and related reviews are presented in section 2.3. This chapter also summarises the brief history of surface texture from profile to areal characterisation in section 2.1. The origins and core ideas of GPS are also discussed in section 2.2 and it is explained that the methods carried out in the ensuing chapters are based on GPS requirements.

2.1 Surface texture – profile to areal characterisation

The texture is one of the key features of a surface. The definition of the term 'surface texture[4]' has been debated for a century although this term is used worldwide. The earliest official definition for 'surface texture' probably was "*relatively finely-spaced surface irregularities, the height, width, direction, and shape of which establish the predominate surface pattern*" in a US military standard (1949). The previous British standard defined surface texture as "*those irregularities with regular or irregular spacing that tend to form a pattern or texture on the surface*" in 1988 (BS 1134-1, 1988). It is worth noting, however, that the current ISO and British standards do not provide the definition of surface texture. These two definitions both highlight two keywords, which are 'irregularities' and 'pattern'. The complete texture of any surface can be described as, therefore, a combination of irregularities of various kinds and predominate pattern arising from different causes. The irregularities result from machine tool inaccuracies, deformation of the workpiece due to cutting forces,

[4] It is also called surface roughness, surface finish and surface topography. 'Surface texture' is the modern term used in international standards. Unless otherwise indicated, 'surface texture' is the only term in this thesis.

vibrations such as chatter marks, the inherent action of a particular production process, etc. This pattern shows significant correlation with function performance of workpiece (Whitehouse, 1994).

2.1.1 Profile characterisation

Surface texture has traditionally been defined from profiles according to the previous definitions. The 'surface profile' is "*result from the intersection of the real surface by a specified plane*", where '*real surface*' is "*surface limiting the body and separating it from the surrounding medium*" (ISO 4287, 1997). The surface texture defined by a profile is called profile surface texture (PST[5]) in this thesis. The most widely accepted classification of PST is roughness and waviness based on the different wavelengths of the irregularities. As indicated in figure 2.1, 'Lay' is the direction of the predominant pattern of the surface irregularities. The short wave component is defined as roughness, which is generated by the material removal mechanism such as tool marks; the long-wave component is defined as waviness produced by imperfect operation of a machine tool. Reason (1944a; 1944b) commented that this classification is "*neither very precise, nor inclusive of every kind of texture, but it will serve as a basis for discussion*". The previous British standard added another group to the classification: errors of form (BS 1134-1, 1988), which are generated by errors of a machine tool, distortions such as gravity effects and thermal effects, etc. The modern ISO standard 4287:1997 defines the combination of shortwave and long wave component as 'primary profile'. The PST parameters in GPS are defined based on these three different profiles: roughness, waviness and primary.

The characterisation of PST has lagged behind the surface measurement technology for many years. In the early stages, the tactile and visual clarification of PST had existed for decades. The emergence of the primary surface instrument devised by Tomlinson in the late 1910s began the development of instruments for the assessment of surfaces. The first truly commercial instrument named Talysurf 1 was invented by Reason from Taylor Hobson in 1939. Few parameters were defined during this period, such as the average roughness *Ra* and average peak-valley heights *Rz* and *Ry*. *Ra* was used as the control parameter in the UK and USA whereas peak parameters were used in Germany and USSR (Schlesinger, 1942; Schorsch, 1958). At the same time, Abbott

[5] Throughout this thesis, the term 'PST' is used as a substitute for 'profile surface texture'.

& Firestone (1933) developed the Abbott-Firestone curve[6] to characterise the seal and bearing performance. This idea in itself proved to be a fundamental step for the statistical descriptions of the surface. However, this curve cannot convey any spatial information.

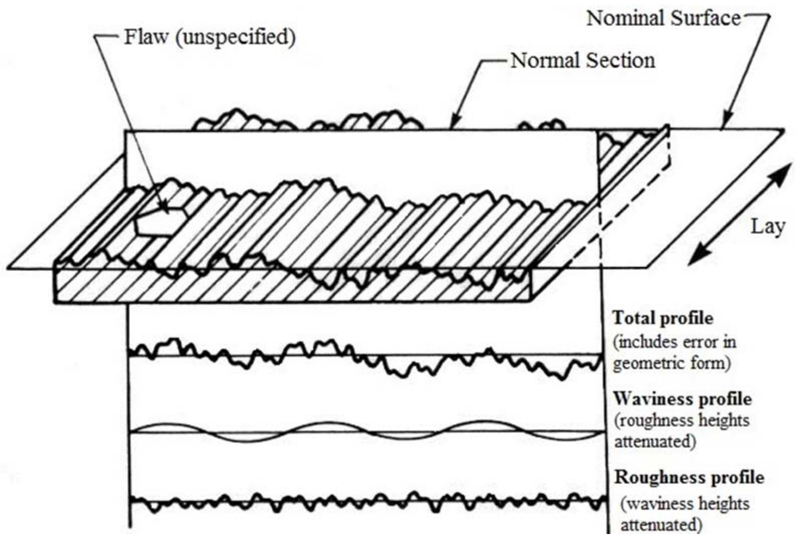


Figure 2.1 The concepts of PST include roughness, waviness, and lay (ASME B46.1, 2002)

Characterisation and instrumentation for PST changed dramatically when digital computers became widely available in the 1960s. The analogue surface signal can be converted to a digital signal, displayed and analysed by a computer automatically. It was realised that many surfaces manufactured by different methods have similar *Ra* values as seen in figure 2.2. Conscious of the limited capabilities of *Ra*, engineers and designers began looking for better ways to quantify a surface using computing capability. Many distinct parameters were designed mainly based upon custom and practice of surface descriptions used in the individual industries of their countries. By 1982, over one hundred parameters had been published many of which do not give independent information about the surface, and some had different names for the same evaluation. This became known as 'the parameter rash' (Whitehouse, 1982) and problems could arise in specification when a product was outsourced for manufacture. Some of the parameters therefore have been abandoned along with the development of International Standards. The parameters that were originally selected in many national standards had been deleted. It was realised that the probable development and

[6] It is also known as bearing area curve (BAC) and material ratio curve, which gives a cumulative statistical distribution of the surface profile's height.

specification of parameters would have been more logical through areal data collection analysis.

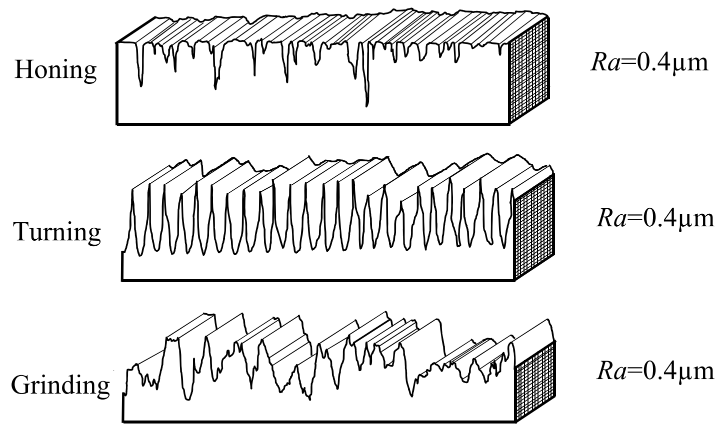


Figure 2.2 Surfaces produced by different processes with similar *Ra* values

2.1.2 Areal characterisation

Profile characterisation and instrumentation have dominated in both the industry and academic field for nearly a century. Many researchers argued that the profile approach was flawed in principle even though it is still greatly practiced (Blunt & Jiang, 2003). Since the first step in areal surface texture (AST[7]) analysis taken by Williamson in late 1960s (Williamson, 1967-1968), technology has progressed with the development of computing technology and areal instruments are now widely available. This has resulted in a paradigm shift from profile to areal characterisation (Jiang et al., 2007b).

2.1.2.1 Historical background for AST characterisation

The early areal instruments were making measurements with parallel traces using conventional stylus systems. The development of new measurement systems was slow until the advent of the new generation of personal computers in the 1980s (Jiang et al., 2007b). Areal instruments were then able to handle the large amount of data involved (Teague et al., 1982; De Chiffre & Nielsen, 1987). In the early 1990s, commercial AST instruments gradually became available. Contact stylus systems became mature, manufactured by companies such as Somicronic (Machpro, France) and Taylor Hobson (UK). Optical systems based on interferometry were also developed such as the WYKO system (Veeco, USA).

[7] The term 'AST' is used as a substitute for 'areal surface texture' throughout this thesis.

However, only a small number of statistical parameters were utilised in these pioneering commercial systems, due to the restrained development of areal characterisation. In the 1970s, a five nearest-neighbour ordinate method in AST data was designed to define a peak or pit (Nayak, 1971; Sayles & Thomas, 1977). In order to investigate contact phenomena of random surfaces, Whitehouse (1994) also defined three areal parameters: summit density; summit height and summit curvature. The three parameters, however, depend on sampling density, and the results could be distorted by measurement noise.

The major shift and development of novel concepts in areal characterisation came in the 1990s. Stout et al. were awarded a grant to produce a rationale for areal characterisation by developing both visual techniques and a subset of parameters to characterise AST (Stout & Blunt, 1994). The project report introduced the first definition of the so-called 'Birmingham 14' parameters. In 2001, an EU-funded AutoSurf project under the leadership of Rover/Brunel University developed an AST characterisation method for sheet material automotive applications. This project included characterisation for oil retention during storage of the coils, pressing performance and paint performance. A feature toolbox was used to solve real surface texture problems. At the same time, a project entitled 'SurfStand' under the leadership of Huddersfield University was founded by the EC. This project further developed the 'Birmingham 14' parameters, resulting in the introduction of a 'Feature' toolbox and robust and wavelet filter technologies. It laid the foundations for the standardisation of AST analysis. After the SurfStand and AutoSurf projects presented to ISO/TC 213 in 2002, a working group (WG) in ISO/TC 213 was set up to develop AST standards, which became the future ISO 25178 series.

Currently, the ISO 25178 series concerning terms and definitions, specifications and verification operators[8] is being developed by WG 16 in ISO/TC 213. It is the foremost series of standard providing a redefinition of the foundations of surface texture, and is based upon the principle that their nature is intrinsically 'three-dimensional'. It is anticipated that future work will extend these new concepts into the domain of profile metric surface analysis, requiring a total revision of all current PST standards (ISO 1302, ISO 4287, ISO 4288, ISO 11562, ISO 12085, ISO 13565 series, etc.). A recent

[8] Details will be discussed in section 2.2.

ISO/TC 213 meeting proposed to draft a new profile standard series (still named ISO 1302) with different parts matching ISO 25178 series (Scott, 2013). Table 2.1 shows all AST and PST standards in the general GPS matrix[9], where ISO 25178

- part 1 defines the indication of AST;
- part 2 defines the terms, definitions and AST parameters which include field and feature parameters (Scott, 2009);
- part 3 defines AST specifications operators;
- part 6 series (ISO 25178-6, ISO 25178-601, ISO 25178-602, ISO/DIS 25178-603, ISO/DIS 25178-604 and ISO/CD 25178-605) define the measurement methods and instruments;
- part 7 series (ISO/CD 25178-70, ISO/DIS 25178-71 and ISO 25178-701) define calibration requirements and software measurement standards.

Table 2.1 AST and PST standards in general GPS matrix

Chain link No.	*Geometrical characteristic of feature*	*PST standards*	*AST standards*
1	Product documentation indication - Codification	ISO 1302	ISO 25178-1(D)
2	Definition of tolerances - Theoretical definition and values	ISO 4287, 11562,12085, 13565-1,13565-2, 13565-3	ISO 25178-2
3	Definition for actual feature - Characteristic or parameter	ISO 4287, 4288, 11562, 12085, 13565-2,	ISO 25178-3
4	Assessment of the deviations of the workpiece - Comparison with tolerance limits	ISO 4288,12085	
5	Measurement equipment requirements	ISO 3274, 11562	ISO 25178-6, 25178-601, 25178-602, 25178-603(D), 25178-604(D), 25178-605(D), 25178-606 (D), 25178-607 (D)
6	Calibration requirements - Measurements standards	ISO 5436-1, 5436-2, 12179	ISO 25178-70(D), 25178-71, 25178-72 (D), 25178-701, 25178-702(D), 25178-703 (D)

Note: The symbol (D) denotes standards under development

In 2010, ISO 25178-6, ISO 25178-601, ISO 25178-602 and ISO 25178-701 became the first four published standards in AST; and ISO 25178-2, 25178-3 and 25178-71

[9] Details will be discussed in section 2.2.

were published in 2012. According to the schedule of WG16, other standards are expected to be published shortly.

2.1.2.2 New concepts for AST

PST parameters provide a simple approach to control the manufacturing process. They indicate changes in the process such as vibration of machine tool or tool wear. They are, however, not capable of diagnosing product functional performance directly (Jiang et al., 2007b).

The AST method attempts to characterise the fundamental and functional topographical features of the surface, including assessment of texture shape and direction, estimation of feature attributes and differentiation between connected and isolated features. AST characterisation is a genuine attempt to characterise areal features rather than a simple extension from profile to the areal case (Stout et al., 1993; Blunt, Jiang & Stout, 1999; De Chifre, 2000). Among 41 areal parameters defined in ISO 25178-2, only 15 of them are extended from profile. Many innovative concepts are introduced in the ISO 25178 series.

One of the new concepts in ISO 25178 series is the scale-limited surface. The term 'scale' can be recognised as an extension of the notion of the original term 'wavelength' in PST. Figure 2.3 shows the components of a scale-limited surface. The scale-limited surface depends on the filters or operations used. The S-filter removes unwanted small-scale lateral components of the surface such as measurement noise. The L-filter removes unwanted large-scale lateral components of the surface. The F-operation removes the nominal form. It is called an operation rather than filtration because it firstly uses optimisation to determine a best fit to the nominal form, and then removes the fitted form from the surface. Some F-operations such as association operation (introduce in section 2.2.2) have a very different action to that of filtration. Though their action can limit the larger lateral scales of a surface, this action is very fuzzy hence the fuzzy line for the action of the F-operation in figure 2.3 (ISO 25178-3, 2010).

The S-F surface is derived by using an S-filter and F-operation in combination on a surface, and an S-L surface by using an L-filter on an S-F surface. Both S-F surface and S-L surface are called scale-limited surfaces.

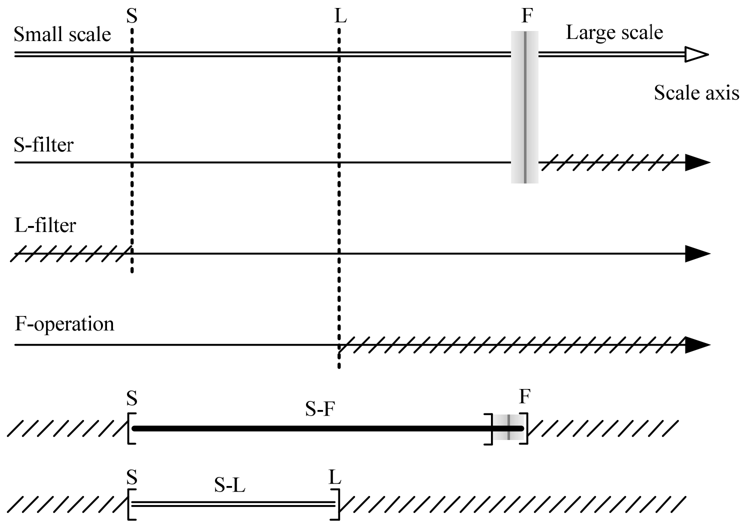


Figure 2.3 Scale limited surfaces in AST

The filtrations and operations of scale-limited surface are controlled by the nesting index. A nesting index is an extension of the notion of the original cut-off wavelength and is suitable for all types of filtrations. For example, the nesting index for a Gaussian filter is equivalent to the cut-off wavelength, and for a morphological filter (ISO/DIS 16610-41, 2012; ISO/DIS 16610-49, 2012) with a circular structuring element, the nesting index is the radius of the circular element.

Another difference between PST and AST is the filtration used. A profile extracted from a scale-limited surface is not mathematically the same as a profile measured according to the PST chain of standards. PST uses profile filtration in the traverse direction only which is orthogonal to the lay. AST uses areal filtration in both the X and Y directions which may or may not be related to the lay direction. This areal filter can produce very different results even with the same filter type and cut-off/nesting index.

AST characterisation does not require three different groups of parameters as profile parameters. For example, in AST parameters only *Sa* is defined for the arithmetical mean height parameter rather than the primary parameter *Pa*, waviness *Wa* and roughness *Ra* in the PST (ISO 4287, 1997).

2.2 Geometrical Product Specifications (GPS)

ISO/TC 213 defines GPS as "*an Internationally accepted concept covering all different requirements - indicated on a technical drawing - to the geometry of industrial workpieces (e.g. size, distance, radius, angle, form, orientation, location, run-out, surface roughness, surface waviness, surface defects, edges, etc.) and all related verification principles, measuring instruments and their calibration*". ISO/TC 213 was set up in 1996 by combining ISO/TC 57, ISO/TC 10/SC5 and ISO/TC 3[10]. The driving force was the necessity to consolidate specification and verification standards in the same technical committee such that there could be a communication dialogue between those who specify geometry and those who measure it. A series of concepts was proposed to facilitate fast and flexible manufacturing such as GPS-matrix structure, operation and operator, duality principle and uncertainties.

2.2.1 GPS Masterplan

One of the first documents resulting from what was to become ISO/TC 213 was ISO/TR 14638, the 'GPS Masterplan' (ISO/TR 14638, 1996). This document fits the general GPS standards into a matrix that contained what is known as the 'chains of standards' [11] and defined the 6 chain links that were necessary in order for a specification to be unambiguously and the measuring result used to verify it as traceable. The general GPS matrix is shown in table 2.2.

Chain link 1 deals with the drawing indication (often in a sort of 'coded'- symbol) of the characteristic of the workpiece. The standards in it define the symbols, how to use the symbol and the associated rules of 'grammar'. The standards also define the small difference in a symbol which could cause a major shift in meaning.

Chain link 2 defines the numerical values related to the code- symbols. The standards in it define the rules of translating from the code to 'human understandable' and 'computer understandable' values into SI-units e.g. the size in mm and vice versa.

---

[10] ISO/TC 57 "Metrology and properties of surfaces"; ISO/TC 10/SC5 "Technical drawings, product definition and related documentation - Dimensioning and tolerancing"; ISO/TC 3 "Limits and Fits".

[11] All related standards concerning the same geometrical characteristics (ISO/TR 14638).

Standards in chain link 3 make the supplementary definitions to extend the meaning of the theoretically exact feature. The non-ideal real world geometry[12] is always unambiguously defined in relation to the tolerance indication on the drawing.

Table 2.2 The General GPS matrix

Chain Link No.					
1	2	3	4	5	6
Product documentation indication - Codification	Definition of tolerances - Theoretical definition and values	Definitions for actual feature - Characteristic or parameter	Assessment of the deviations of workpiece - Comparison with tolerance limits	Measurement equipment requirements	Calibration requirements - Measurement standards
Specification of GPS Characteristics			Verification of GPS Characteristics		

Chain link 4 defines the detailed requirements for the assessment of the deviations of the work piece from the code-symbol, taking into account the definitions in chain link 2 and 3.

Chain link 5 describes specific measuring equipment or types of measuring instruments. It defines the characteristics of measuring equipment, which are influencing the uncertainty of the measuring process in which the equipment is involved.

Chain link 6 describes the calibration standards and the calibration procedures to be used, and verifies the functional requirements of the specific measuring equipment in chain link 5.

Chain links 1-3 describe the requirements for specification and verification is defined in chain links 4-6.

[12] It is called 'actual feature characteristic' which is based on sets of data points.

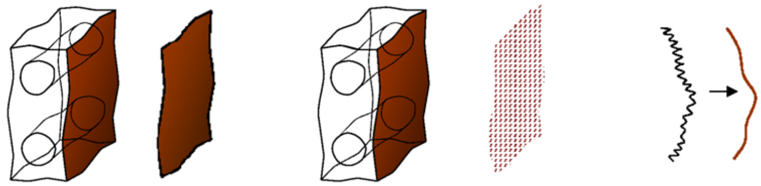
2.2.2 Operation and operator

The terms ‘operation’ and ‘operator’ are defined in ISO 17450-1 (2011) and 17450-2 (2012) respectively. These standards build on the ideas of the Masterplan and the early work carried out in TC 57 defining surface texture in terms of ideal measuring instruments. It was realised that these standards defined measurands rather than measuring instruments (Nielsen, 2012). In ISO 17450-1, ‘operation’ is defined as “*specific tool required to obtain features or values of characteristics, their nominal value and their limit(s)*”. Seven operations are defined which are termed ‘partition’, ‘extraction’, ‘filtration’, ‘association’, ‘collection’, ‘construction’ and ‘evaluation’.

- *Partition* is to identify bounded features such as point, straight line or plane.
- *Extraction* is used to identify a finite number of points from a feature, with specific rules.
- *Filtration* is used to distinguish between roughness, waviness, structure and form etc.
- *Association* is to fit ideal features to non-ideal features according to specific criteria which give an objective for a characteristic and can set constraints.
- *Collection* is to identify and consider some features which together play a functional role.
- *Construction* is to build ideal features from other features.
- *Evaluation* is to indentify either the value of a characteristic or its nominal value and its limit(s).

An 8th operation has recently been defined ‘reconstruction’ to reconstruct a continuous feature from a finite number of points and is the inverse of extraction (Scott, 2013).

Operator is an ordered set of operations. These operations can be used in any order. For example, partition, extraction and filtration are the three operations to obtain the ideal or non-ideal features of surface texture as shown in figure 2.4.



(a) Partition (b) Extraction (c) Filtration

Figure 2.4 Partition, Extraction and Filtration operations

2.2.3 Duality principle

From the definitions of ISO 17450-2, it begins to view specification and verification in terms of operators that consist of a number of operations in a defined order. Some operations are mechanical, such as the tactile sensing of the surface, others are mathematical. These operators define characteristics and specifications and put constraints on these characteristics. The verification operator (i.e. what happens in the measurement) then can be determined from the mapping of specification operator (i.e. the definition of the measurand). This allows the comparison between the two operators and provides the quantification of the differences between them in terms of uncertainties. It allows users of the GPS to decide on a case by case basis whether a given measuring process is good enough to be used to verify a particular specification, or whether the uncertainty is too high (Nielsen, 2012).

In this context, the so-called 'duality principle' is formally introduced in ISO 17450-1. As shown in figure 2.5, this principle is the way to view the verification as ideally being a mirror image of the specification, but not necessarily be the same.

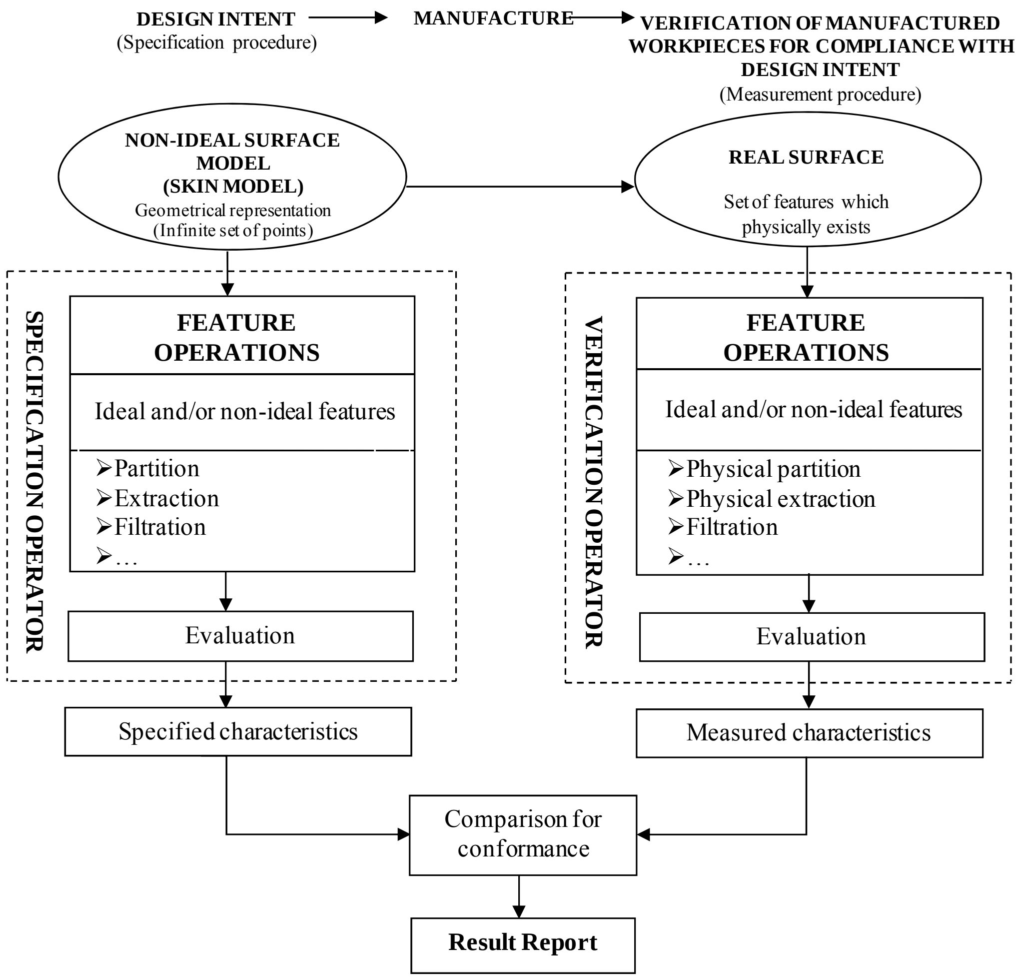


Figure 2.5 The duality principle

2.2.4 Extended uncertainty system

The main work of ISO/TC 213 has been focusing on decreasing the ambiguities of GPS language. A significant tool to describe the ambiguities is ‘uncertainty’. It was realised that disagreements on the measurement values cannot always be explained by the presence of conventional measurement uncertainty only (Nielsen, 2006), thus an extended uncertainty system has been developed. In this system, ‘uncertainty’ is extended as an expression of ‘lack of information’ in different stages of the entire product lifecycle more than measurement process.

The extended uncertainty system defines seven uncertainties in the stages of ‘function’, ‘specification’, ‘manufacture’ and ‘verification’. These uncertainties are shown in figure 2.6. The uncertainty arising from the difference between the specified specification and the related functional requirement is defined as correlation uncertainty. The incompleteness of the specification is defined as specification uncertainty. It was realised that disagreements on the measurement values cannot always be explained by the presence of conventional measurement uncertainty only. The extended measurement uncertainty is the combination of method uncertainty and implementation uncertainty. Method uncertainty expresses how well a selected verification process mirrors the specification. It occurs when the actual verification operators are compared to actual specification operators. Implementation uncertainty is only involved in the verification process, and it describes the accuracy of the instruments used, the influence of the environment, and the metrologist, etc.

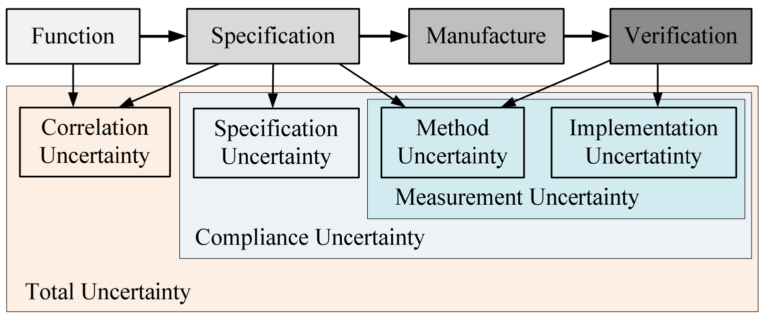


Figure 2.6 Extended uncertainty system in GPS

In order to explore the extended uncertainties, the ISO 14253 series (ISO 14253-1, ISO 14253-2 and ISO 14253-3) have been published to estimate uncertainty for GPS measurement and introduces the novel idea of a target uncertainty and the PUMA (Procedure for Uncertainty MAnagement) method. The PUMA aims at proving the actual uncertainty is less than the target uncertainty with minimum effort, rather than estimating the actual uncertainty as accurately as possible. To evaluate measurement uncertainty, the updated GUM (Guide to the expression of uncertainty in measurement) introduced the Monte Carlo method for uncertainty evaluation (JCGM 100, 2008; JCGM 101, 2008). The concepts and methods given in the GUM can without problems are used on the specification operator, and the resulting specification uncertainty values can therefore be compared with the corresponding measurement uncertainty values. However, the specification uncertainty values evaluated for specifications given on existing engineering drawings, generally are much larger (5-10 times or even more) than the 'normal' measurement uncertainty used for the measurements in industry to verify the conformance with the specification (Bennich, 2003). Far too much resources is used in measuring the wrong/unnecessary characteristics with a high precision, compared with the resources allocated to set up proper specifications - which would have a small specification uncertainty.

2.3 Knowledge gap analysis

This section aims to analyse the knowledge gap existing in/between different phases of DMMs. Six gaps will be analysed as shown in figure 2.7.

- Gap 1 - the limitations in existing surface texture systems;
- Gap 2 - the restrictions for existing data representation methods for surface texture;
- Gap 3 - the integration problems between surface texture system and CAx systems;
- Gap 4 - the limited correlation between function and specification;
- Gap 5 - the knowledge gap from specification to the manufacture and measurement;
- Gap 6 - the knowledge gap from measurement to specification.

Each gap is analysed in the following sub-sections.

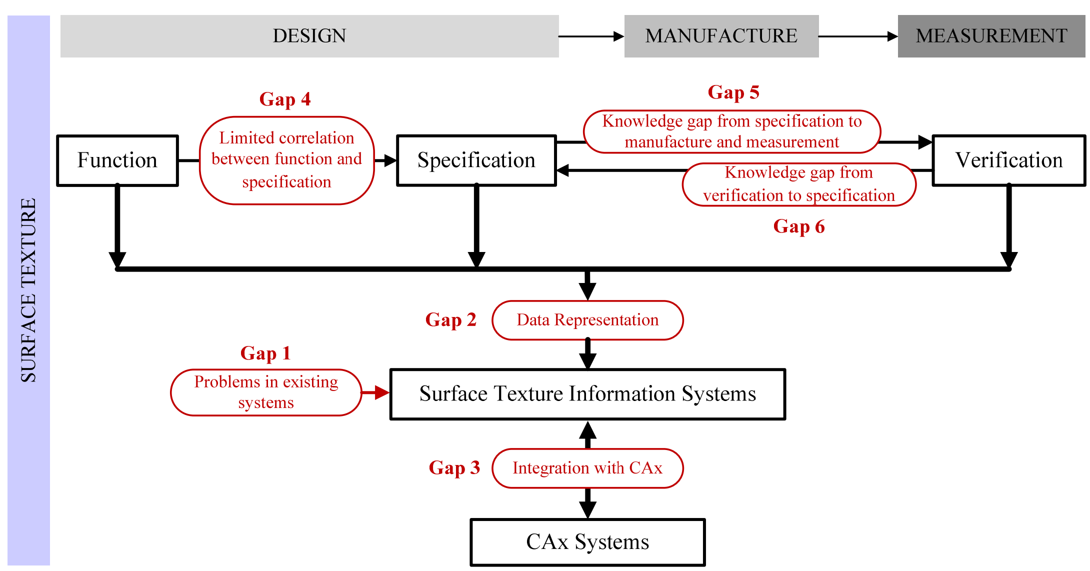


Figure 2.7 Knowledge gaps between DMMs

2.3.1 Gap 1 - the limitations in existing surface texture systems

There are more information systems dealing with measure surface metrology information than systems for surface texture design intent. Bui (2007) has proposed an internet-based surface texture analysis and information system to deal with surface metrology information such as filtration and parameter evaluation. Sacerdotti et al. (2003) has established a so-called 'SCOUT' surface characterisation open-source universal toolbox which was focus on software support on areal characterisation of steel sheet. Different reference software were developed by National Institutes such as PTB (Physikalisch-Technische Bundesanstalt) (Jung et al., 2001) and NIST (National Institute for Standards and Technology) and NPL (Li et al., 2011).

The only surface texture designing support system in the 20th century was an interactive surface modelling system (ISM) which was proposed by Rosen (1995). This system was based on the traditional PST standards which utilise a symbolic language for expressing tolerances in technical drawings. ISO 1302:2002, one of the latest PST standards in the GPS framework defines the specification of PST. There is more information concerning design, manufacture and metrology in this standard, however, it has to this point not been implemented very successfully during the design stage. Many designers have not yet adopted complete surface texture specification, in order to bridge the knowledge gap in product life cycle, to reduce the product cost, and to improve manufacturing efficiency and qualification rate. The majority of manufacturing companies and commercial CAD systems are still employing old surface texture standard versions or do not completely conform to the standards, which leads to big specification uncertainty (Bennich & Nielsen, 2005) compared with ISO 1302:2002.

In this context, a so-called 'VirtualSurf' project was undertaken to develop a novel knowledge-based system for PST at the University of Huddersfield (Wang, 2008; Xu, 2009). A unified categorical object modelling mechanism based on category theory was developed to structure the knowledge of PST. An initial 'VirtualSurf' system was designed based on the implementation of a categorical database management system (DBMS). The first stage of this project (Wang, 2008) was the design of the framework for the 'VirtualSurf' system, which included the novel utilisation of category theory. The second stage (Xu, 2009) was focused on the implementation of the categorical DBMS. This system can be considered as a **milestone in the utilisation of PST in GPS**, however, the 'VirtualSurf' has not been developed as a comprehensive functional system for practical implementation. For example, the 'Function' part[13] of the system was a 'function performance report' rather than having practical correlation with specification; and the 'Verification' part[14] only provided very basic measurement information for specifications. Moreover, a comprehensive surface texture system with support of AST is required due to the high functional demand of surface texture in industry, and with the rapid development of areal characterisation and standards publication.

2.3.2 Gap 2 - the restrictions of existing data representation methods for surface texture

It was discovered by Wang and Xu that traditional data models such as relational model and object-oriented model had limitations to efficiently support complex data structures and to reflect the complicated relationships among engineered artefacts and surface texture GPS standards (Wang, 2008; Xu, 2009; Lu, 2012). The relational model will not benefit some new applications such as engineering databases, e.g. CAD, because the attributes of simplicity, including minimalism and non-redundancy make the relational model unrepresentive of the way humans model the world. The object-oriented data model has also been found to lack both a universal formal basis and mathematical foundations to ensure that the database remains a coherent and reliable system.

Currently, there are twenty-eight large tables related with selection of specification or measurement parameters within twelve PST standards (see table 2.1). AST involves seventeen standards which include more than sixty large tables. A large number of GPS terms and complex relationships between them are defined. If a relational model is utilised, in order to undertake the relationship normalization, more than eighty-eight large tables have to be divided into smaller tables. As there are complex relationships between different attributes within tables, a large number of new relationship tables are required to be defined. It is difficult to tackle a large number of tables by using a

[13] The 'Function' part in the 'VirtualSurf' system was designed to explain functional requirements, and the 'Verification' part was developed to provide suggested measurement parameters.

relational database especially the standards in which terms or data need to be updated on a frequent basis.

An object database is suitable for applications dealing with very complex data, as it can store complex data and relationships between data directly. However, there are a large number of multi-level relationships that have to be considered for example relationships between different objects in a class or relationships between different objects in different classes which mean that more classes and functions have to be defined to express these complex relationships. A certain number of relationships are very general and ambiguous and are unable to be structured by mandatory mathematical functions. Using a collection of objects which are interlinked via pointers of some sort, the relationship normalization practices can be complemented in an object database, however, based on the 'graphic theory' (including trees), construction of the object model is rather difficult if one is to fully and intelligibly establish the complex relationships.

The 'VirtualSurf' project presented a unified categorical object modelling mechanism based on category theory. Category theory is a relatively new and high-level (abstract) form of mathematics language that focuses on how things behave rather than on what their internal details are (Walters, 1991; Barr & Wells, 1995). It has the capability for providing an effective and natural formalism for object-based databases (Rossiter, Nelson & Heather, 1994). One of the attractions of category theory is the ability to combine diagrammatic formalisms as in geometry with symbolic notations as in algebra: in computing science, diagrams are a common way of mastering complexity and symbolic notation is used for proofs and computation.

The 'VirtualSurf' project utilised category theory to develop an object-based modelling mechanism. Due to the clear and logical mappings in the modelling, the devised categorical DBMS (see section 2.3.1) has been proved to be on average 10 times faster than an analogue mySQL product when processing a query operation, as well as an average 1/3 memory cost of traditional relational DBMS when containing more than 500k data in memory (Wang, 2008; Xu, 2009; Lu, 2012). This formalism, however, has still thrown up some issues. One of the essential problems is the rigorous application of category theory. The major definitions of category theory (to be detailed in chapter 3), are based on the categories, objects and arrows in/between them. The object-based categorical model was focused on the objects rather than categories and relationships between categories. It significantly limited the effectiveness of category theory in dealing with complex relationships. A more rigorous categorical model is required to completely utilise the advantages of category theory.

Another problem is the implementation of the categorical model. Currently, there is no particular database available for category theory. Using an object-based database structure to implement categorical model has limited the functionality of category theory. This issue however is beyond the scope of this thesis. A new project proposal is required to solve this problem.

2.3.3 Gap 3 - the integration problems between surface texture information systems and CAx systems

Currently, the domain CAD/CAM/CAE multi-platform commercial software suites, such as CATIA (Dassault Systemes), AutoCAD (Autodesk), Pro/Engineer[14] (PTC Inc.), SolidWorks (Dassault Systemes) and UGS NX (Siemens), have all developed the geometrical specification systems. Most of the geometrical specification systems include PST symbol support. As shown in table 2.3, AutoCAD does not provide any PST support whereas AutoCAD Mechanical provides a PST symbol tool which is a simplified version from ISO 1302:2002. CATIA and SolidWorks from Dassault Systemes both provide PST tools, while the former utilises a very old version of ISO 1302 (1965), and the latter uses the same version as AutoCAD Mechanical. The PST tool in Pro/Engineer has the same old version as CATIA. The UGS NX from Siemens is utilising the surface texture standards of United States - ASME Y14.36M-1996. None of the listed systems have database support for PST.

It is arduous and time consuming to finish an unambiguous surface texture specification for designers without the availability of a support tool in current commercial CAx systems, because of the greater number of GPS standards and intricate related knowledge concerning in DMMs (see table 2.1). This gap necessitates integration between surface texture systems and CAx systems.

[14] A product now is called as PTC Creo, created by Parametric Technology Corporation (PTC).

Table 2.3 Status of surface texture specification design in commercial CAx systems

Commercial CAD Systems		*Surface Texture Specification Design*	*Surface Texture Standards*		*Database support*
			Versions	*Indications*	
Autodesk	AutoCAD	None	None	None	None
	AutoCAD Mechanical	Surface Texture Symbol Tool	A simplified version from ISO 1302:2002	milling Ra 3.2 ⊥	None
Dassault Systemes	CATIA	Roughness Symbol Tool	ISO 1302:1965	3.2	None
	SolidWorks	Surface Finish Symbol Menu	A simplified version from ISO 1302:2002	milling Ra 3.2 ⊥	None
PTC	Pro/Engineer	Surface Finish Tool Menu	ISO 1302:1965	3.2	None
Siemens	NX (Unigraphics)	Surface Finish Symbol Tool	ASME Y14.36M-1996	Mill 2.5/Rz 6.3 =	None

2.3.4 Gap 4 - the limited correlation between function and surface texture specification

Designers have responsibility to ensure that the assigned surface texture specification will satisfy functional requirements. However, some functions, such as engine scenario are very complex and almost impossible to express purely in terms of surface texture or geometry without having to be overly restrictive. In most cases, the assigned specification does not always truly express the functional requirements since it is really challenging to find a rigorous correlation. This difficulty, as described by Whitehouse (2012) is "*perhaps the biggest inverse problem in manufacturing*". The difference arises from a less than perfect correlation between a specification and the intended function of the workpiece, expressed in the term of correlation uncertainty. It characterises the fact that the intended functionality and the specified characteristics may not be perfectly correlated, expressing the knowledge gap between function and specification.

It is not very common to establish an evaluation approach for correlation uncertainty, although there is large amount of research concerning surface texture in application areas such as tribology and lubrication. The concept of correlation uncertainty has been rarely studied in engineering, because of both vastness and diversity of the functional requirements and also the number of specification items which are required to simulate a function. The only correlation uncertainty study was proposed by Dantan (2010) and proposed a model for the expression and an evaluation method of the correlation uncertainty in the application of gear conformity based on the Axiomatic Design matrix and the Monte Carlo Simulation.

In researching particular function cases in surface texture, studies relating to the relationship between functional requirements and surface texture must vary widely in scope. To clarify the large range of functions related to surface texture, Whitehouse (2001) classified the functions and surface features using the separation of the surfaces and their lateral movement. This classification is an essential element in trying to understand how functional performance is influenced by surface texture. However, identifying specific surface texture parameters and relating these to function is still fraught with problems. Little or no convincing evidence is available to link very specific surface parameters to function. In light of this uncertainty, a pragmatic empirical approach is usually adopted in that a number of parameters are investigated to get the best correlation between parameter and function, and then the limit value is tightened such that the workpieces in the grey zone (uncertainty zone) will be rejected. A lower correlation uncertainty would obviously allow for the rejection of fewer potentially good parts.

Some investigations of PST and AST influence the functional performance which contribute for estimating correlation uncertainty are described following.

2.3.4.1 Function performance and profile surface texture

Long before scientific studies of surface texture developed in the twenty-first century a number of interesting concepts emerged relating surface characteristics to friction, lubrication and to a limited extent, wear. Most of the evidence for the growth of this conceptual appreciation of the role of surface texture is drawn from the writing of natural philosophers and engineers published in the last few centuries. A number of detailed studies of surfaces and the quantification of surface feature of importance in tribology are from the late 1930s related to the new high performance aircraft engines. It was realised that surface texture could play a significant role in engine performance but no evidence has been found that there existed any precise ideas about what this role actually was.

The rapid development of PST characterisation and instrumentation from the 1960s onwards, provided a solid foundation for the analyses of the role of surface texture in different applications especially in tribology and many papers were subsequently written on the effect of surface texture on the wear of engine components. The effects of surface texture on the progressive wear on deep drawing dies were studied by Christiansen and De Chiffre (1997). The wear progress was quantified using *Rk* family parameters in both profile and areal. It was concluded that progressive wear in deep drawing dies can be suitably characterised by the areal parameters *Spk* and *Svk*. Kumar (2000) introduced an engine liner wear volume calculation method by calculating the material ratio curve difference before and after wear. The method used profile measurement and assessment which is in fact inadequate for true wear a further methodological issue identified in the experimental result was that there was no relocation technique applied in the measurement. The effects of surface texture on the wear of the liner and rings in an engine in particular were studied by Lakshminarayanan (2008), where parameter *Ra*, and *Rk* family parameters were applied in the investigation. It was found *Rk* and *Rvk* could be substituted into the wear rate formula for normal surfaces with peaked roughness, in an effort to enhance the applicability to wide-ranging surface texture. At the same time, Pawlus (2008) presented a method to determine truncation parameters during an abrasive machining process. Other methods to measure microscopic wear on general engineering surfaces based on the PST parameters of the worn surfaces have been developed, such as Jeng's (2002) method which does not require any information of the initial surface.

2.3.4.2 Function performance and areal surface texture

Although profile line roughness characterisation has been useful to date, the resulting parameters do not contain information on detailed spatial variation, or areal and volumetric aspects of lubricant retention capability. However, the roughness characterisation of an area of surface, through mapping the geometric features over an area can provide insight into the physical and functional behaviour of surface. Gåhlin (1998) introduced a method to measure the areal local wear volume and to map the distribution of wear by comparing the topography of the same surface region before and after testing. They used AFM and inherent AFM software to calculate bearing volume and display the wear distribution. In this method, the bottom of sharp cavities or other topographical feature that can be considered to be unaffected by the wear process are used as positioning references for repositioning. Suh (2003) found that areal functional parameters, such as the surface bearing index and fluid retention index, clearly showed progressive changes as the surfaces wear and reach scuffing. It was shown that the functions and related parameters can be used to correlate the topographical changes to the meaningful physical changes occurring during this process. Krzyzak (2006) studied the changes of AST of a piston skirt during a 'zero-wear' process and analysed the effect of initial AST on piston skirt wear.

Lubrication at the workpiece-tool interface also plays an important role in the product quality control of sheet metal forming processes. Surface microstructures of sheets have a great influence on the development of lubrication films. Liu (2009) used a strip drawing test to investigate the effects of the rolling direction of aluminium alloy sheet and lubricant on the friction behaviour in sheet metal forming. The measurement results of the AST of the sheets indicate that the surface parameters of the sheets such as *Sa* and peak-valley height decrease after the strip drawing test at different angles between the sliding and rolling directions (Evans, Snidle & Sharif, 2009; Krupka, Svoboda & Hartl, 2010).

How AST parameters play role for surface wear in bio-materials tribology has been discussed recently. There are several naturally occurring circumstances in biology where surface texture is important e.g. the wear of orthopaedic implants. Blunt (2009) carried out qualitative examples to illustrate how the use of advance co-ordinate and surface metrology has made measurement of wear possible for the newest generation of orthopaedic materials. A case study of wear ranking of hard-on-hard bearing for prosthetic hip joints illustrated the capability of advanced surface metrology to pre-screen materials and by analysing in details their surface texture expensive and time-consuming testing can be avoided.

Table 2.4 summarises the correlation of different kind of surface texture parameters on different function situation reviewed previously. All the related research in the connection between surface texture and function in different engineering applications contributed to the estimation of correlation uncertainty. As commented by Whitehouse (2002), even though table 2.4 may not help much in associating one parameter with specified type of function, it do indicate that only types of parameters are meaningful.

Table 2.4 The influence of surface texture on function.

Key: √√√√ High correlation, √ Very little correlation

	Surface Texture Parameters							
	Profile				*Areal*			
Function	Heights	Distribution and shape	Slopes and curvature	Lengths and peak spacing	Amplitude areal	Spacing areal	Functional areal	Hybrid areal
Typical parameters	*Ra Rq Rt*	*Rsk Rku*	*RΔq*	*RSm HSC*	*Sa Sq*	*Sal Std*	*Vvv Vmp*	*Sdr Sdq*
Forming & Drawing	√√√√	√√√	√√√	√√√√	√√√	√√√√	√√√√	√√
Painting & Plating	√√√√	√√√	√√√	√√√	√√√	√√√√	√√√√	√√√
Friction	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√
Galling	√√√√	√√√	√√√√	√	√√√		√√√√	√√√√
Wear	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√
Joint stiffness	√√√√	√√√√	√√√	√√√	√√√√	√√	√√√√	√√
Slideways	√√√√	√√√√	√√√	√√√√	√√√√	√√√√	√√√√	√√
Electro-contacts	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√	√√√√
Bonding & Adhesion	√√√√	√√√√	√√√	√√√	√√√√		√√√√	√√
Fatigue	√√√√	√√√	√	√	√√√		√√√√	
Stress & Fracture	√√√√	√	√√√		√√		√√√√	√√
Reflectivity	√√√√		√√√√	√√√√	√√	√√	√√√√	√√√√
Hygiene	√√√√	√√√	√√√		√√√		√√√√	√√
Bearings	√√√√	√√√√	√√√	√√√	√√√√	√√	√√√√	√√√√
Seals	√√√√	√√√√	√√√√	√√√	√√√√	√√	√√√√	√√√√

2.3.5 Gap 5 - the knowledge gap from specification to the manufacture and measurement

Surface texture specification is a design step in which control elements are stated, accommodating the design requirements of parts and their functional surfaces commensurate with production capabilities. The assigned specification will be interpreted by engineers and metrologists involved in the component manufacture and measurement. Surface texture verification takes place after the specification process. It defines how specification data will be converted into measurement parameters and how a metrologist determines whether the surface of a workpiece conforms to the specification. Sometimes the language of a standard is open to interpretation or gives equal value to choices that are not equivalent. In those cases an ambiguity (interpreted as specification uncertainty) is built into the specification. There will be a large number of choices while the metrologist attempts to make a verification process decision according to an incomplete specification.

The purpose of a specification is to guide the manufacturing and measurement, thus a single specification without a verification process is meaningless. The incomplete specification generates specification uncertainty only when applied to a verification process, therefore, the specification uncertainty can be used to quantify the gap between specification and verification. Specification uncertainty generally is much larger than the 'normal' measurement uncertainty used for other GPS measurements in industry.

In order to reduce the specification uncertainty, the specification of PST gets more complicated as shown in figure 2.8. According to ISO 17450-2:2002, the specification uncertainty quantifies the ambiguity in actual operators set out by the specification. Specification uncertainty can leads to ambiguous verification process selections by metrologists. Examples of issues that can cause specification uncertainty in surface texture are as follows.

1. Ambiguous definitions in standards, for example parameter *RSm* definition given in ISO 4287 (1997), different calculation directions cause different parameter results (Leach & Harris, 2002; Scott, 2006).
2. Incomplete standard definition of control elements. As shown in figure 2.8d, for PST specification, there are ten different control elements. The absence of any one or more of the elements will result in specification uncertainty. For example undefined transmission band or surface texture lay[15].
3. Ambiguous understanding about default operations, e.g. default value of comparison rule[15] in ISO and ASME is the '16%-rule', but in some internal company standards it is the 'max-rule'.

The first issue caused by the ambiguity in standards cannot be avoided, however, through rigorous control of the specification control elements and conscious explanation for default operations the latter two issues can be tackled.

[15] Will be detailed in chapter 4.

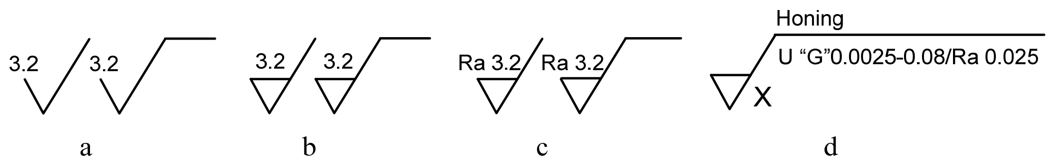


Figure 2.8 Different versions of the surface texture indication used in the drawing. a. the 1955 version, high specification uncertainty. b. the 1965 version , up to 300% specification uncertainty. c. the 1991 version, up to 30% specification uncertainty (Bennich & Nielsen, 2005). d. the ISO 1302: 2002 version, low specification uncertainty

A complete, unambiguous specification would enable metrologists to quickly discern implementation of the measurement of the surface easily. A complete specification is not one which specifies all of the possible measurement details, but rather one which can achieve communication with the verification, and with a minimum number of operations to give the most measurement details.

2.3.6 Gap 6 - the knowledge gap from measurement to specification

When a measurement process is performed by following the specification, there is always specification uncertainty, so a verification process must be selected from the series of verification operations generated from the interpretation of specification. The method uncertainty expresses how well a selected verification process mirrors the specification. It occurs when the actual verification operators are compared to actual specification operators and is the last step of verification. The method uncertainty is utilised to express the gap between verification and specification. Implementation uncertainty is only involved in the verification process, and it describes the accuracy of the instruments used, the influence of the environment, operator, etc. The method uncertainty is perfect to express the gap from verification to specification.

According to ISO 17450-2:2002, method uncertainty is the uncertainty arising from the differences between an actual specification operator and the actual verification operator, disregarding the physical deviations of the actual verification operator. This uncertainty accounts for the difference between what the specification calls for and what is implemented in the verification process, assuming that the verification process has no physical deviations.

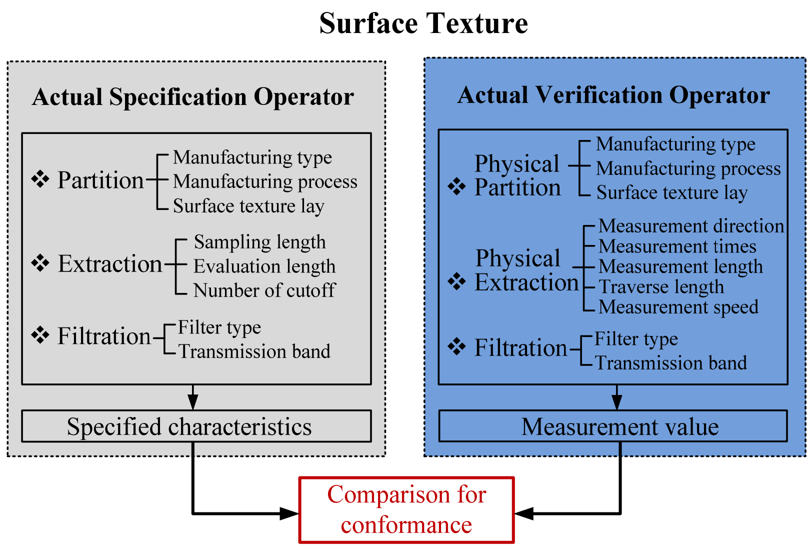


Figure 2.9 Method uncertainty - difference between actual specification operator and actual verification operator.

As shown in figure 2.9, the actual specification operator of surface texture includes partition, extraction, filtration and evaluation operations. As there are only ten control elements in the PST specification, it is impossible and unnecessary to detail every measurement procedure and condition in these operations. The main sources of method uncertainty from the difference of these operations between specification and verification are listed below.

1. **Difference between the extraction operations of specification and verification.** The extraction operation of the verification process is composed of the measurement direction, number of measurements, measurement length, traverse length, measurement speed, etc. As not all of these verification operations are detailed in the specification; the number of measurements, measurement length and traverse length can be determined by other control elements e.g. number of measurements can be determined by the comparison rules and the upper or lower limit. Measurement direction and measurement speed are totally determined by the metrologist, which will generate different measurement values.

2. **Difference between the filtration operations of specification and verification.** The difference in implementation of a filter is the main factor in the filtration operation. For example, if a Gaussian filter is detailed in the

specification, in the implementation of the verification process, there are different kinds of algorithms that can be utilised i.e. convolution algorithms (Whitehouse, 1967-68; Raja & Radhakrishnan, 1979), fast and reliable convolution algorithms (Krystek, 1996), Fourier transform based algorithms (Raja & Radhakrishnan, 1979) and approximation algorithms (Yuan, 2000). Any difference between the outcomes of these algorithms is one of the sources of method uncertainty.

3. **Difference between the evaluation operations of specification and verification.** In surface texture, the evaluation operation is the calculation procedure of the specified parameter value. In the verification process, a different instrument may have some differences in their interpretation of the calculation of a parameter. For example, the definition of parameter *Ra* in ISO 4287 of $Ra = \frac{1}{l}\int_0^l |Z(x)|dx$ is a continuous model, but in implementation, PTB and NIST use a discrete model, whereas NPL use a continuous model based on interpolation between discrete points, i.e. a particular reconstruction operation.

The implementation uncertainty defined in ISO 17450-2 is the narrow definition of traditional measurement uncertainty. The evaluation of method uncertainty assumes the implementation uncertainty is zero. But even if the implementation uncertainty is zero, it is impossible to reduce the measurement uncertainty below the method uncertainty. To reach a low measurement uncertainty it is not only necessary to have accurate instruments, a good environment, a trained operator, etc, it is also necessary that the measuring process measures what the specification requires. A method is needed to generate a series of detailed verification parameters according to the specification and guarantee the measuring process measures exactly what the specification requires thus reducing the method uncertainty.

As far as cost is concerned, if the metrologist invests in the ability to measure a workpiece with low measurement uncertainty while specification uncertainty is high, the design cost may be low, and measurement costs can be much higher, thus the total cost can still increasing and the total uncertainty is still high. If designers create a complete specification with low specification uncertainty then measurement uncertainty will also be decreased. In this case the design cost may be high but measurement cost will be low while the total cost may not change and the total uncertainty is lower. This is because a complete specification can give inspectors detailed information about how to measure the component so the method uncertainty and related measurement uncertainty will decrease with the reduction of specification uncertainty. Hereby, the latter can give us clear information - the control of specification uncertainty is able to distribute the product resource in a more effective and economical manner.

2.4 Summary

This chapter has reviewed some of the key topics related to this project. The knowledge gaps stated above were analysed summarised as follows:

1) A better support for AST is required following the latest development step in standards while the AST characterisation is becoming more widely used.
2) The mathematical based GPS language requires more implementation in industry.
3) The knowledge gaps 1-3 highlight the requirement of an integrated surface texture support system and rigorous knowledge modelling.
4) The knowledge gaps 4-6 indicate the ambiguities in/between different steps of function, specification and verification. A more precise understanding of the core ideas of GPS language is the key to assess the degree of such ambiguity.

In the following chapters, knowledge modelling for surface texture and a CatSurf system will be designed and developed to address the above knowledge gaps.

3. Knowledge modelling methodology

This chapter is set out to tackle gap 2 (the restrictions of existing data representation methods for surface texture) which was presented in the last chapter. The objective of this chapter is to update the existing categorical object model to be fully functional. A rigorous categorical model is developed, based on category theory, to model the specialised knowledge in AST and PST.

3.1 Introduction for category theory

Category theory can represent all standard mathematical structures and manipulations as predefined categories. It explores the relationships between different kinds of mathematical objects, and ignores unnecessary detail to give general definitions and structural results. It is a high-level (abstract) and efficacious language that focuses on how things behave rather than on what their internal details are (Walters, 1991; Barr & Wells, 1995). With the facility to specify formally transformations between different types of mathematical structures, category theory provides a powerful way of modelling complex systems with heterogeneous structures. Some good starting literature on category theory includes: Pierce (1991), Barr & Wells (1995) and Awodey (2006).

Category theory is based on the concept of a morphism, which is an abstraction derived from structure-preserving mappings between two mathematical structures, generally thought of as an arrow and represented by '→' (Lane, 1971). The arrows can denote any static condition or dynamic operation and therefore can cope with descriptive, prescriptive equivalent views. For example, the arrow is a generalisation of mathematical symbols such as $>$, $=$, $\subset$, $\in$ and $f(x)$ with the usual respective meaning of comparison, equality, partition, membership and functional image.

A *category* **C** consists of a collection of objects A, B, C, … and a collection of morphisms or arrows between objects $f: A \to B$, $g: B \to C, \ldots$, that are closed under composition and satisfy the following conditions.

- For each arrow f there are given objects: dom(f), cod(f) called the domain and codomain of f. We write: f: $A \rightarrow B$ or $A \xrightarrow{f} B$ to indicate that A = dom(f) and B= cod(f).

- Given arrows f: $A \rightarrow B$ and g: $B \rightarrow C$, that is, with: cod(f) = dom(g), there is a given arrow: $g \circ f$: A $\rightarrow C$, called the composite of f and g.

- For each object A, there is an identity arrow id_A: $A \rightarrow A$ satisfying the identity law: for any arrow f: $A \rightarrow B$, $id_B \circ f = f$ and $f \circ id_A = f$.

The collection of all morphisms from A to B in category **C** is denoted $\mathrm{hom}_{\mathbf{C}}(A, B)$ and called the *hom-set* between A and B (the collection of morphisms is not required to be a set). A number of types of morphisms defined in category theory are monic (monomorphism), epic (epimorphism) and isomorphic.

In category theory, a morphism f: $A \rightarrow B$ is an isomorphism if and only if there is an inverse morphism g: $B \rightarrow A$ such that $g \circ f = id_{\mathrm{A}}$ and $f \circ g = id_{\mathrm{B}}$. The morphism f: $B \rightarrow C$ is monic if for any two morphisms between A and B in a same category g: $A \rightarrow B$ and h: $A \rightarrow B$, the equality $f \circ g = f \circ h$ implies $g = h$. The morphism f: $A \rightarrow B$ is epic if for any morphisms in the same category g: $B \rightarrow C$ and h: $B \rightarrow C$, the equality $g \circ f = h \circ f$ *implies* $g = h$. Figure 3.1 shows diagrams for an isomorphism, a monic and an epic in category theory.

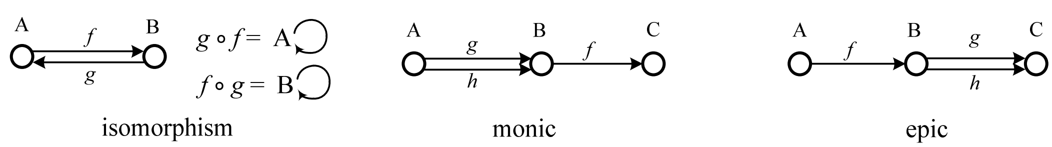


Figure 3.1 Isomorphism, monic and epic

In the category **Set** (objects are sets, morphisms are set functions), monic is the same as injective (one-to-one function), epic is the same as surjective (onto) and isomorphic is the same as bijective (one-to-one and onto). Note that in other types of categories a morphism may not be an isomorphism even if it is monic and epic.

SC is a *subcategory* of category **C** (with collection of objects $\mathrm{obj}_{\mathbf{C}}$) if for objects o_i, o_j in **SC** (collectively written as $\mathrm{obj}_{\mathbf{SC}}$) have

$\mathrm{obj}_{\mathbf{SC}} \subseteq \mathrm{obj}_{\mathbf{C}}$ and $\mathrm{Hom}_{\mathbf{SC}}(o_i, o_j) \subseteq \mathrm{Hom}_{\mathbf{C}}(o_i, o_j)$ $(\forall o_i, o_j \in \mathrm{obj}_{\mathbf{SC}})$

All of the objects and arrows in subcategory **SC** are to be found in the parent category **C**, the source and targets of arrows in **SC** and the same as those in **C**. Generally, subcategories only contain some of the objects and arrows of their parent categories. However, **SC** is a full subcategory of **C** if **SC** has the same arrows for each pair of objects as in **C**. Clearly any category is a full subcategory of itself.

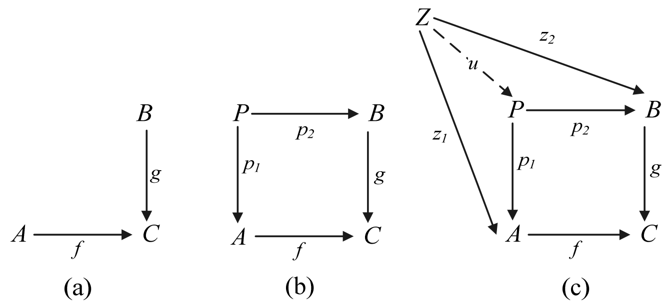


Figure 3.2 Arrows and pullbacks

A *pullback* of the pair of arrows f, g with cod(f) = cod(g) as shown in figure 3.2.a is an object P and a pair of arrows p_1 and p_2 as shown in figure 3.2.b such that $f \circ p_1 = g \circ p_2$. And if z_1: $Z \rightarrow A$ and z_2: $Z \rightarrow B$ are such that $f \circ z_1 = g \circ z_2$, then there exists a unique u: $Z \rightarrow P$ with $z_1 = p_1 \circ u$ and $z_2 = p_2 \circ u$ (as shown in figure 3.2.c).

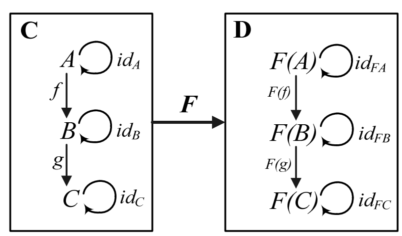


Figure 3.3 Functor

An arrow between categories **C** and **D** is termed a *functor* (as indicated in figure 3.3) if it satisfies some structure-preserving requirements:

(1) For each arrow f: $A \rightarrow B$ in **C**, there is an arrow $F(f)$: $F(A) \rightarrow F(B)$ in **D**.

(2) For each object A in **C**, the equation $F(id_A)=id_{FA}$ holds in **D**.

(3) For each pair of arrows $A \xrightarrow{f} B \xrightarrow{g} C$ in **C**, the equation $F(g \circ f)=F(g) \circ F(f)$ holds in **D**.

This type of arrow provides the facility for transforming from one type category to another category type. *Functors* are therefore basically structure-preserving morphisms from a source category to a target category. An obvious case is when the shape of the target category is determined by the functor, that is it accommodates all assignments from the source category and has no other structure of its own. However, one of the major features of functors is that it connects two different mathematical structures by structure-preserving mapping. One particular example is a forgetful functor which is defined from a category of algebraic structures (group or vector spaces) to the category of sets. The forgetful functor forgets the arrows, remembering only the underlying set and regardless of their algebraic properties.

A *natural transformation* is a mapping of one functor to another functor. If F and E are functors between the categories **C** and **D**, as shown in figure 3.4, a natural transformation η from F to E associates to every object X in **C** a morphism η_X: $F(X) \rightarrow E(X)$ between objects of **D**, called the component of η at X, such that for every morphism f: $X \rightarrow Y$ in **C** we have:

$\eta_Y \circ F(\mathrm{f}) = E(f) \circ \eta_X$

$$\begin{tikzcd} F(X) \arrow[r, "F(f)"] \arrow[d, "\eta_X"'] & F(Y) \arrow[d, "\eta_Y"] \\ E(X) \arrow[r, "E(f)"] & E(Y) \end{tikzcd}$$

Figure 3.4 Natural transformation

The basic understanding of category theory is that a category consists of objects and morphisms between the objects within the category, functors as morphisms between categories, and natural transformation as morphisms between functors. But there are also morphisms directly between objects in different categories. These cross-category object morphisms are called heteromorphisms (Ellerman, 2005). The theory of *adjoint* shows that all adjunctions arise from the representations of heteromorphisms between the objects of different categories. If there are two functors F: **C**→**D** and G: **D** → **C** between categories **C** and **D**. Then F and G are said to be a pair of adjoint functors or an adjunction, if for any X in **C** and Y in **D**, there is an isomorphism η natural transformation in X and in Y:

$$\eta_{X,Y}: Hom_{\mathbf{D}}(F(X), Y) \cong Hom_{\mathbf{C}}(X, G(Y))$$

The functor F on the left is called the *left adjoint*, and the functor G is the *right adjoint*. Both the maps that appear in the adjunction isomorphism, $F(X)\rightarrow Y$ and $X\rightarrow G(Y)$, go from the 'X-thing' (i.e., either X or the image $F(X)$) to the 'Y-thing' (either the image $G(Y)$ or Y itself), so we see a direction emerging from **C** to **D**. That direction of an adjunction is the direction of the left adjoint (which goes from **C** to **D**). Then **C** might be called the sending category and **D** the receiving category.

3.2 The categorical model for surface texture

The knowledge of surface texture includes massive diverse concepts and structures which cover specification definitions, definition categories, semantic understanding, algebraic structures, structured entities and relationships between all of them. The range of knowledge covers mechanical design, manufacturing information, surface metrology and information technology. The diverse nature of the knowledge makes it hard to apply in computing science. Using the categorical constructions introduced so far, a categorical model is constructed to capture the semantics of surface texture. The minimum objectives for the categorical model are:

1. A clear separation between intension[16] and extension[16] structures from design, manufacture and measurement in AST and PST.
2. Encapsulation[17] of objects, categories and subcategories. This includes the abstractions in the standard information system and includes inheritance and compositions such as aggregation, classification and association.
3. Manipulation of relationships, such as pullbacks, categories pullbacks and functors.
4. A query language to provide results.
5. A multilevel architecture with internal structures, high-level schema and the rigorous mapping between them.

[16] Intension structures will be represented by objects in a category, and extension structures will be represented by categories.

[17] Encapsulation - to encase the objects to the associated categories; to associate the subcategories with related categories, etc.

3.2.1 Categories, objects and arrows

Based on the characteristics of category theory, we can use categories to express all different kinds of structures in surface texture, and objects and arrows in a category to describe internal structures and relationships between elements respectively. Category theory ignores the unnecessary details of different definitions and structures and focuses on the categories and relationships between and in them.

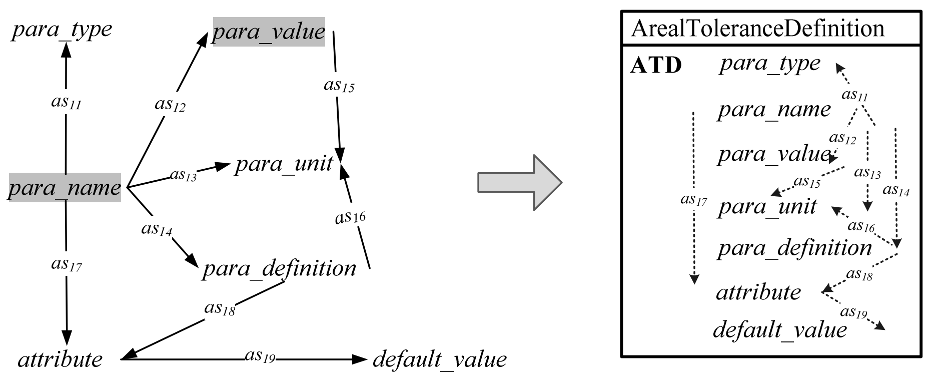


Figure 3.5 Objects and arrows in a category **ATD** (Areal Tolerance Definition)

The separation strategy between intension and extension structures from DMMs is designed according to the philosophy of GPS. The extension structures in surface texture are derived from the general GPS matrix which consists of individual chains of standards related to specific controls along the design and verification phases in the product development process.

The structures in surface texture can be determined by the definitions in different stages of specification or verification. To give an example, the tolerance definition from chain link 1 for the AST (see table 2.1) concerning the definition of areal parameters, and related terms such as the type of parameter, the unit of parameter, the limit value of parameter, the attribute and default value of the parameter. Then, all of the tolerance definitions are designed as the objects in the category named Areal (surface texture) Tolerance Definition, written as **ATD**. It is composed of seven objects (as indicated in figure 3.5):

> *para_type*: the type of the parameter, such as height parameters, spatial parameters and feature parameters in areal surface texture indicated in table 3.1,
>
> *para_name*: the name of the defined parameter, e.g. *Sq*, *Sal*, *Str*, *Vvv*, *Spd* etc.,

para_value: the assigned limit value for the parameter,

para_unit: the unit of parameter,

para_definition: the definition of parameter,

attribute: the attribute of parameter,

default_value: the default attribute value for parameter,

and nine arrows:

as_{11}: *para_name* → *para_type*,

The arrow as_{11} states every parameter belongs to a parameter type, for example the parameter *Str* (texture aspect ratio) is classified by spatial parameters as listed in table 3.1. Arrow as_{11} is epic as all parameters are defined into different types e.g. height, spatial, feature parameter, etc.

as_{12}: *para_name* → *para_value*,

The arrow as_{12} represents the parameter value is decided by the parameter name. For instance, for a specified honing surface, the parameter value of parameter *Sal* (auto-correlation length) can be 0.06mm, and parameter *Sa* of 0.728µm.

as_{13}: *para_name* → *para_unit*,

The arrow as_{13} shows that every parameter has a related unit.

as_{14}: *para_name* → *para_definition*,

The arrow as_{14}:*para_name*→*para_definition* express that every parameter has a unique parameter definition and then as_{14} is isomorphism.

as_{15}: *para_value* → *para_unit*,

The arrow as_{15} denotes that every parameter value should include a unit.

as_{16}: *para_definition* → *para_unit*,

The arrow as_{16} indicates that the parameter definition determines the type of parameter unit.

as_{17}: *para_name* → *attribute*,

The arrow as_{17} means some parameters have an attribute. For instance, the attribute of parameter *Str* is the fastest/slowest decays to *s* (with $0 \leq s < 1$).

as_{18}: *para_definition → attribute*,

The arrow as_{18} presents that it is the definition of parameter which determines the attribute.

as_{19}: *attbribute → default_value.*

The arrow as_{19} denotes that every attribute has a default value (1:N relationship). For example, the default value of *s* which is the attribute of parameter *Str* is 0.2.

Table 3.1 Data examples for characteristic of areal surface texture parameters (ISO 25178-3, 2010)

Parameter type	*Param-eter*	*Parameter name*	*Default unit*	*Attribute*	*Default value*
Height parameters	*Sq*	root mean square height	µm	-	-
	Ssk	skewness	Unitless	-	-
	Sa	arithmetical mean height	µm	-	-
Spatial parameters	*Sal*	autocorrelation length	µm	fastest decay to a specified values s, with $0 \leq s \leq 1$	*s*=0.2
	Str	texture aspect ratio	Unitless	fastest & slowest decay to s, with $0 \leq s \leq 1$	*s*=0.2
Functions and related parameters	*Vvv*	dale void volume	ml/m^2	material ratio *p*	*p*=80%
	Vvc	core void volume	ml/m^2	material ratios *p* and *q*	*p*=10%, *q*=80%
	Vmp	peak material volume	ml/m^2	material ratio *p*	*p*=10%
	Vmc	core material volume	ml/m^2	material ratio *p* and *q*	*p*=10%, *q*=80%
	Sxp	peak extreme height	µm	material ratio *p* and *q*	*p*=2.5%, *q*=50%
Feature parameters	*Spd*	density of peaks	$1/mm^2$	Wolfprune[18] Nesting Index *X*%	*X*%=5%
	Spc	arithmetic mean peak curvature	1/mm	Wolfprune Nesting Index *X*%	*X*%=5%
	S5p	five-point peak height	µm	Wolfprune Nesting Index *X*%	*X*%=5%
	S5v	five-point pit height	µm	Wolfprune Nesting Index *X*%	*X*%=5%

3.2.2 Families of categories

The inheritances of categories in the categorical model are in accordance with the philosophy of GPS. The definitions and terms defined in GPS determine the family tree and relationships between them. To give an example, figure 3.6 shows the category **AFC** (Areal Feature Characteristic) representing the feature characteristic as a family of *partition* objects **AP** (Areal Partition), *extraction* objects **AE** (Areal Extraction) and filtration objects **AF** (Areal Filtration).

[18] The term 'Wolfprune' presents Wolf's pruning method (Wolf, 1991) which consists of finding the peaks or pit with the smallest height difference and combining it with the adjacent saddle point on the change tree. The details of Wolf pruning method are presented in ISO 25178-2:2012.

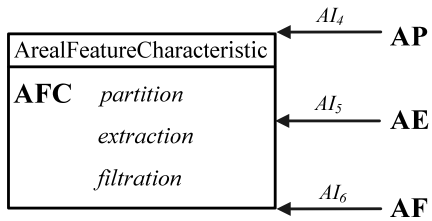


Figure 3.6 Category **AFC** and inherited categories **AP**, **AE** and **AF**

The inheritances of categories actually are adjoints. To given an example, in figure 3.6, category **AP** is inherited from object *partition* in category **AFC**. Subcategory **AFCP** with only one object *partition* is from category **AFC**. There are two functors between **AFCP** and **AP** which are F: **AFCP** → **AP** and G: **AP** → **AFCP**. Functor F denotes category **AFCP** is the family of category **AP**, the object partition is the family of all the objects in category **AP**. Functor G express that category **AP** is derived from category **AFCP**, and all of the objects in category **AP** belong to the only object partition in category **AFCP**. Given P_i $(0 < i \leq n) \in Obj_{\mathbf{AP}}$, if for *partition* in **AFCP** and any P_i in **AP**, there is an isomorphism η natural transformation in object *partition* and in P_i:

$$\eta_{partition,Pi}: Hom_{\mathbf{AP}}\ (F(partition),\ P_i) \cong Hom_{\mathbf{AFCP}}(partition,\ G(P_i))$$

where $G(P_i) = partition$ and $Hom_{\mathbf{AFCP}}(partition,\ G(P_i)) = Hom_{\mathbf{AFCP}}(id_{partition})$.

3.2.3 Relationships

The relationships in the categorical model are represented by pullbacks and functors as described in section 3.1.

3.2.3.1 Pullbacks

To give an example, consider the pullback of S and E over P shown in figure 3.7, where S and E are objects in the categories for parameter name and parameter value respectively and P is the transmission band[19] in PST.

[19] A pair of cut-off to obtain required surface characteristics in PST, i.e. roughness, waviness and primary.

$$\begin{tikzcd}
 & S \arrow[dr, "\lambda_1"] & \\
S \times_p E \arrow[ur, "\pi_1"] \arrow[dr, "\pi_2"'] & & P \\
 & E \arrow[ur, "\lambda_2"'] &
\end{tikzcd}$$

Figure 3.7 Pullback of *S* and *E* over *P*

$S \times {}_pE$ is the subproduct of *S* and *E* over *P*. It represents the subset of the universal product $S{\times}E$ that actually occurs for the relationship *P* which represents all instances of this type of association between parameter name and value. Instances of *P* are of the form $\{<s, e, p> \mid \lambda_1(s) = \lambda_2(e), s{\in}S, e{\in}E, p{\in}P\}$ where *p* is information such as lower nesting index and upper nesting index of the transmission band and is an element in the powerset of *P*.

The arrow π_1 is a projection of the subproduct $S{\times}E$ over *S* representing all parameter names.

- If π_1 is epic then every parameter name appears at least once in the subproduct. Thus every parameter name participates in the relationship and the membership object of *S* is indicated as mandatory. If, however, π_1 is not epic, then not every parameter name participates in the relationship and the membership object of *S* is indicated as optional.
- If π_1 is monic then each parameter name appears just once in the subproduct. If, however, π_1 is not monic, then a parameter name may participate more than once in the relationship.
- If π_1 is isomorphic then each parameter name appears once in the subproduct and *S* has mandatory participation in the relationship.

The arrow π_2 is a projection of the subproduct $S{\times}E$ over *E* representing all parameter values.

The normal understanding of parameter name and value data would be either monic or epic. It is because different parameter names have different series of related values, the parameter value is selected to match the parameter name, and one parameter name can only have one related value at a time.

The arrow λ_1 which maps from S to P represents associations between parameter name and transmission band.

The arrow λ_2 which maps from E to P represents associations between parameter value and transmission band.

When $\lambda_1(s) = \lambda_2(e)$, there is an intersection between the two associations, that is a parameter name and parameter value both point at the same transmission band: a set of such transmission band values is associated with a particular parameter name-value pair.

Table 3.2 represents some examples of pullback of S and E over P.

Table 3.2 Examples of the relationship between S and E over P

S parameter name	E parameter value (μm)	P transmission band (mm)
Ra	0.008	0.0025 - 0.08
Ra	0.1	0.0025-0.25
Ra	1.2	0.0025-0.8
Rz	0.1	0.0025 - 0.08
Rz	0.4	0.0025-0.25
Rz	3.2	0.0025-0.8
RSm	0.13	0.0025-0.25
RSm	0.4	0.0025-0.8
RSm	1.3	0.008-2.5

3.2.3.2 Categories Pullbacks

Pullbacks normally appear between objects in the same category. However, there are numbers of relationships between objects in different categories which appear not as functors but more like pullbacks between different categories. This type of relationships is denoted ‘categories pullbacks’ in this thesis.

Figure 3.8 gives an example of categories pullback AP_4 - the deduction of **AE**-objects *max_sampling_distance* and *max_sphere_radius*.

Category **AP** (Areal Partition) represents the partition operation in specification. There are four objects in this category:

- The arrow as_{20} as $\hom_{\mathbf{AP}}$(*manufacturing_process*, *manufacturing_type*) is epic which states that every manufacturing process belongs to a kind of manufacturing type such as MRR (material removal required process) type or NMR (no material removed process) type;

- The arrow as_{21} as $\hom_{\mathbf{AP}}$(*manufacturing_process*, *surface_texture_lay*) means every manufacturing process will generate different indication types of surface lay such as '=', 'X' and 'C' (ISO 1302, 2002)(1:N relationship).

Category **AE** (Areal Extraction) represents the extraction operation in specification. Five objects are involved:

- The arrow as_{22} as $\hom_{\mathbf{AE}}$(*sampling_length*, *evaluation_area*) is isomorphism which expresses that the evaluation area can be calculated according to the sampling length;
- The arrow as_{23} as $\hom_{\mathbf{AE}}$(*max_sphere_radius*, *max_sampling_distance*) is isomorphism which means that the value of max sphere radius determines the value of max sampling distance for mechanical surfaces;
- The arrow as_{24} as $\hom_{\mathbf{AE}}$(*max_lateral_period_limit*, *max_sampling_distance*) is isomorphism which means that the value of max lateral period limit decides the value of max sampling distance for optical surfaces.

Category **ANI** (Areal Nesting Indices) inherited from a Category presents the filtration operation in specification. Four **ANI**-objects present the nesting index for different filters. The arrow as_{27}, as_{28} and as_{29} means that the ratio between nesting index for S filter and F operation, or S filter and L filter is the bandwidth ratio.

The product of object *surface_type* in category **AP** and object *S_filter* in category **ANI** determines **AE**-objects *max_sampling_distance* and *max_sphere_radius*. In the pullback structure, the objects *surface_type* and *S_filter* from the product of categories **AP** and **ANI** constitute a subcategory **SAA**. Since $\pi_1 p_4 \circ \lambda_1 p_4 = \pi_2 p_4 \circ \lambda_2 p_4$, ($\mathbf{SAA}\times\mathbf{AE}$, $\pi_1 p_4$, $\pi_2 p_4$) is the pullback of (AP_4 (…), $\lambda_1 p_4$, $\lambda_2 p_4$). Here, AP_4 (…) is a category with only one object and one identity arrow. Data examples of AP_4 are shown in Table 3.3. For example, if the nesting index of S filter is 0.1 µm for a mechanical surface, the max sampling distance and max sphere radius are 0.02 µm and 0.07µm respectively when a stylus instrument is applied. For an optical surface with the same S filter, they are 0.03 and 0.1 µm respectively.

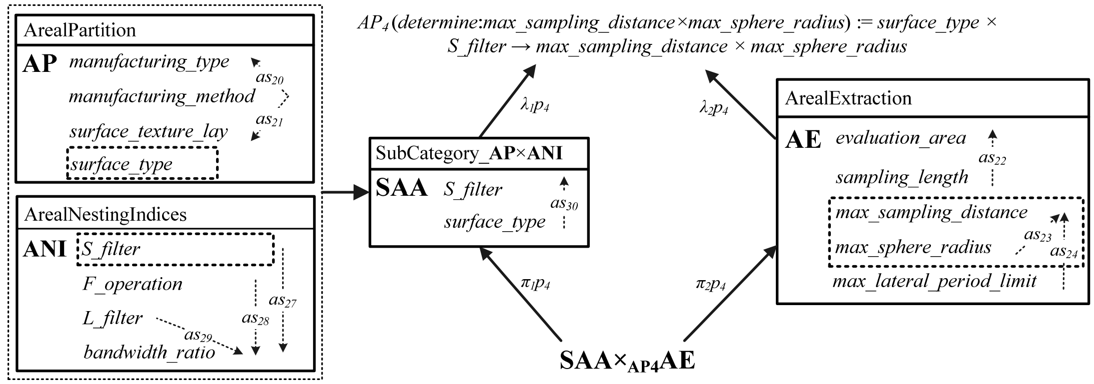


Figure 3.8 An example of categories pullback AP_4 - the determination process of **AE**-objects

Table 3.3 Data examples of pullback AP_4

AP	**ANI**	**AE**	
surface_type	*S_filter* (μm)	*max_sampling_distance*(μm)	*max_sphere_radius*(μm)
Mechanical surface	0.1	0.02	0.07
Optical surface	0.1	0.03	0.1
Mechanical surface	2.5	0.5	2
Optical surface	2.5	0.8	2.5

3.2.3.3 Functors – mapping from specification to verification

In this thesis, functors are utilised to reveal the structure-preserving mapping between categories in specification and verification. In figure 3.9, AF_1:**ATD→ATS** is the functor between categories **ATD** (Areal Tolerance Definition) and **ATS** (Areal Tolerance Specification). **ATD** is one of the categories in specification and **ATS** is one of the categories in verification. Thus, functor AF_1 is one of the mappings between specification and verification. According to the definition of functors, for each object and arrow in category **ATD**, there is a mapped object and arrow in category **ATS**. Therefore, for **ATD**-objects *para_value* and *para_name*, there are AF_1 (*para_value*), and AF_1 (*para_name*) in **ATS**-objects, and AF_1 (*para_value*) = *limit_value*, AF_1 (*para_name*) = *para_name* in **ATS**-objects. Similarly, for **ATD**-arrows as_{11} and as_{12}, there are $AF_1(as_{11})$, and $AF_1(as_{12})$ in **ATS**-arrows, and $AF_1(as_{11})=av_1$, $AF_1(as_{12})=av_2$. The functor AF_1 here is a covariant functor which preserves the directions of arrows, i.e., every arrow as_i:$A \rightarrow B$ is mapped to an arrow $F(as_i)$: $F(A) \rightarrow F(B)$. Here, the **ATD**-objects in specification and **ATS**-objects in verification are independent, but they are however related by the so called 'Duality

Principle' in GPS (as discussed in the last chapter). For example, if the object *para_value* in **ATD** is the limit value for the assigned parameter in specification, the object *limit_value* in **ATS** will be the same limit value when the specification is interpreted to verification.

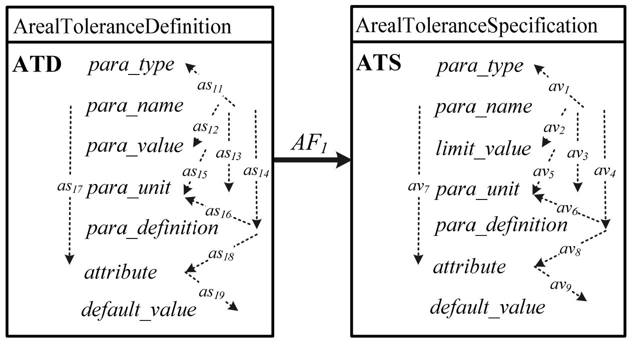


Figure 3.9 The functor between category **ATD** and **ATS**

3.2.4 Manipulation

Obtaining an output from a database is not easy for some object-based systems as the output is a subset of variables in an object without any consideration of the arrows which are an equally important part of the data. This difficulty is readily handled in a formal manner by arrows, pullbacks and functors which provide the basis for a query mechanism in a natural manner.

The query language developed in the thesis is therefore based on arrows, pullbacks and functors. The arrows in a category can produce the result of a co-domain object when the domain object is known. The pullbacks in a category can produce the result of objects for multiple relationships; and the pullbacks for different categories can produce a new category. The functors produce the output categories or subcategories if the input is known. The query output on a category will therefore be another category complete with arrows and objects. The output category could contain structured values not present in the source category and assigned by another functor. Hence, it is possible to create complex categories via manipulating values from a number of categories. Alternatively, a forgetful functor (as mentioned in section 3.1) applied to a category could also be used.

An example of a query is given below. As shown in figure 3.10, three categories are defined:

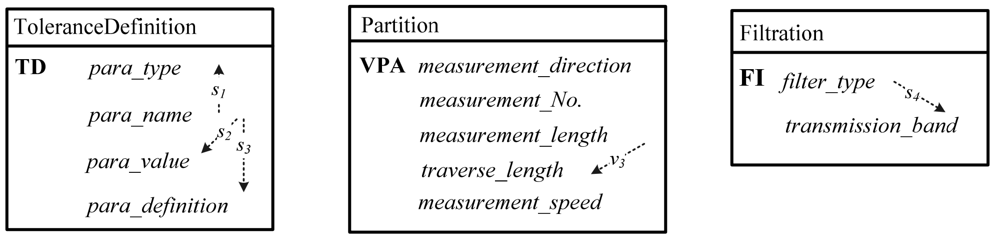


Figure 3.10 Categories **TD**, **VPA** and **FI** in PST

- **TD** (Tolerance Definition) is the category for the tolerance definition in the specification of PST

  Arrows:

  s_1: *para_name → para_type*

  s_2: *para_name → para_value*

  s_3: *para_name → para_definition*
- **VPA** (Verification Partition) is the category for the partition operation in the verification of PST

  Arrow:

  v_3: *measurement_length→ traverse_length*
- **FI** (Filtration) is the category for the filtration operation in the specification of PST

  Arrow:

  S_4: *filter_type→ transmission_band*

The natural language query is "*When the specified parameter is Ra with 0.2µm limit value, what are the related transmission band and measurement length?*"

The series of functors and pullbacks are given below.

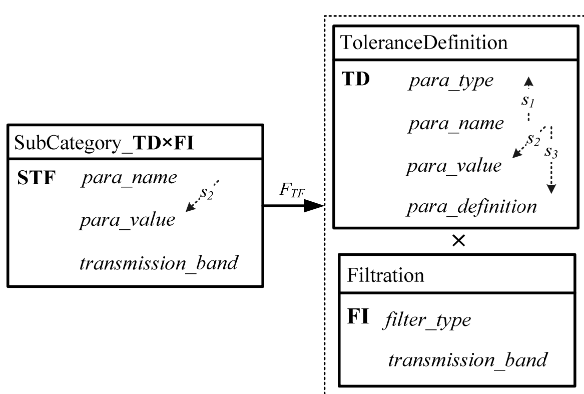


Figure 3.11 Subcategory **STF** of category **TD** and **FI**

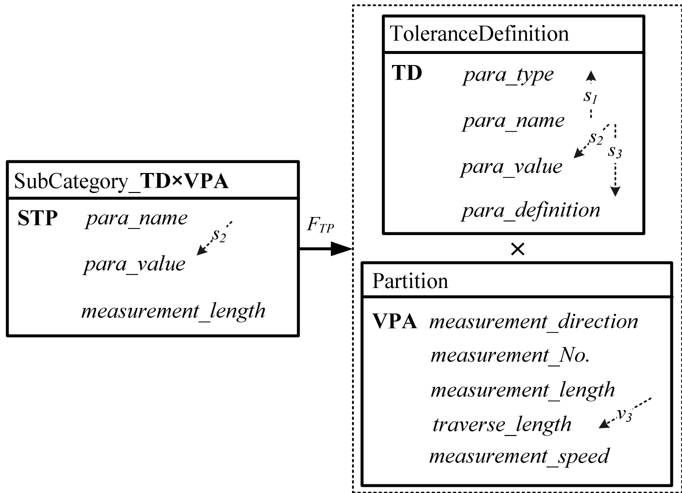


Figure 3.12 Subcategory **STP** of category **TD** and **VPA**

Two functorial operations are given.

- $\boldsymbol{F_{TF}}$: **STF** → **TD**×**FI** (Hom-set in **STF** = s_2; subobjects in **STF**=(*para_name*│*para_name* = 'Ra', *para_value*│*para_value* = 0.2μm, *transmission_band*))
- $\boldsymbol{F_{TP}}$: **STP** → **TD**×**VPA** (Hom-set in **STP** = s_2; subobjects in **STP**=(*para_name*│*para_name* = 'Ra', *para_value*│*para_value* = 0.2μm, *transmission_band*, *measurement_length*))

The first functor $\boldsymbol{F_{TF}}$ derives the subcategory **STF** from the composition of categories **TD** and **FI** (as shown in figure 3.11) to produce the subcategory **STF** with subobjects *para_name* of '*Ra*', *para_value* of 0.2μm and *transmission_band*.

The second functor $\boldsymbol{F_{TP}}$ derives the subcategory **STP** from the composition of categories **TD** and **VPA** (as shown in figure 3.12) to produce the subcategory **STP** with subobjects *para_name* of '*Ra*', *para_value* of 0.2μm and *measurement_length*.

The pullback presented in figure 3.14 produces the answer of the value for *measurement_length*.

Note that the strategy involves a selection of objects and related arrows. The selection of objects from different categories produces new subcategories. Results are produced according to the pullbacks in a same category. Category pullbacks can also be utilised to generate the required results in some cases.

$$
\begin{tikzcd}
 & \mathit{para\_name} \arrow[dr, "\pi_1"] & \\
\mathit{para\_name} \times_{\mathit{transmission\_band}} \mathit{para\_value} \arrow[ur, "\lambda_1"] \arrow[dr, "\lambda_2"'] & & \mathit{transmission\_band} \\
 & \mathit{para\_value} \arrow[ur, "\pi_2"'] &
\end{tikzcd}
$$

Figure 3.13 Pullback of *para_name* and *para_value* over *transmission_band*

The pullback presented in figure 3.13 produces the answer of the value for *transmission_band*.

$$
\begin{tikzcd}
 & \mathit{para\_name} \arrow[dr, "\pi_3"] & \\
\mathit{para\_name} \times_{\mathit{measurement\_length}} \mathit{para\_value} \arrow[ur, "\lambda_3"] \arrow[dr, "\lambda_4"'] & & \mathit{measurement\_length} \\
 & \mathit{para\_value} \arrow[ur, "\pi_4"'] &
\end{tikzcd}
$$

Figure 3.14 Pullback of *para_name* and *para_value* over *measurement_length*

3.3 Conclusions

Based on category theory, a categorical model is developed to model the knowledge concerning design, manufacture and measurement in AST and PST. A clear separation between intension and extension structures (objects and categories respectively) is presented. The established inheritance of the categories is according to the philosophy of GPS. The query language to manipulate the objects and categories is also developed. The categorical modelling mechanism will be utilised in knowledge modelling for PST and AST in the following two chapters.

4. Knowledge modelling for Profile Surface Texture (PST)

Using the categorical model established in the last chapter, this chapter sets out to model the knowledge in PST. The knowledge model of PST is divided into specification and verification and is presented in section 4.2 and 4.3 respectively.

4.1 Introduction

In the development of surface texture characterisation, more than 100 profile parameters and 40 areal parameters have been defined. The specification of surface texture is getting more complicated (as shown in figure 2.8). There is a large amount of surface texture specification and verification data with associated information regarding functional requirements, manufacturing process and measurement that needs to be expressed, transferred, stored or analysed. As more data is being collected, there is a need for sharing data and associated information effectively, to eliminate redundancy in data collection and analysis. Thus a complete and unambiguous expression of the surface texture for a connection between design, manufacture and measurement needs to be achieved.

According to the general GPS matrix, the expression of surface texture can incorporate two processes: specification and verification processes. As shown in figure 4.1, the left part and right part are specification and verification processes respectively. In order to make a clear expression of surface texture for designers and engineers, an unambiguous expression schema of PST is proposed. Based on the GPS philosophy, the PST knowledge in specification and verification will be structured by the categorical model in this chapter.

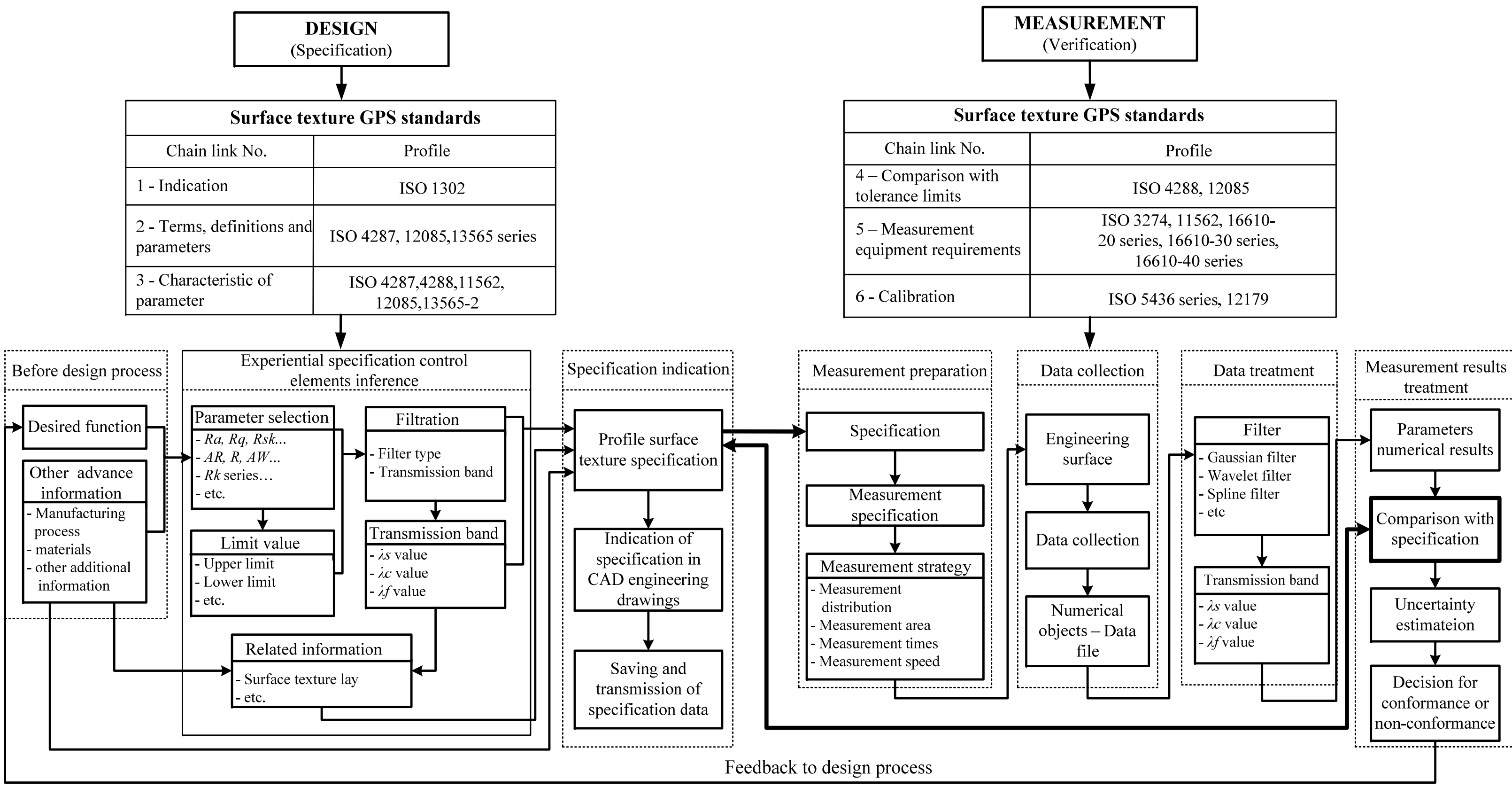
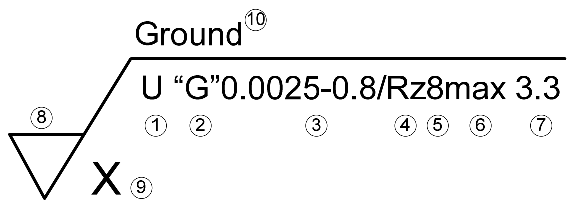


Figure 4.1 Scheme of general GPS matrix model in PST

4.2 Knowledge modelling for specification

4.2.1 The specification process of PST

The surface texture specification process is the design step where the field of permissible deviations of a set of control elements of surface texture is stated, accommodating the required functional performance of the workpiece. ISO 1302:2002 gives ten different control elements (see figure 4.2) which state as following:



① Indication of upper (U) or lower (L) specification limit
② Filter type
③ Transmission band
④ Profile parameter
⑤ Evaluation length as the number of sampling length
⑥ Comparison rule
⑦ Limit value in micrometres
⑧ Type of manufacturing process
⑨ Surface texture lay
⑩ Manufacturing process

Figure 4.2 Ten control elements in PST specification indication in ISO 1302:2002

① Indication of upper (U) or lower (L) specification limit: the surface texture requirement are indicated as a unilateral or bilateral tolerance.

② Filter type: the type of filter used to obtain required features of PST.

③ Transmission band: a pair of cut-off values to obtain required surface characteristics i.e. roughness, waviness and primary.

④ Profile parameter: profile parameters for roughness, waviness and primary parameters which are defined in ISO 4287 (1997), ISO 12085 (1996) and ISO 13565 series.

⑤ Evaluation length as a multiple of sampling length: default evaluation length is five times the sampling length.

⑥ Comparison rule: rules for comparison of the measured values with the tolerance limits.

⑦ Limit value in micrometers: the assigned limit value for the chosen profile parameter.

⑧ Type of manufacturing process: there are three types which are Material Removal Required Process (MRR), No Material Removed (NMR) and Any Process Allowed (APA).

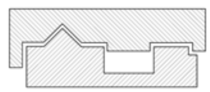
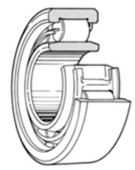
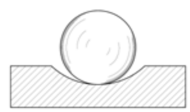
⑨ Surface texture lay: the surface lay and direction of the lay emanating from the manufacturing process such as traces left by tools.

⑩ Manufacturing process: the manufacturing process that produces the specified surface.

The purpose of the specification process is to establish those control elements associated with the design requirements of parts and their functional surfaces commensurate with production capabilities for use on design and engineering drawings.

In many applications surface texture is closely allied to function, for example in an instance where two surfaces are in close moving contact with each other their surface textures will affect their sealing or wear properties (as shown in table 4.1). This might suggest that it is a case of 'the smoother the better', but this is not always true as other factors may be involved. The financial impact of such decisions has to be considered: it costs a large amount of money to produce very smooth surfaces and the expense of this exercise can considerably add to the bill without gaining a great deal of performance. It can be seen that some thought must be given to surface texture at the design stage, with the designer specifying the texture required to give the correct performance. It follows that the production engineer must use the correct machine tools to obtain the required surface texture and advise the operator of the tolerances allowed. However, identifying very specific parameters of the surface texture with function is fraught with problems, usually because of time and expense.

Table 4.1 Examples of application categories requiring controlled surface texture (Curtis & Farago, 2007)

Functional objectives	Applications		Critical Characteristics of the surface texture
	Examples	Symbols	
Resistance to wear	Machine tool guideways		Surface texture limits the area available to carry the load, and causes increased wear rate, or may require run-in before operation at maximum capability is feasible.
Reduced vibration and noise	Antifriction bearing pathway in the direction of rolling		High frequency vibrations, can originate from closely spaced lobing which, by the standard terminology is classified as a component of surface texture when occurring within the selected cutoff width.
Preservation of an uninterrupted lubricant film	The track of a ball bearing ring		The peak of a rough surface will impede the continuity of the lubricant film which should prevent metal-to-metal contact.

The latest PST specification standard gives the tools to control the PST by a relatively unambiguous specification on technical drawings. The standard assists the designers to indicate the intended PST specification with the least possible effort, also making it possible for the reader of a given specification to understand, implement or verify the requirement without mistakes. Although the standard still has a certain specification uncertainty, the specification elements in them are considered to provide enough important information for manufacturers and metrologists. When all elements are specified in one specification, the symbol may appear much longer than traditional ones which mean more drawing space is needed. A simplified version or reference symbol can be applied but should be without any significant information loss (Qi et al., 2013).

The specifications of PST are assigned to transfer more manufacture and measurement information, based on the GPS requirements. In contrast to traditional tolerance systems, the design process of the specification is mapped to and receives feedback from the manufacture and measurement. Figure 4.3 shows an integrated specification model in PST. In the design phase, functional requirements and other factors such as manufacturing processes and component types should be considered for a function design of PST. All of the specification control elements defined in ISO

1302:2002 can be established according to the inputs and the inference of relationships. After the inference procedure, all of the inferred specification elements can be combined into a complete specification. Then the specification can be generated and saved by a CAD system to an indication in engineering drawing.

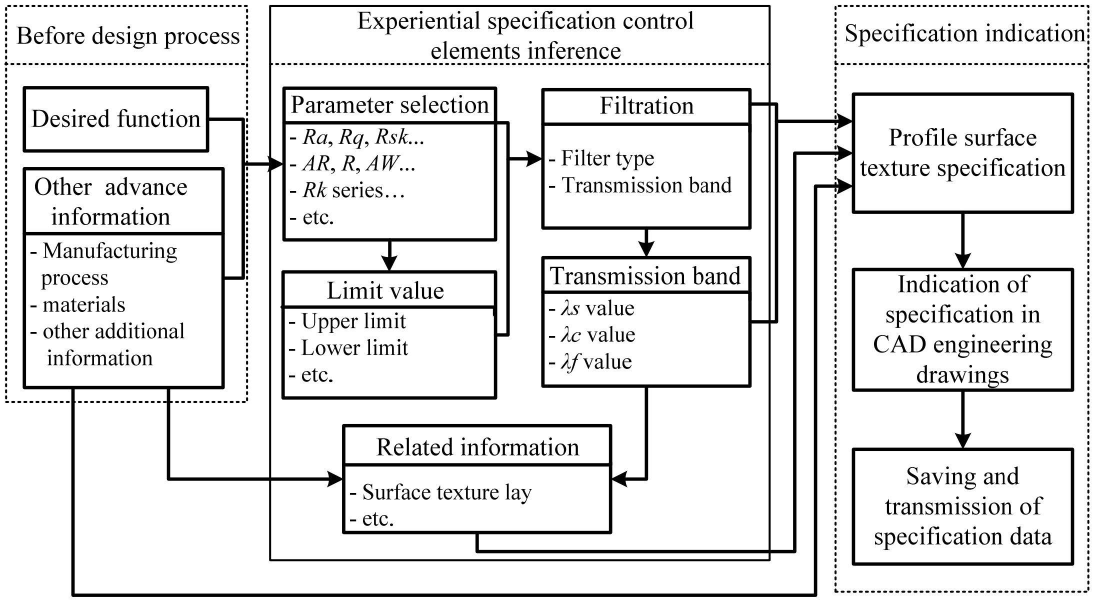


Figure 4.3 The specification process model in PST

As shown in Figure 4.4, the manufacturing processes of the specified surfaces can be determined by the functional requirements and/or the component types of the surfaces. When the manufacturing process is assigned, the design and manufacture cost can be estimated accordingly. All the information about functions, component types and manufacturing processes can be used to deduce the partial specification elements such as the parameters and related limit value. Utilising the categorical model, the complete ten specification elements for PST specification can be deduced. Then the related measurement requirements for the assigned specification can be inferred and the measurement cost can be estimated as well. The measurement cost then will be added to the total cost which can be used to balance the design and measurement details. For example, if the specified surface is one of the faces of a helical gear tooth (component type), and related functional requirement is the wear during gear meshing, the related manufacturing process can be grinding with profile parameter, *Ra* of 0.4µm. According to these partial specification elements, a series of complete surface texture specification elements can be determined. Then the related measurement information of the specification is deduced, which includes measurement length of 4.8mm, sampling length of 0.8mm, evaluation length of 4.0mm, transmission band of 0.0025-0.8mm, tip radius of 5μm, sampling spacing of 0.55μm, etc. In this model, the designer can access the measurement information, and then the measurement cost can be estimated and added to the total cost. As the complete specification can be generated by the categorical model according to the functional requirements, the specification design cost will be decreased. If the estimated manufacture and measurement cost increases, the specification can then be modified with a larger limit value which can still meet the functional requirements.

It should be noticed that although the specification should be designed in sufficient detail that any uncertainty is negligible in comparison with the functional requirements, it must be recognised that this may not be always practicable. The design may be incomplete because the definition of the PST parameter is ambiguous in some situations. Or it may imply conditions that can never be fully met and whose imperfect realisation is difficult to take into account. Currently, so-called 'complete' and 'unambiguous' expressions are an estimate of the probability of nearness to the best expression that is consistent with presently available knowledge. In addition, the extent of integrity is correlated to function and cost requirements, and extra integrity beyond these requirements is unnecessary and costly. It is important to find a way to satisfy the requirements by omitting other detail offset specifications (Qi, Jiang, Liu & Scott, 2010).

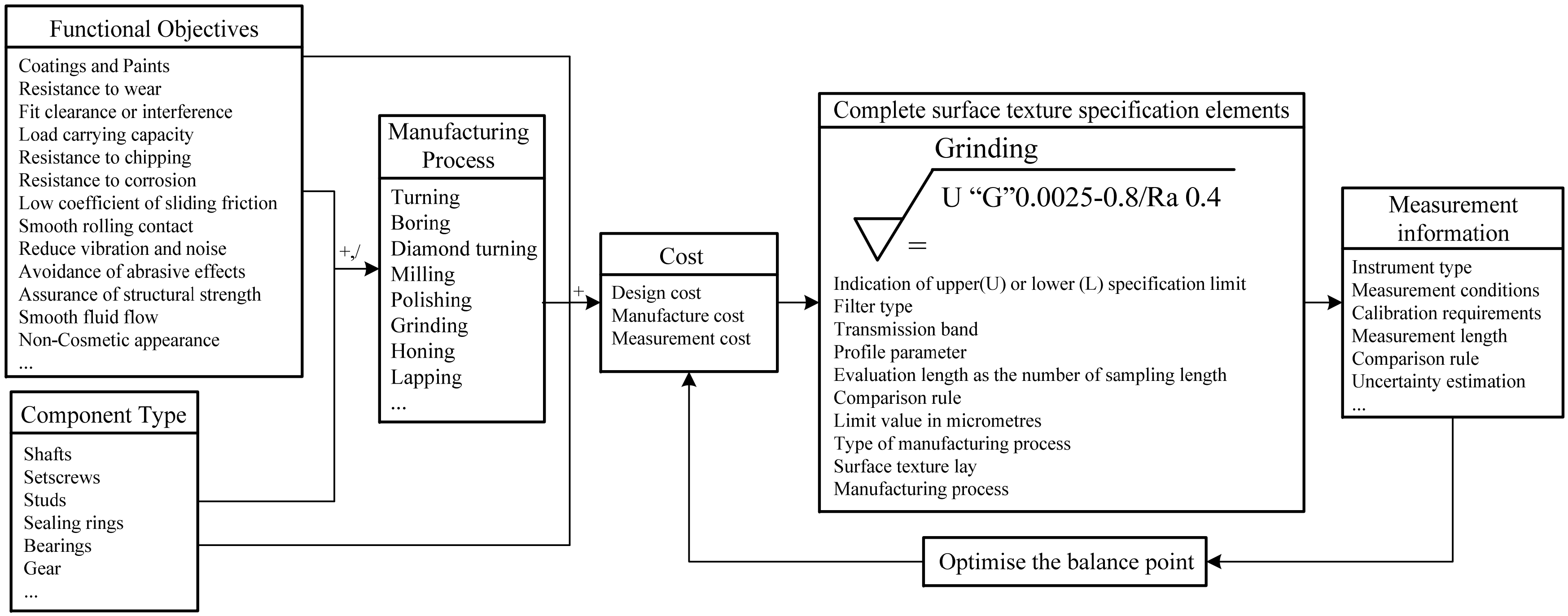


Figure 4.4 The design process for a complete and functional PST specification

4.2.2 The categorical model for PST specification

According to the specification model, a series of categories are structured in this section.

The category for ‘Input’ written as **IN** as shown in figure 4.5 includes the elements that designers need to input for completing the specification. There are three objects *surface_function*, *material* and *manufacturing_process* which denote the desired functions, the material of the specified surface and the manufacturing process that produce the specified surface respectively.

Input	
IN	*surface_function*
	material
	manufacturing_process

Figure 4.5 Category **IN** for input in PST specification

The category for ‘Codification’ written as **CO** as shown in figure 4.6 belongs to the chain link 1 which will determine the indication of the callout. The object *indication_type* indicates the three graphical symbols for APA (see Section 4.2.1), MRR and NMR. The object *specification_type* denotes the first control element in figure 4.2.

Codification	
CO	*indication_type*
	specification_type

Figure 4.6 Category **CO** for codification in PST specification

The category for ‘Tolerance Definition’ written as **TD** as shown in figure 4.7 belongs to the chain link 2 which is the definition of PST parameters and value. Four objects *para_type*, *para_name*, *para_value* and *para_definition* present the type, name, limit value and definition of the parameter respectively. There are three arrows between the four objects. The arrow s_1 states every parameter belongs to a parameter type, for example the parameter *Rsm* is classified by spacing parameters. The arrow s_2: *para_name* → *para_value* represents the parameter value that is decided by the parameter name. For example, *parameter_name RSm* has related *parameter_value* range such as 0.013-4μm. The arrow s_3: *para_name* → *para_definition* expresses that every parameter has a unique parameter definition.

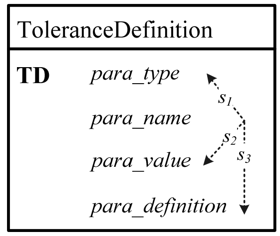


Figure 4.7 Category **TD** for tolerance definition in PST specification

The category for 'Feature Characteristic' written as **FC** is chain link 3 and is composed of three different feature operations which are Partition, Extraction and Filtration as shown in figure 4.8. Categories **PA** (Partition), **EX** (Extraction), and **FI** (Filtration) are inherited from three objects *partition*, *extraction* and *filtration* in category **FC**.

The category **PA** expresses the partition operation as described in chapter 2. There are three objects *manu_type*, *manu_process* and *surface_texture_lay* which present the type of manufacturing process, manufacturing process and surface texture lay respectively. The arrow s_4: *manu_process* → *manu_type* states every manufacturing process belongs to a kind of manufacturing type such as MRR type or NMR type. The arrow s_5: *manu_process* → *surface_texture_lay* means every manufacturing process will generate different indication types of surface lay such as '=', 'X' and 'C'.

The category **EX** represents the extraction operation in specification. Three objects are involved. The arrow s_6: *sampling_length* → *evaluation_length* expresses that evaluation length can be calculated according to the sampling length. For example, the default evaluation length is five times the sampling length. The arrow s_7: *num_cutoff* → *evaluation_length* states the number of the sampling length and can determine the value of evaluation length.

There are two **FI**-objects involved in the filtration operation in specification. The objects *filter_type* and *transmission_band* are the control elements ② and ③ respectively as shown in figure 4.2. Category **TB** for 'Transmission Band' is inherited from category **FI**. The objects *upper_limit* and *lower_limit* are the two components in the transmission band. The arrow s_8: *upper_limit*→ *lower_limit* states there are different stationary ratios between the upper and lower limit of the transmission band.

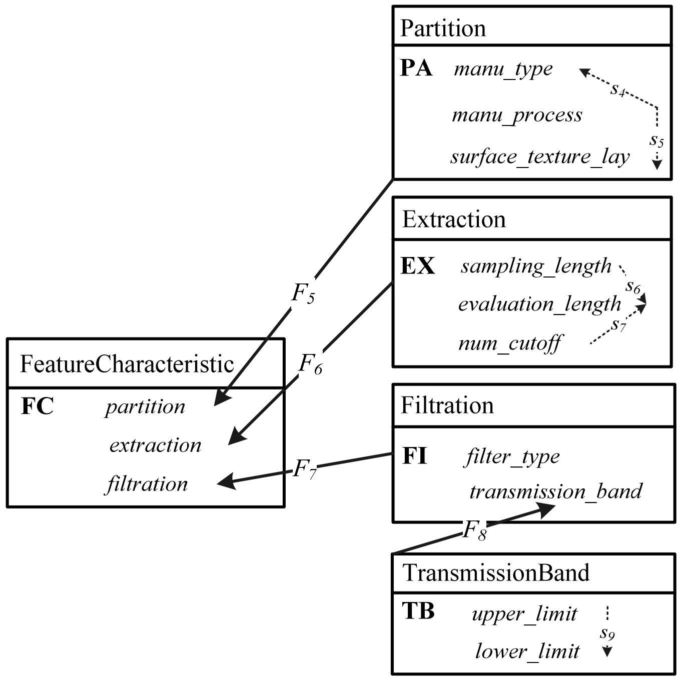


Figure 4.8 Category **FC** for feature characteristic and inherited categories **PA**, **EX**, **FI** and **TB** in PST specification

The Category **CP** (Comparison) states the comparison process in specification as shown in figure 4.9. The object *compa_type* is the control elements ⑥ in figure 4.2, whereas the object *compa_definition* is the definition of the comparison type. There are only two comparison types specified in PST, the '16%-rule' and the 'max-rule'. The default comparison rule in both ISO and ASME standards is the 16%-rule, but in a few company standards it is the max-rule. The comparison rule in the verification process determines whether the workpiece is accepted or rejected according to measurement results. Used as one of ten control elements in specification, the comparison rule must be specified in the specification process to reduce the specification uncertainty. The comparison rule is also an essential tool for the mapping between the specification and verification processes.

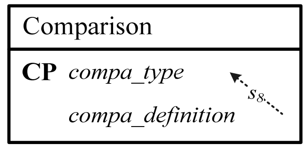


Figure 4.9 Category **CP** for comparison in PST specification

The objects in category **CA** for 'Callout' are the most important part for a PST specification design to be shown on the engineering drawing. The **CA**-objects are composed of 10 control elements. As shown in figure 4.10, these elements belong to four categories which are the chain links 1-3 in the general GPS matrix respectively and category **CP**.

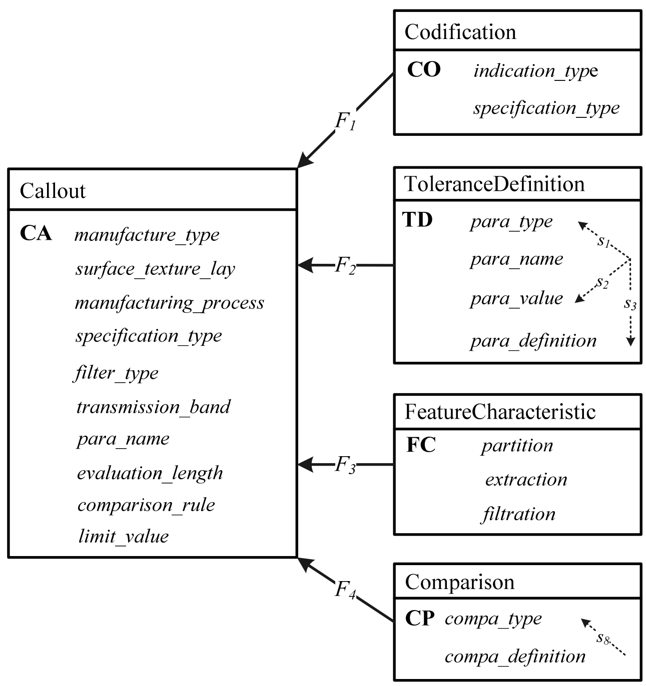


Figure 4.10 Category **CA** for callout in PST specification

In accordance with the categories structures stated above, the whole high-level abstract categorical model for PST specification is shown in figure 4.11. The relationships between different objects in the same category are represented by dashed line arrows with labelled s_i (i is an integer, the range of i is depend on the total number of arrows in all categories of specification). The dashed arrows R_j ($1 \leq j < 20$) represent the complicated relationships between objects in different categories. These relationships are expressed by pullbacks which will be described in the next section. The solid line arrows F_k show the direction of the inheritance.

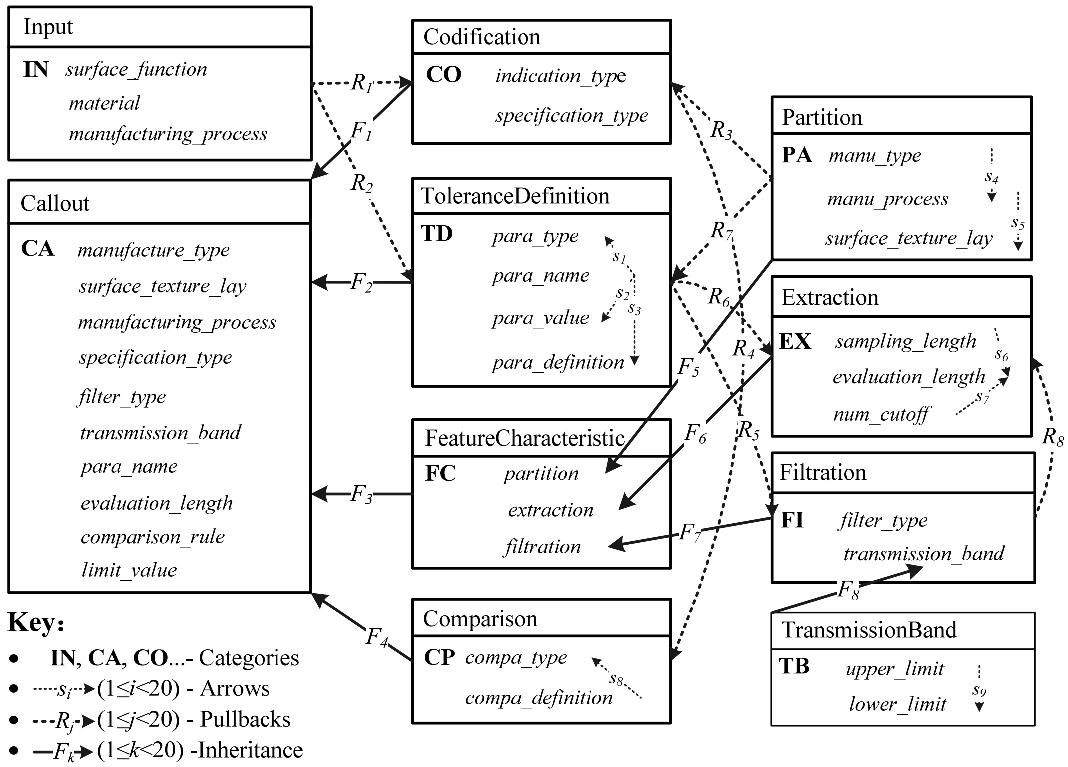


Figure 4.11 The high-level abstract categorical model diagram for PST specification

4.2.3 Relationships

There are eight pullback relationships in the PST specification model. The list of all the pullbacks is shown below:

R_1 - the relationship between objects in the categories **IN** (Input) and **CO** (Codification);

R_2 - the relationship between objects in the categories **IN** and **TD** (Tolerance Definition);

R_3 - the relationship between objects in the categories **PA** (Partition) and **CO**;

R_4 - the relationship between objects in the categories **CO** and **CP** (Comparison);

R_5 - the relationship between objects in the categories **TD** and **FI** (Filtration);

R_6 - the relationship between objects in the categories **TD** and **EX** (Extraction);

R_7 - the relationship between objects in the categories **PA** and **TD**;

R_8 - the relationship between objects in the categories **FI** and **EX**.

4.2.3.1 The Pullback R_1

A single R_j may express two or more relationships. These relationships can be regarded as refinements.

Figure 4.12 demonstrates the pullback R_1, where c_1 demonstrates the relationship between the categories **IN** (Input) and **CO** (Codification). It stores all the possible relations and extra information between objects of **IN** and **CO**. The expression

"*determine*: *indication_type* × *specification_type*:= **IN**-objects: *manufacturing_process…*"

is the name and type of the determination procedures. The notations $\pi_1 c_1$ and $\pi_2 c_1$ are projections of c_1 into the initial objects of **IN** and **CO** respectively, while $\lambda_1 c_1$ and $\lambda_2 c_1$ are represented as arrows injecting the initial instance objects into the pool of instances of this constraint relationship.

There are two different refinements of the c_1. Refinement $c_{1\text{-}1}$ expresses that the object *surface_function* in the category **IN** determines *specification_type* in the **CO** objects. Refinement $c_{1\text{-}2}$ presents the *indication_type* in the **CO**-objects is determined by *manufacturing_process* in the **IN** objects. Table 4.2 gives three examples of these relationships.

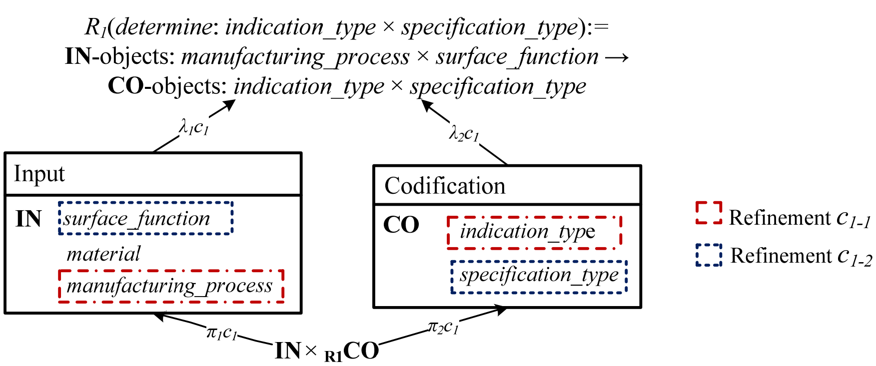
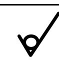
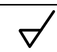


Figure 4.12 Pullback R_1 - the determination of **CO**-objects *indication_type* and *specification_type*

Table 4.2 Examples of relationships between objects in category **IN** and **CO**

Category **IN**		Category **CO**	
surface_function	*manufacturing_process*	*indication_type*	*specification_type*
Drive Shaft – sealing diameter for garter spring type oil seals	Polish		U
Sheet metal	Cold rolled		U
Bearing diameter	Turning		U and L

4.2.3.2 The Pullback R_2

R_i can also expresses multiple relationships. The pullback R_2 as shown in figure 4.13 is the relationships between categories **IN** (Input) and **TD** (Tolerance Definition). The combinations of all objects in category **IN** determine two objects *para_value* and *para_name* in category **TD**. In general meaning, the specified profile parameter and related value is determined by the desired surface function, the material of the surface and the manufacturing process that produced the specified surface. Table 4.3 gives three examples of these relationships.

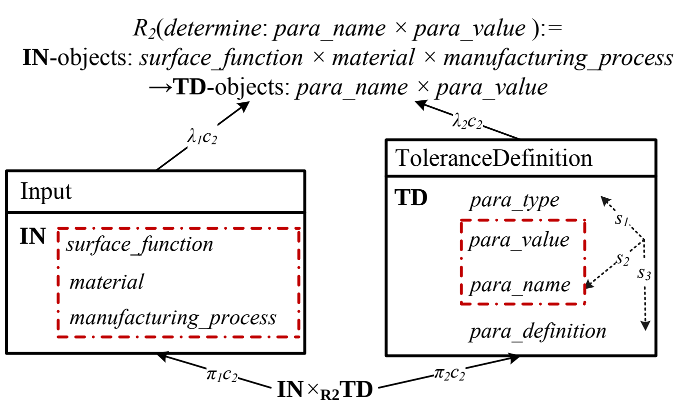


Figure 4.13 Pullback R_2 - the determination of **TD**-objects *para_value* and *para_name*

Table 4.3 Examples of relationships between objects in category **IN** and **TD**

Category **IN**			Category **TD**	
surface_function	*material*	*manufacturing_process*	*para_value* (μm)	*para_name*
Drive Shaft – undercuts	steel	turning	6.3	*Ra*
Sheet metal	alloy	cold rolled	2	*Ra*
Cylinder liner	cast iron	plateau honing	1.8	*Rvk*

4.2.3.3 The Pullback R_3

The pullback R_3 as shown in figure 4.14 is the relationship between two objects in categories **PA** (Partition) and **CO** (Codification). Similar to refinement $c_{1\text{-}1}$ in the pullback R_1, the *indication_type* in **CO**-objects is determined by the type of manufacturing process (*manu_type*).

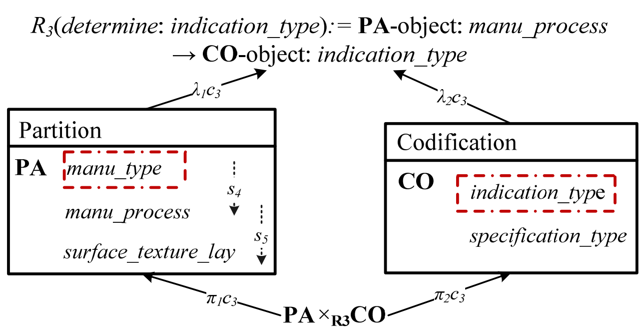


Figure 4.14 Pullback R_3 - the determination of **CO**-object *indication_type*

4.2.3.4 The Pullback R_4

The pullback R_4 as shown in figure 4.15 is the relationship between two objects in categories **CO** (Codification) and **CP** (Comparison). The object *compa_definition* in category **CP** is partially determined by the *specification_type* in category **CO**. To given an example, for the max-rule, when the *specification_type* is 'L', the related comparison definition will be "if the measured value is lower than the limit value, then it is accepted" and vice versa.

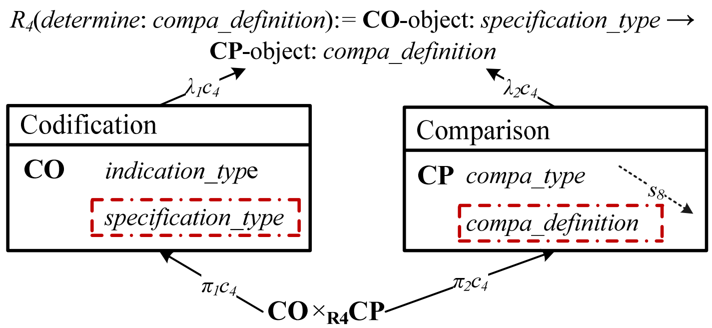


Figure 4.15 Pullback R_4 - the determination of **CP**-object *compa_definition*

4.2.3.5 The Pullback R_5

Figure 4.16 represents the pullback R_5. There are two different refinements of the c_5. Refinement $c_{5\text{-}1}$ expresses that the combination of *parameter_type*, *parameter_value* and *parameter_name* in the category **TD** (Tolerance Definition) determines *transmission_band* in the category **FI** (Filtration). Refinement $c_{5\text{-}2}$ presents the *filter_type* in the category **FI** is determined by *parameter_type* in the category **TD**. Table 4.4 gives three examples of these relationships. The table shows the transmission band of the Gaussian filter for profile spacing parameter *RSm* with value 0.04 µm is 0.0025(λs)-0.08mm (λc); the transmission band of the Gaussian filter for profile amplitude parameter *Ra* with value 0.8 µm is 0.0025(λs)-0.8mm (λc); the transmission band of the Motif filter for motif roughness parameter *R* with value 1.6 µm is 0.008 (λs)-0.5mm (*A*, see ISO 12085:1996).

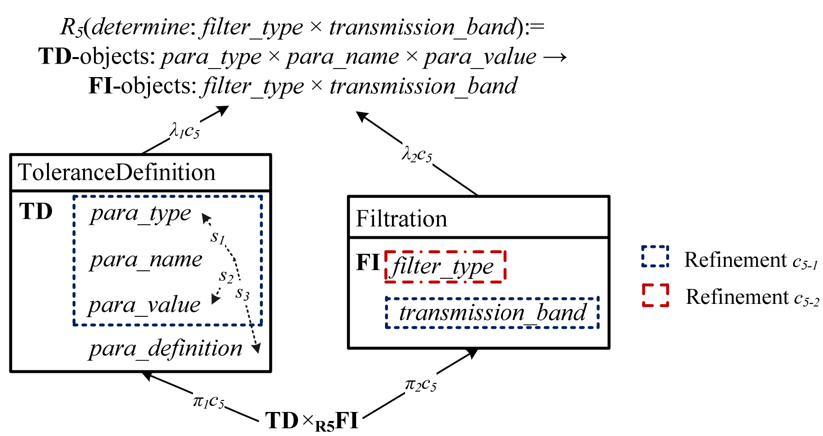


Figure 4.16 Pullback R_5 - the determination of **FI-**objects *filter_type* and *transmission_band*

Table 4.4 Examples of relationships between objects in category **TD** and **FI**

Category **TD**				Category **FI**	
parameter_type	*parameter_name*	*parameter_value*	*parameter_definition*	*filter_type*	*transmission_band*
Profile spacing parameters	*RSm*	0.04µm	Mean value of the profile element widths within a sampling length	Gaussian filter	0.0025-0.08mm ($\lambda s - \lambda c$)
Profile amplitude parameters	*Ra*	0.8µm	Arithmetical mean deviation of the assessed profile	Gaussian filter	0.0025-0.8mm ($\lambda s - \lambda c$)
Motif roughness parameter	*R*	1.6µm	Mean depth of roughness motifs	Motif filter	0.008-0.5mm ($\lambda s - A$)

4.2.3.6 The Pullback R_6

The pullback R_6 as shown in figure 4.17 represents the relationships between categories **TD** (tolerance Definition) and **EX** (Extraction). The combination of objects *para_value* and *para_name* in category **TD** determines two objects *sampling_length* and *evaluation_length* in category **EX**. Table 4.5 gives four related examples.

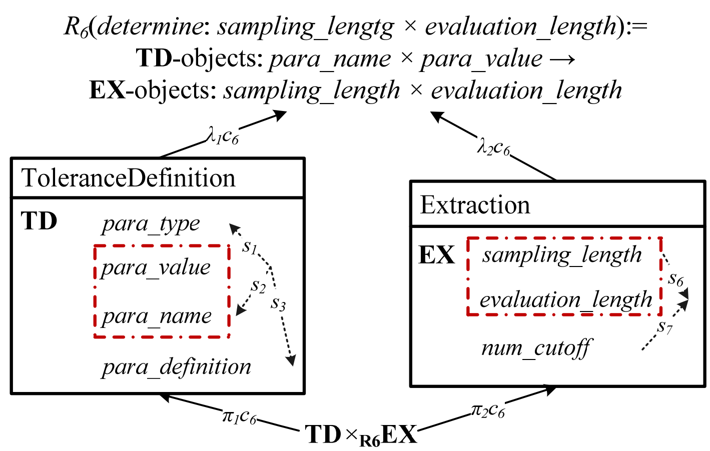


Figure 4.17 Pullback R_6 - the determination of **EX**-objects *sampling_length* and *evaluation_length*

Table 4.5 Examples of relationships between objects in category **TD** and **EX**

Category **TD**		Category **EX**	
para_value (µm)	*para_name*	*sampling_length* (mm)	*evaluation_length* (mm)
0.1	*Ra*	0.25	1.25
0.8	*Ra*	0.8	4
12	*Rz*	2.5	12.5
0.04	*Rsm*	0.08	0.4

4.2.3.7 The Pullback R_7

The pullback R_7 as shown in figure 4.18 is the restriction of objects *para_value* and *para_name* in categories **TD** (Tolerance Definition). The object *manu_process* in category **PA** (Partition) restricts two objects *para_value* and *para_name* in categories **TD**. In general meaning, every manufacturing process has a related range of profile parameter values as shown in table 4.6.

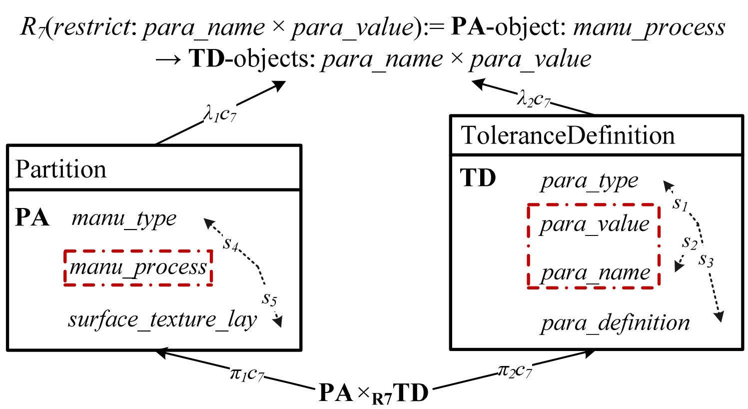
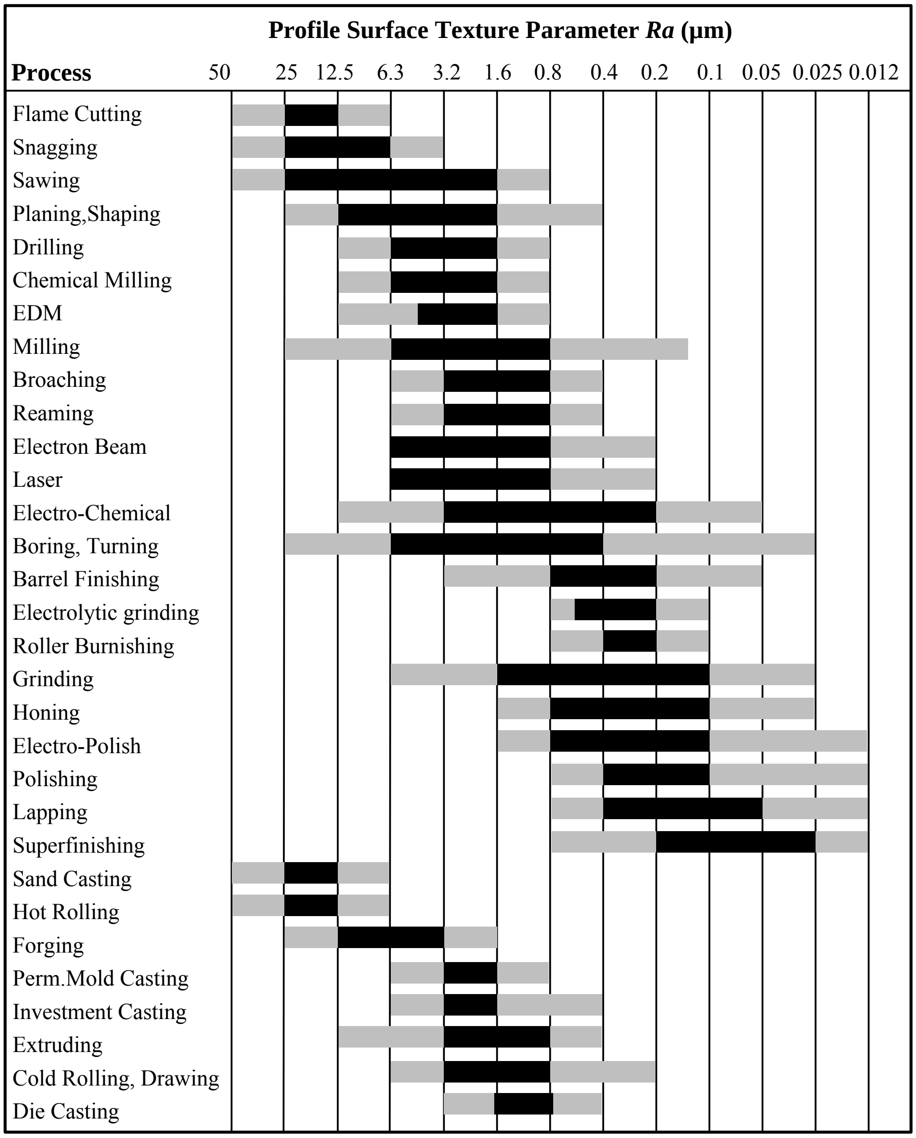


Figure 4.18 Pullback R_7 - the restriction of **TD**-objects *para_value* and *para_name*

Table 4.6 The value range of profile parameter *Ra* produced by common manufacturing processes (Hoffman, McCauley & Hussain, 2000)



Note: the ranges shown above are typical of the processes listed higher or lower values may be obtained under special conditions.

Key
average application
less frequent application

4.2.3.8 The Pullback R_8

The pullback R_8 as shown in figure 4.19 is the determination of object *sampling_length* in categories **EX** (Extraction). The object *transmission_band* in category **FI** (Filtration) is related with object *sampling_length* in category **EX**. For roughness parameters, the transmission band is composed with λs and λc, the value of λc is equal to the sampling length.

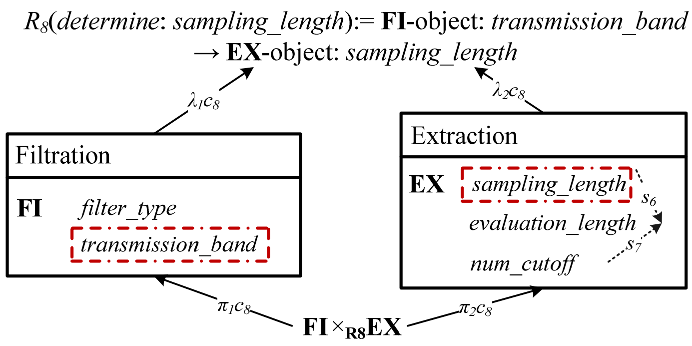


Figure 4.19 Pullback R_8 - the determination of **EX**-object *sampling_length*

4.3 Knowledge modelling for verification

4.3.1 The verification process of PST

The surface texture verification process takes place after the specification process. It assists manufacturing and inspection areas in the interpretation of drawing information and method of assessment, and explains the terms, symbols and values shown on drawings. It defines how surface texture specification data will be interpreted, and how a metrologist determines whether the surface of a workpiece conforms to the specification.

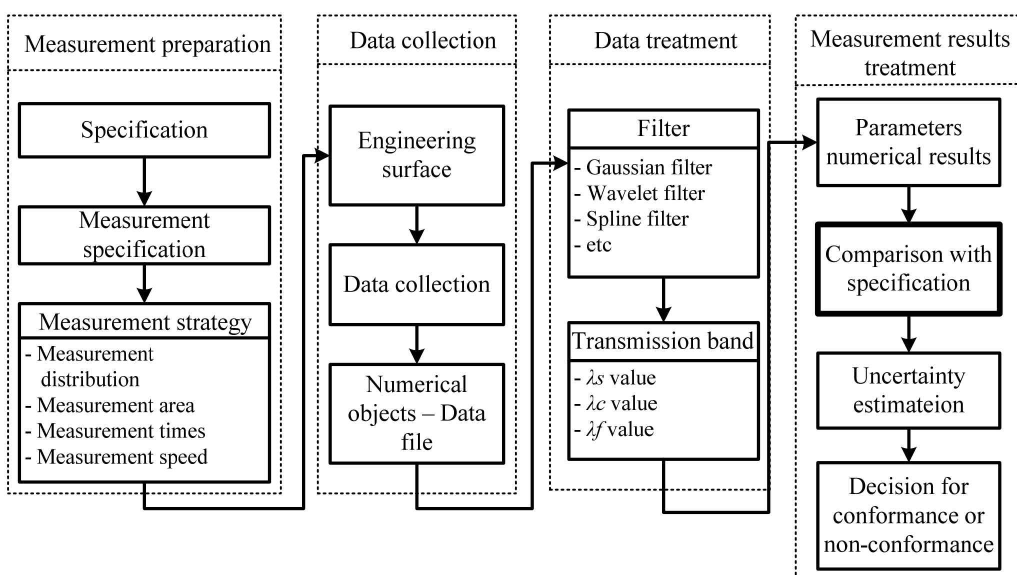


Figure 4.20 The verification process model in PST

As shown in figure 4.20, metrologists measure the surface texture and determine whether the surface is accepted according to the specification. Firstly, the metrologist analyses the specification, and translates it to a measurement specification which will take into account the measurement conditions. Following the measurement strategy, the metrologist carries out the measurement and obtains the measurement data. In this step, the metrologist selects different options for the form removal and filtration of the data. Then the software calculates the numerical result of the specified parameter according to the data treatment selection. Based on the numerical result and uncertainty estimation, the metrologist should provide a decision on the conformance or non-conformance with the specified specification. Finally, the measurement result and the whole measurement procedure can be fed back to the design stage in order to compare with the desired function and estimate the measurement cost to help improve the design process.

4.3.2 The categorical model for PST verification

A series of categories are structured in this section according to the verification model.

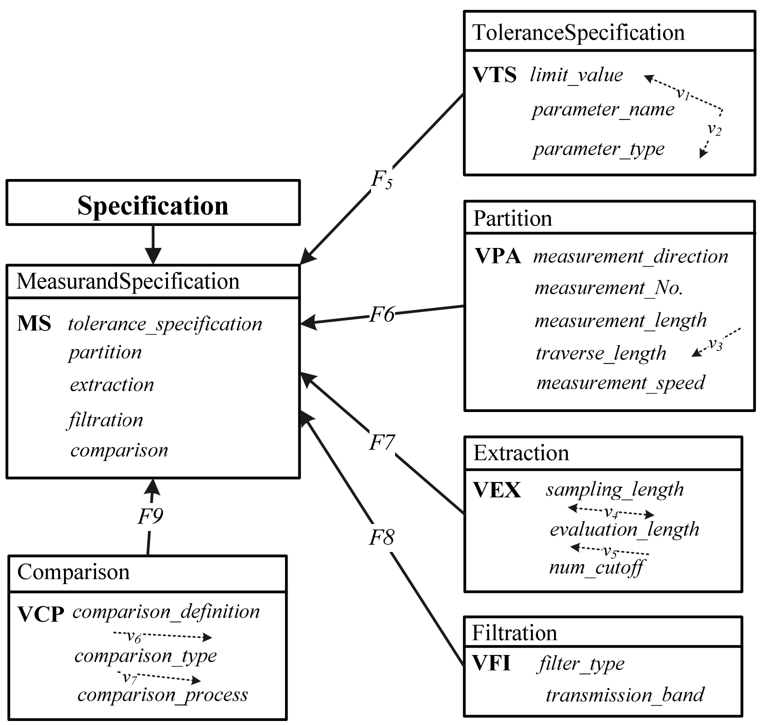


Figure 4.21 Category **MS** and inherited categories **VTS**, **VPA**, **VEX**, **VFI** and **VCP** in PST verification

As shown in figure 4.21, the first category for ‘Measurand Specification’ written as **MS** is determined by the specification process. It interprets the specification and explains the terms, symbols and values shown on engineering drawings. It includes categories **VTS** (Verification Tolerance Specification), **VPA** (Verification Partition), **VEX** (Verification Extraction), **VFI** (Verification Filtration) and **VCP** (Verification Comparison) which are the major mapping operations from categories **TD**, **PA**, **EX**, **FI** and **CP** respectively in the specification model.

The category ‘Measurement Equipment’ written as **ME** as shown in figure 4.22 belongs to the chain link 5 which is the measurement equipment requirements. Six objects *instrument_type*, *tip_radius*, *sampling_spacing*, *instrument_resolution, filter_cutoff* and *measuring_range* represent the type of instrument, the radius of the tip for contact instrument, the sampling spacing, resolution of instrument, the cutoff of filter and the measuring range respectively.

The arrow v_8: *instrument_type* → *tip_radius* states only a contact-method instrument can choose the radius of the tip.

The arrow v_9: *tip_radius*→ *instrument_resolution* represents the fact that the radius of the tip can partially determines the resolution of instrument.

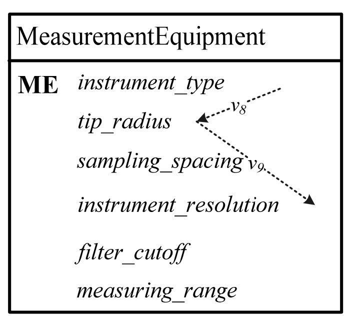


Figure 4.22 Category **ME** for measurement equipment in PST specification

The category ‘Calibration Requirement’ written as **CR** as shown in figure 4.23 belongs to the chain link 6. Five objects *calibration_place*, *calibration_certificate*, *measurement_standard*, *instrument_metrological_characteristics* and *uncertainty_measurement* represent the place that the calibration process takes place, the calibration certificate, the measurement standards, the instrument metrological characteristics and measurement uncertainty. The arrow v_{10}: *calibration_place* →

calibration_certificate states the place that the calibration will take place will be added to the certificate of calibration.

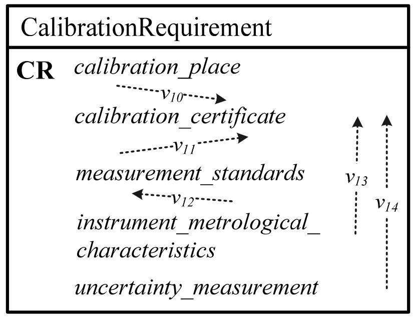


Figure 4.23 Category **CR** for calibration requirement in PST specification

The category 'Measurement Result' written as **MR** as shown in figure 4.24 has two objects *uncertainty_range* and *accept_or_reject* which represent the uncertainty range and the result of whether the measurement result is accepted or rejected. Details of the comparison process will be presented in the next section.

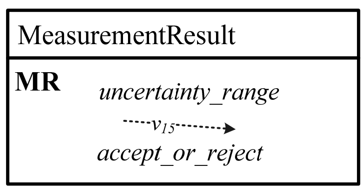


Figure 4.24 Category **MR** for measurement result in PST specification

With reference to the general GPS matrix, PST verification includes a measurand's specifications, and the chain links 4, 5 and 6 which describe the measurement and calibration requirements. A high-level abstract diagram of the categorical model for PST verification is shown in figure 4.25. The internal relationships of category objects are presented by dashed line arrows with label v_i. The **MS**-object determines the **ME** and **CR** objects. The **MR**-objects are generated according to the realisation of the **VCP** objects. As an example, the *comparison_definition* and *comparison_type* determine the *comparison_process* in the **VCP** object, the *limit_value* in the **VTS** object and *comparison_process* in the **VCP** object determine the *measurement_No.* in the **VPA** object which is a part of the **MS** object.

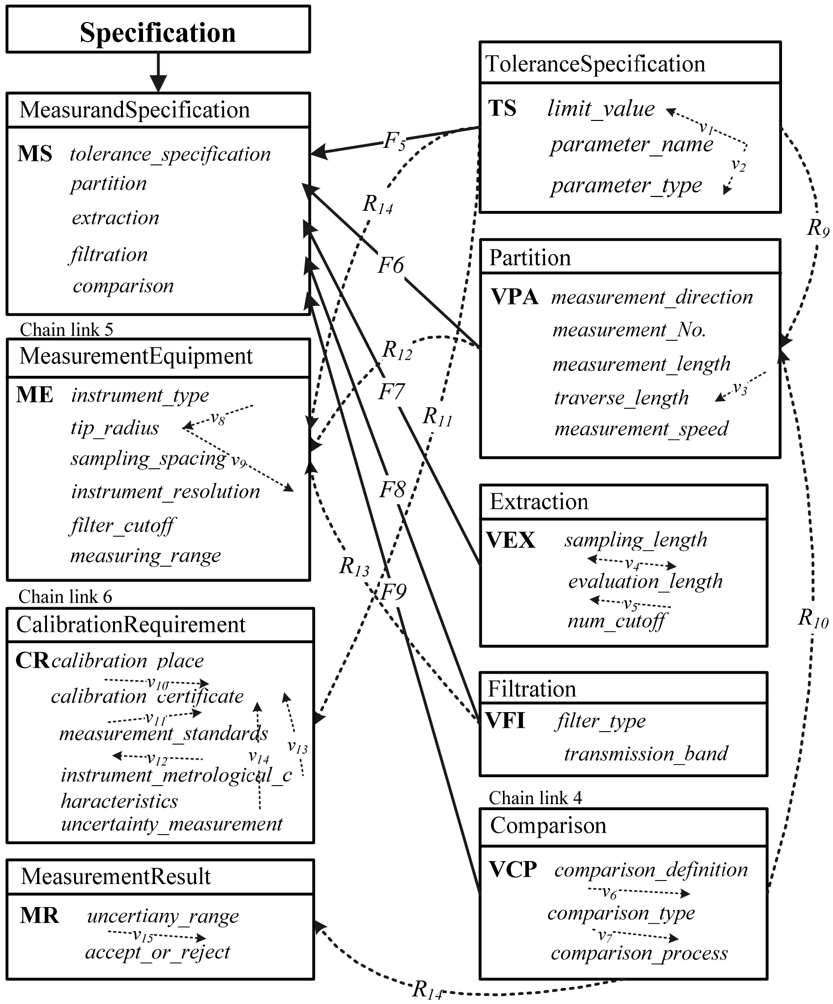


Figure 4.25 The high-level abstract categorical model diagram for PST verification

4.3.3 Relationships

The list of all the pullback relationships in the verification model is shown below:

R_9 - the relationship between objects in the categories **VFI** (Verification Filtration)**, VEX** (Verification Extraction) and **VPA** (Verification Partition);

R_{10} - the relationship between objects in the categories **VCP** (Verification Comparison) and **VPA**;

R_{11}- the relationship between objects in the categories **TS** (Tolerance Specification) and **CR** (Calibration Requirement);

R_{12}- the relationship between objects in the categories **VPA** and **ME** (Measurement Equipment);

R_{13} - the relationship between objects in the categories **VFI**, **TS** and **ME**;

R_{14} - the relationship between objects in the categories **VCP** and **MR** (Measurement Result).

4.3.3.1 The Pullback R_{13} - the determination of instrument type and instrument parameters

For a given specification, firstly, the metrologist needs to choose an appropriate instrument type and related instrument parameters for the measurement. It is their responsibility to find the most appropriate measurement instrument type allowing for low environment demands, low instrument cost, easy operation and calibration. There are several items that should be considered within the instrument selection process.

- The limit value in specification and related sampling interval determines the instrument type i.e. stylus or non-contact methods such as Interferometer, SEM (scanning electron microscope) and AFM (atomic force microscopy).
- Once the instrument type is determined, the limit value can determine the detailed instrument parameters e.g. tip radius, traverse length and data sampling interval.
- Confirm if the instrument software provides the specified filter selection (filter type and cut-off wavelengths) and specified parameter calculation (e.g. *RSm*, Motif series, etc).

Figure 4.26 gives an example of the pullback R_{13} to determine the *tip_radius* in the **ME**-objects using the **VFI** and **TS** objects. Firstly, the objects in the **VFI** and **TS** in the verification are mapped from the **FI** and **TD** objects in the specification respectively. Then, the *transmission_band* in the **FI** object and all of the elements in the **TS** object determine the *tip_radius* in the **ME**-objects. Table 4.7 gives a data example for this determination procedure.

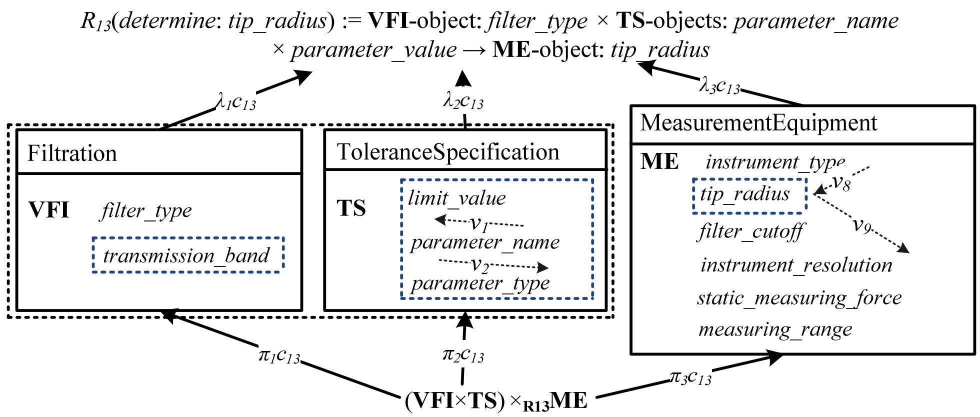


Figure 4.26 Pullback R_{13} - determination procedure of tip radius for stylus instrument

Table 4.7 Examples of relationships between objects in the categories **VFI**, **TS** and **ME**

Category **VFI**	*filter_type*	Gaussian filter
	transmission_band	0.0025-0.8mm
Category **TS**	*limit_value*	0.8μm
	parameter_name	*Ra*
	parameter_type	Profile amplitude parameter
Category **ME**	*instrument_type*	Stylus
	tip_radius	2μm
	filter_cutoff	0.0025-0.8mm
	instrument_resolution	≤0.4μm
	static_measuring_force	0.75mN
	measuring_range	≥5.2mm

4.3.3.2 The Pullback R_{11} -the determination of the calibration process

Once the instrument type is determined, the instrument should have a means of checking its accuracy and repeatability. To achieve this confidence level, a calibration process should be undertaken when a change is made to the basic elements of the system which intentionally or unintentionally modifies the measured profile. However, only those task-related instrument metrological characteristics which are relevant for the intended measurements should be selected for calibration. For example, for the measurement of height parameters such as *Rz*, the spacing profile component need not be calibrated.

Figure 4.27 sketches an example of the pullback R_{11} when determining the *measurement_standards* in the **CR** object by the **TS** object. The **TS** is a mapping from the **TD** in specification, and the *parameter_type* in the **TS** object determine the *measurement_standards* in the **ME** object.

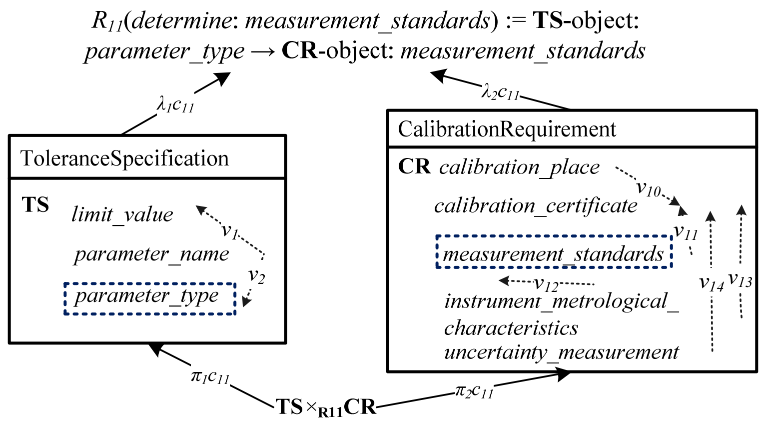


Figure 4.27 Pullback R_{11} - determination procedure of measurement standards.

4.3.3.3 The Pullback R_9 - the determination of measurement length and traverse length

After the calibration process, a series of instrument settings are needed prior to the measurement e.g. metrology environment control, sample preparation, sample set-up, traverse length and traverse speed selections etc. Figure 4.28 illustrates of the pullback R_9 for the determination of measurement length and traverse length; where measurement length is the length over which data is processed. After filtering, a certain amount of data is removed from the measurement length to leave the evaluation length.

For a Gaussian filter, the *measurement_length* = (*num_cutoff* +1) × *sample_length* because half of the first sample length and half of the last sample length are discarded.

For the ISO 2CR filter, the first two sample lengths are discarded, such that *measurement_length* = (*num_cutoff* +2) × *sample_length*.

The traverse length is defined as the distance over which the stylus traverses the surface, and is longer than the measurement length as it is necessary to allow a short over travel to allow for mechanical acceleration and deceleration. For example, these distances for a Taylor Hobson Form Talysurf are 0.3mm at the start of the measured profile and 0.1mm at the end. Assuming that a Gaussian filter is the specified filter type, then *traverse_length* for the Talysurf is (*num_cutoff* +1) × *sample_length* + 0.4mm. In summary, for a specification, the measurement length and traverse length can be deduced according to the category modelling determination procedure.

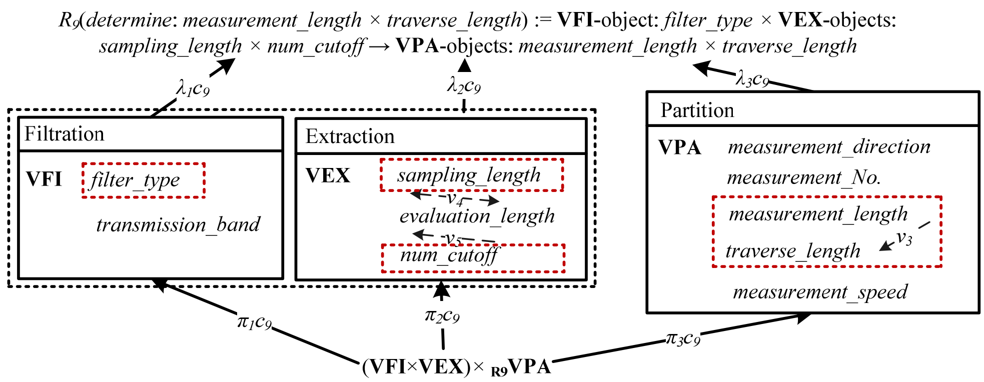


Figure 4.28 Pullback R_9 - determination procedure of measurement length

4.3.3.4 The Pullbacks R_{10} and R_{14} - comparison procedure for conformity assessment

Once the measurement procedure begins, the metrologist needs to know when they should stop the measurement and make a conformity assessment. If the limit value and comparison type are stated within the specification, then the comparison (category **VCP**) operation can determine the number of measurements and specify the form that the measurement result will take, as shown in figure 4.29 and 4.30. The detailed *comparison_process* (object in the category **CO**) flow chart as shown in figure 4.31 and 4.32.

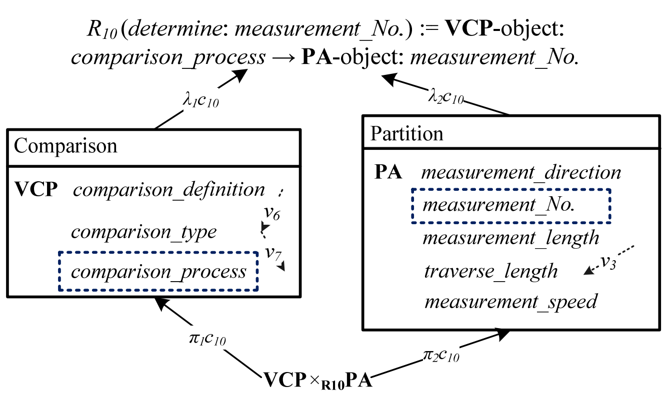


Figure 4.29 Pullback R_{10} - the determination of **PA**-object *measurement_No.*

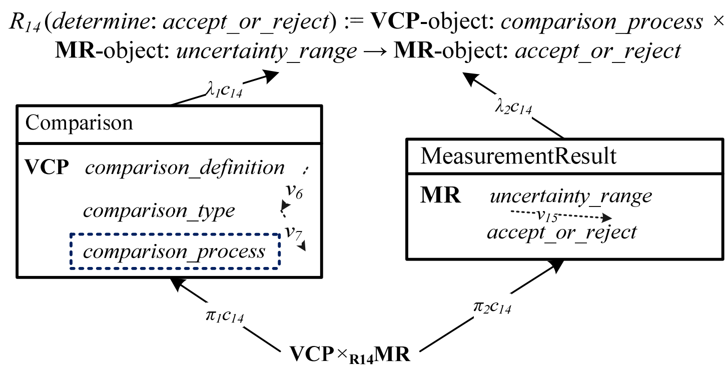


Figure 4.30 Pullback R_{14} - the determination of **MR**-object *accept_or_reject*

The comparison procedure is as follows:

a) adjudge whether the specification starts without a lower limit (denoted as 'L'). If yes, go to the 'Upper limit' section (left side), otherwise go to the 'Lower limit'section;

b) in 'Upper limit' and 'Lower limit' sections, adjudge if the specification does not contain 'max' (max-rule). If yes, make the first measurement, otherwise go to 'e' below;

c) compare the first measured value P_1 with the 70% of V_U. If P_1<0.7V_U or P_1>0.7V_L, then the surface will be accepted and the test procedure stopped; If P_1≥0.7V_U or P_1≤0.7V_L, then two extra measurements are taken;

d) count how many measured values are outside the conformance zone. In 'Upper limit' section, if P_i> (V_U-U), then P_i falls outside the conformance zone and j +1. If P_i< (V_L+U), then P_i falls outside the lower limit conformance zone and j +1;

  1) when three measurements are taken (m=3), if all of the first three measured values are in the conformance zone (j=0), then the surface will be accepted and test procedure stopped; if j>0, then three extra measurements are taken. After the six measurements, go back to the beginning of procedure d;

  2) when m=6, if j=1, then the surface will be accepted and test procedure stopped; if j>1, then six extra measurements are taken. After the twelve measurements, go back to the beginning of procedure d;

  3) when m=12, if j=2, then the surface will be accepted and test procedure stopped; if j>2, then the workpiece is to be rejected and test procedure stopped;

e) in the max comparison section, at least three measurements are taken (m=3). After the measurements, if j=0, the surface will be accepted and test procedure stopped, otherwise go the uncertainty part;

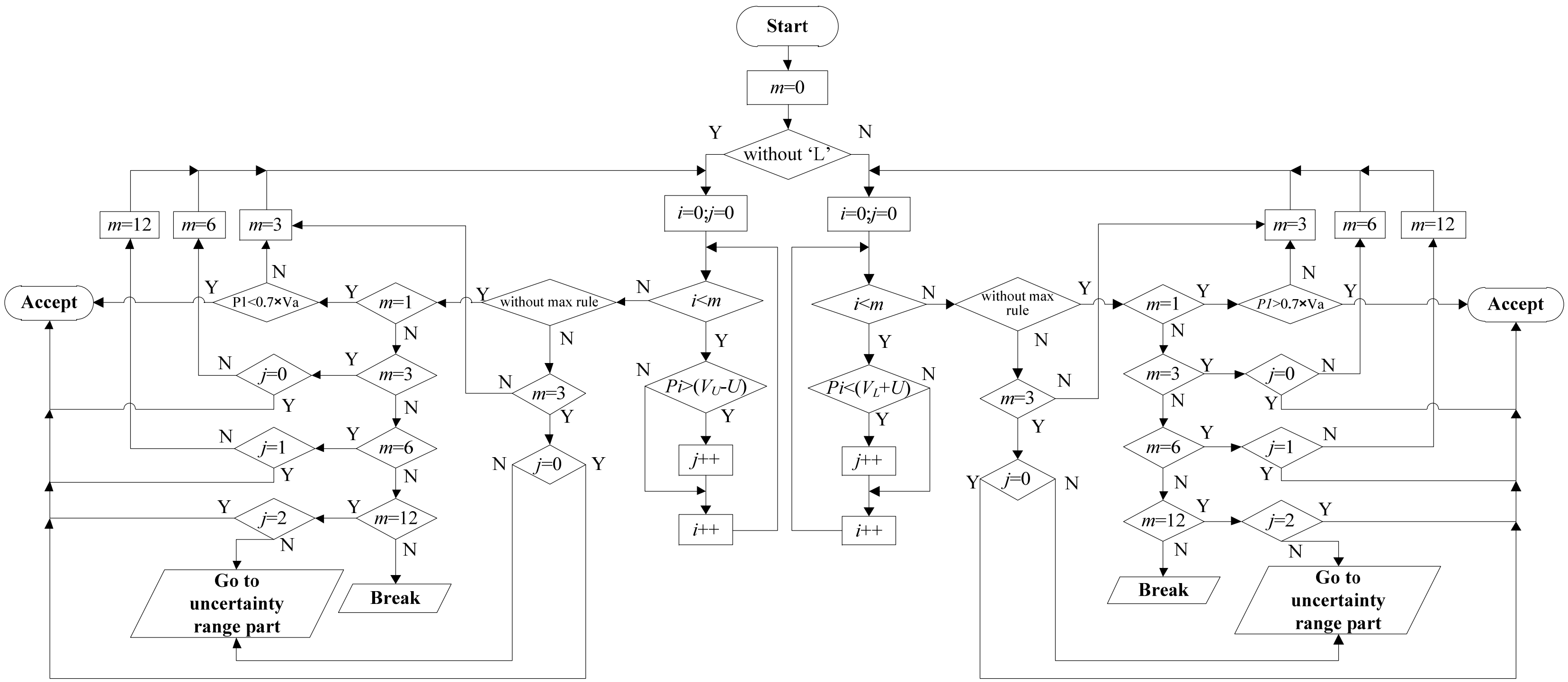
f) if the measured value is outside the conformance zone, there are two possibilities either it is in the uncertainty range or the non-conformance zone. In the 'Upper limit' and 'Lower limit' sections, if $P_i > (V_U+U)$ or $P_i < (V_L-U)$, then P_i exceeds the uncertainty range and n+1. Then adjudge if the specification does not contain 'max';

   1) when the specification contains 'max', if n>0, then the surface is rejected, otherwise the surface is in the uncertainty range;

   2) when the specification does not contain 'max', if n>2, then the surface is rejected, otherwise the surface is in the uncertainty range.

With the implementation of ISO 14253-1:1999, the new zone of conformance is larger than the traditional conformance zone. More workpieces will be in the uncertainty range and less workpieces will be rejected, leading to cost savings by expanding tolerances while still meeting functional requirements. The greater the number of measurements and the longer the evaluation length, the greater is the reliability of the decision as to whether the surface being inspected meets the specification, and the lower is the uncertainty of the parameter mean value. However, an increase in the number of measurements leads to an increase in both the time and the cost of measurement. Therefore, the inspection procedure shall necessarily reflect a compromise between reliability and cost (ISO 1302, 2002).

Furthermore, as the value of U is the main factor in this application, the choice of the uncertainty ratio (relationship between the specification and the uncertainty) is a big issue in saving money. However, there is no scientifically proven guidance on how to choose the right level of uncertainty for measuring a given specification. The only guidance provides a rule of thumb, such as a 4:1 or 10:1.



Key:
m - the number of measurements
P_i ($0\leq i \leq m$) - the measured *Ra* value
V_U and V_L - the upper and lower limit value specified in the specification respectively
j - the numerical count of how many measured values are outside the conformance zone (ISO 14253-2)
U - the measurement uncertainty of the measurement
n - the numerical count of how many measured value are outside the non-conformance zone

Figure 4.31 Flow chart of the comparison process to deduce the measurement time and measurement result

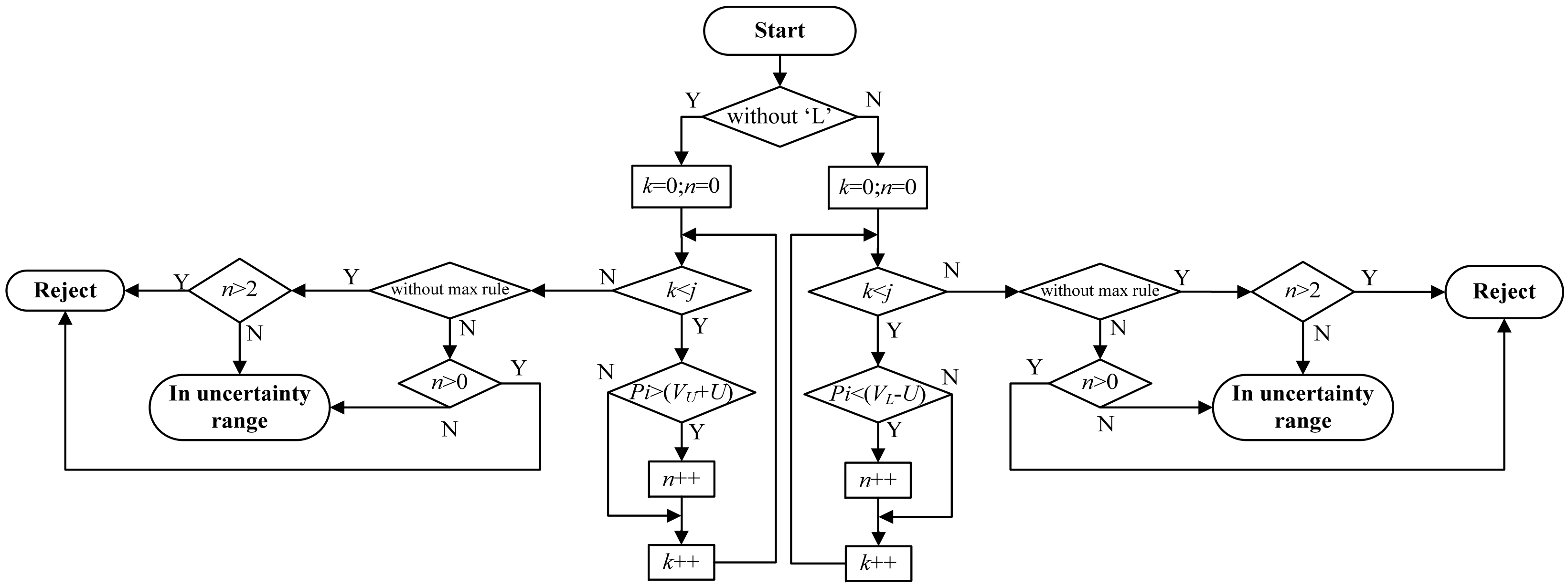


Figure 4.32 The flow chart of uncertainty range deduction

4.4 Conclusions

In this chapter, the categorical model of specification and verification has led to a structured unambiguous expression schema of PST. Categories and objects are applied to represent different knowledge structures; arrows and pullbacks are used to diagram diverse connection between objects; functors are utilised to reveal the mapping between categories in specification and verification. In particular, the manipulation of pullbacks in this thesis is considered as a pullback inference mechanism as most of the objects can be determined by the pullbacks.

The basic philosophies of GPS are the key to connecting specification and verification of surface texture. The utilisation of the categorical model enables the diagramming of sophisticated knowledge in PST as well as AST regardless of the details of structures or connections.

Furthermore, as the uncertainty concepts are still under development, a quantitative specification or measurement uncertainty for a specified PST specification or verification currently is not effective. What we can do to satisfy the requirements is to detail the specification as far as possible consistent with presently available knowledge (especially up-to-date ISO standards).

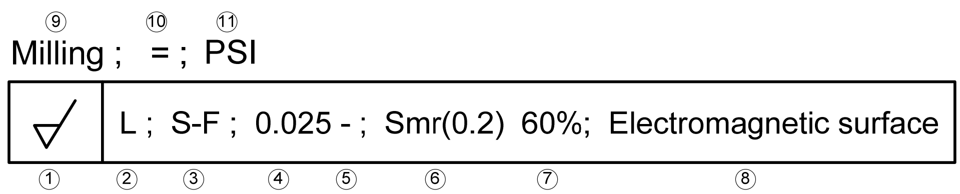
5. Knowledge modelling for Areal Surface Texture (AST)

This chapter details the process of modelling the knowledge of specification and verification in AST. It includes the modelling of the specification and verification process, the categorical model of the specification and verification in AST.

5.1 Knowledge modelling for AST specification

5.1.1 The specification process of AST

Eleven control elements have been defined in the AST specification as shown in figure 5.1. Considering all of the published and unpublished standards in AST, the specification process of AST has been modelled as shown in figure 5.3.

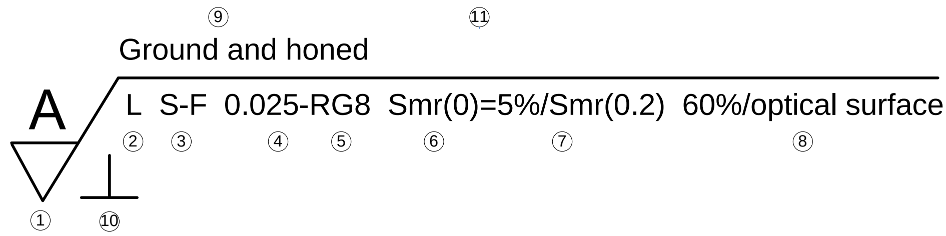


① Surface texture graphical symbol
② Type of tolerance upper (U) or lower (L)
③ Type of scale-limited surface
④ Nesting index - S filter
⑤ Nesting index - F operator or L filter
⑥ Areal parameter
⑦ Limit value
⑧ Other non-default(s)
⑨ Manufacturing process
⑩ Surface texture lay
⑪ Other informtion

Figure 5.1 Control elements in indication of AST on engineering drawings (ISO/CD 25178-1, 2009)

During the revision of this thesis, a very latest version of the indication (as shown in figure 5.2) which appears similar with the profile indication has been updated by ISO/TC 213 (ISO/DIS 25178-1, 2013). This thesis is still adopting the indication from

ISO/CD 25178-1:2009 (see figure 5.1), and the latest version will be updated in the future work.



① Surface texture graphical symbol
② Type of tolerance upper (U) or lower (L)
③ Type of scale-limited surface
④ Nesting index – S filter
⑤ Nesting index – F operator or L filter
⑥ Areal parameter
⑦ Limit value
⑧ Other non-default(s)
⑨ Manufacturing process
⑩ Surface texture lay
⑪ Other informtion

Figure 5.2 Control elements in indication of AST on engineering drawings (ISO/DIS 25178-1, 2013)

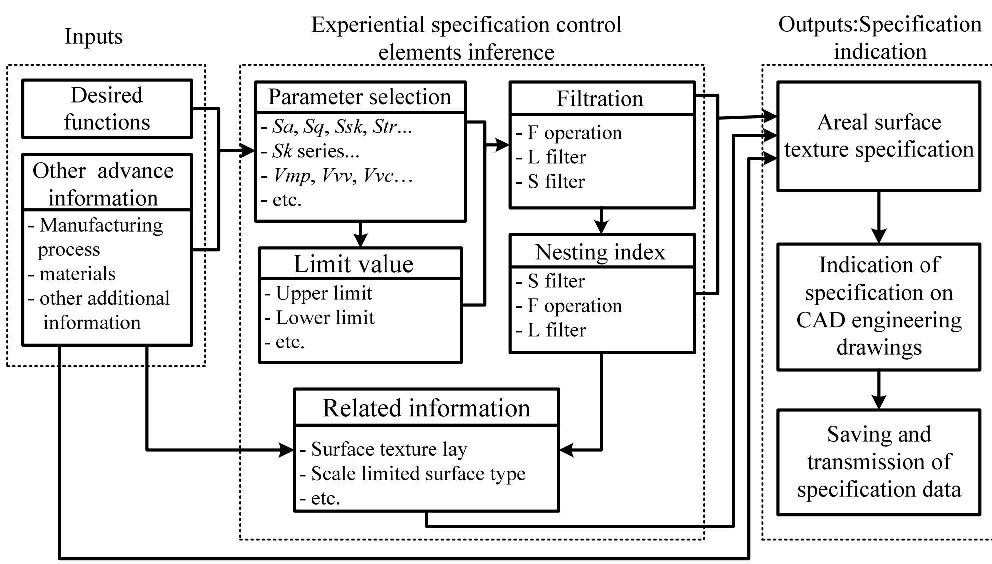


Figure 5.3 The specification process of AST

Desired functions and other information such as manufacturing process and surface materials should be the inputs for a functional design of AST. Different surface components or artefacts may have different input options. The inputs specify that the most appropriate parameter(s) and value should be selected to match the requirements and the scale limited surfaces should be determined according to their functional requirements. Once the scale-limited surface is determined, the nesting indices of the required filters should be assigned. This should be collected with other information such as surface texture lay and other non-default information such that all of the specification control elements defined in ISO/CD 25178-1 should be established according to the inputs and the inference of relationships. After the inference procedure, all of the inferred control elements defined in figure 5.1 can be combined into a complete AST specification. The specification then can be generated by a CAD system on an indication as an engineering drawing and saved as specifications data.

5.1.2 The categorical model for specifications of AST

A series of AST categories are structured according to the specification model.

The category 'Input' written as **IN** as shown in figure 5.4, where **IN**-objects denote the desired functions, the material of the specified surface, the manufacturing process and other information (non-default information about the manufacturing or measurement) that produce the specified surface respectively. The arrow as_1: *surface_function* → *material* states the function of the surface is one of the determining factors for characteristic of material.

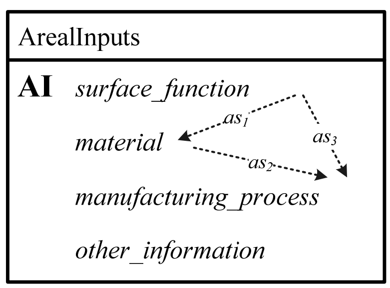


Figure 5.4 The input category **AI** for AST specification

AC (Areal Callout)-objects as shown in figure 5.5 are the eleven control elements in the indication of AST requirements on engineering drawings as shown in figure 5.1. Category **AC** is the most important part of an AST specification, and is inherited by three different categories **ACO** (Areal Codification), **ATD** (Areal Tolerance Definition) and **AFC** (Areal Feature Characteristic) which belong to the first three chain links respectively in the general GPS matrix. Here, AI_j denote the inherited relationships between categories.

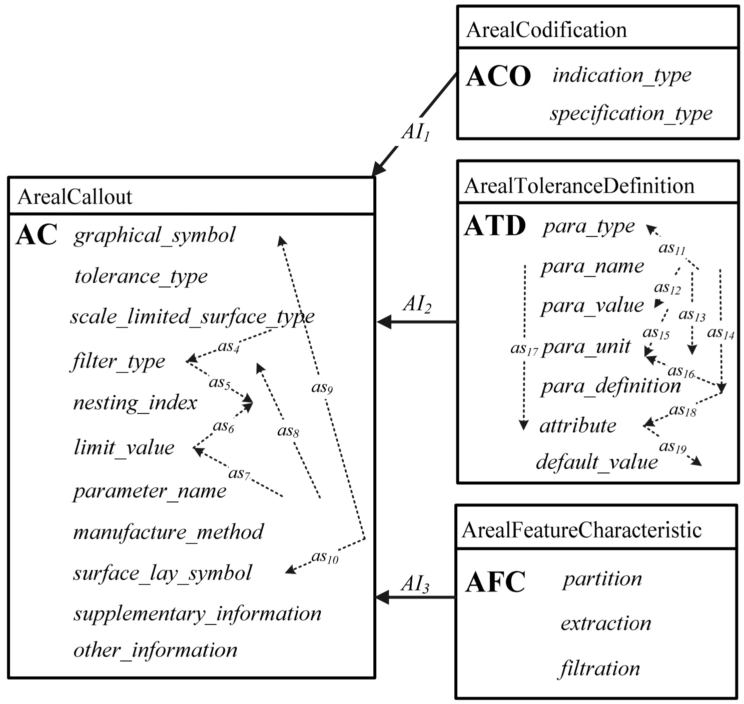


Figure 5.5 The callout category **AC** and related inherit categories **ACO**, **ATD** and **AFC** for AST specification

ACO-objects are the two elements related to specification indication. Object *indication_type* illustrates graphical symbols for three different manufacturing process types; object *specification_type* presents upper and lower specification limit U or L.

Category **ATD** is a category which represents the tolerance definition of AST. It is composed of seven objects (*para_type*, *para_name*, *para_value*, *para_unit*, *para_definition*, *attribute*, *default_value*) and nine arrows (as_{11}, as_{12}, as_{13}, as_{14}, as_{15}, as_{16}, as_{17}, as_{18} and as_{19}). Details of this category can refer to Section 3.2.1.

Category **AFC** represents the feature characteristics in AST. It is composed of partition, extraction and filtration which are three feature operations in GPS. It is inherited from these three categories **AP**, **AE** and **AF** respectively as shown in figure 5.6.

Category **AP** (Areal Partition) represents the partition operation in AST specification. There are four objects and three arrows in this category.

- The arrow as_{20}: *manufacturing_process* → *manufacturing_type* states that every manufacturing process belongs to a kind of manufacturing type such as MRR type or NMR type.

- The arrow as_{21}: *manufacturing_process* → *surface_texture_lay* means every manufacturing process will generate different indication types of surface lay such as '=', 'X' and 'C'.
- The arrow as_{22}: *surface_type* → *manufacturing_process* shows that different surface types such as mechanical or optic surface have related appropriate manufacturing processes.

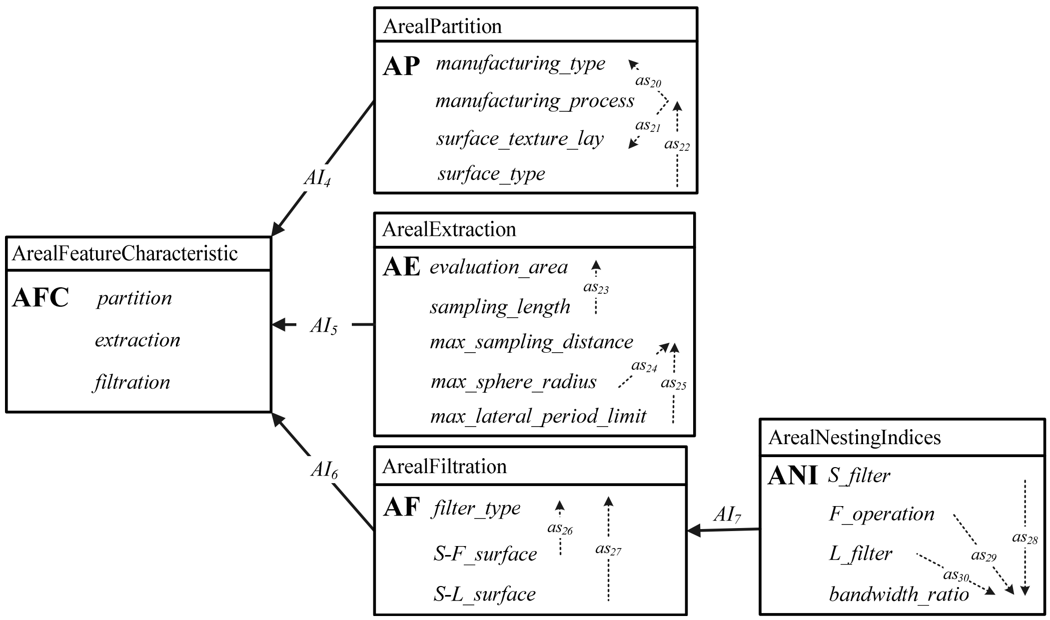


Figure 5.6 Category **AFC** and the inherited categories

Category **AE** (Areal Extraction) represents the extraction operation in specification. Five objects and three are involved.

- The arrow as_{23}: *sampling_length* → *evaluation_area* expresses that the evaluation area can be calculated from the sampling length.
- The arrow as_{24}: *max_sphere_radius* → *max_sampling_distance* means that the value of max sphere radius determines the value of max sampling distance for mechanical surfaces.
- The arrow as_{25}: *max_lateral_period_limit* → *max_sampling_distance* means that the value of max lateral period limit determines the value of max sampling distance for optic surfaces.

There are three **AF**-objects and two arrows involved in the filtration operation in specification *filter_type*, *S-F_surface* and *S-L_surface*.

- The arrow as_{26}: *S-F_surface → filter_type* expresses that an S-F surface has a related filter type which includes an S filter and an F operation.
- The arrow as_{27}: *S-L_surface → filter_type* expresses that an S-L surface has a related filter type which includes both S and L filters.

Category **ANI** is inherited from Category **AF**. Four **ANI**-objects represent the nesting indices for different filters. The arrows as_{28}, as_{29} and as_{30} denote the ratio between nesting indices for an S filter and F operation/L filter. The value of the nesting index for the F-operation or L-filter is normally chosen from the following series:

Table 5.1 The nesting indices for the F-operation or L-filter

…	0.1mm	0.2mm	0.25mm	0.5mm	0.8mm	1.0mm	2.0mm	2.5mm	5.0mm	8.0mm	10mm	...

Informed by the category structures stated above, the whole high-level abstract categorical model for specifications of AST is shown in figure 5.7, where dashed arrows (AP_k) indicate pullbacks between different objects, and are detailed in the following section.

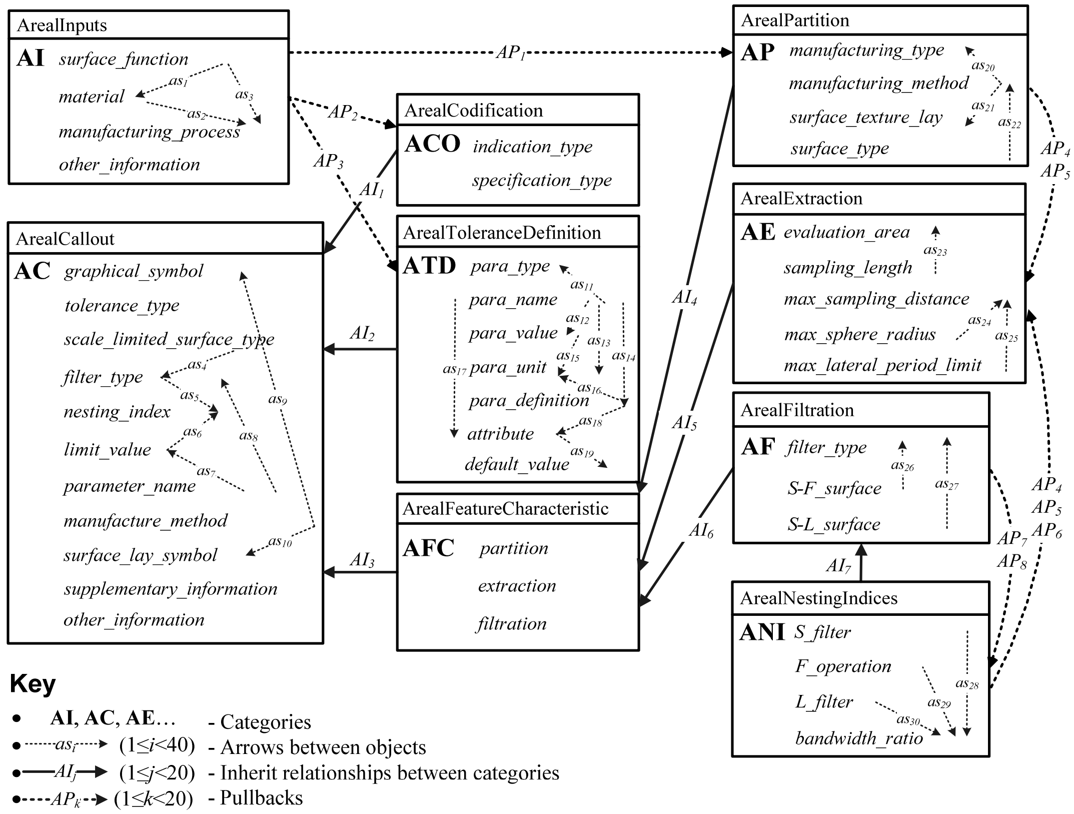


Figure 5.7 The categorical model for **AST** specifications

5.1.3 Relationships

The relationships between objects in different categories are expressed by pullbacks. The list of all the pullbacks in the specification model is shown below:

AP_1 - the relationship between objects in the categories **AI** and **AP**:

AI-object: *manufacturing_process* → **AP**-object: *manufacturing_method*;

AP_2 - the relationship between objects in the categories **AI** and **ACO**:

AI-object: *manufacturing_process* → **ACO**-object: *indication_type*;

AP_3 - the relationship between objects in the categories **AI** and **ATD**:

AI-objects: *functional_surface* × *material* × *other_information* → **ATD**-objects: *para_name* × *para_value*;

AP_4 and AP_5 - the relationship between objects in the categories **AP**, **ANI** and **AE**:

AP-object: *surface_type* × **ANI**-object: *S_filter* → **AE**-objects: *max_sampling_distance* × *max_sphere_radius*;

AP-object: *surface_type* × **ANI**-object: *S_filter* → **AE**-objects: *max_sampling_distance* × *max_lateral_period_limit*;

AP_6- the relationship between objects in the categories **ANI** and **AEL**:

ANI-objects: *F_operation* × *L_filter* → **AE**-object: *evaluation_area*;

AP_7 and AP_8 - the relationship between objects in the categories **AF** and **ANI**:

AF-object: *S-L_surface* → **ANI**-objects: *S_filter* × *L_filter*;

AF-object: *S-F_surface* → **ANI**-objects: *S_filter* × *F_operation.*

In section 3.2.3.2, the details of pullback AP_4 - determination of **AE**-objects *max_sampling_distance* and *max_sphere_radius* has been introduced, see figure 3.8.

Another example of a pullback structure AP_6 - the determination of **AE**-object *evaluation_area* is shown in figure 5.8.

The evaluation area consists of a rectangular portion of the surface over which an extraction is made. If not otherwise specified, the evaluation areal shall be a square whose sides are the same length as the F-operation or L-filter nesting index value. In the pullback structure, the product of object *F_operation* and *L_filter* in category **ANI** determines **AE**-object *evaluation_area*. Data examples of AP_6 are shown in Table 5.2. For example, if the F-operation is a filtration operation, and the nesting index is 0.8mm, the evaluation area is 0.8mm×0.8mm. For the L-filter with nesting index 2.5mm, the evaluation area is 2.5mm×2.5mm.

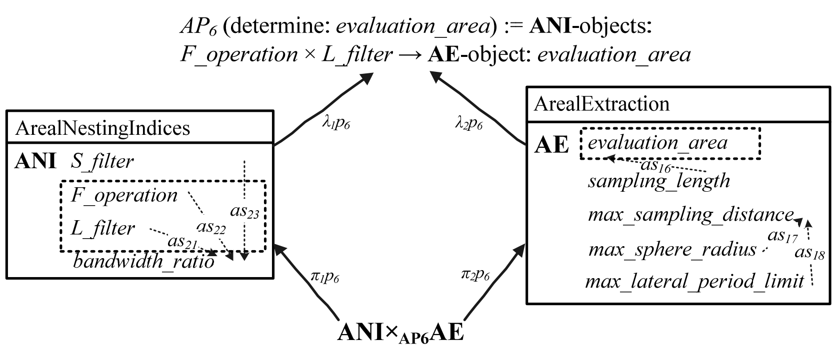


Figure 5.8 Pullback AP_6 - the determination process of **AE**-objects *evaluation_area*

Table 5.2 Data examples of pullback AP_6

ANI		**AE**
F_operation (mm)	*L_filter* (mm)	*evaluation_area* (mm×mm)
0.8	-	0.8×0.8
-	2.5	2.5×2.5

The pullbacks between objects in different categories, allow for most of the objects in the model to be determined. The objects in **AC** can then be inferred by this pullback inference mechanism. This also means that the specifications can be established and the relevant indications can then be generated on engineering drawings.

5.2 Knowledge modelling for AST verification

5.2.1 The verification process of AST

The verification process for AST is modelled as shown in figure 5.9. The figure details the three steps that are required to obtain the final measurement results. In the 'measurement preparation' step, a metrologist analyses the specification, and translates it into measurement specifications which will be used to generate a measurement strategy taking measurement conditions into account. Following the measurement strategy, metrologists carry out the measurement operations and obtain data. Form removal and filtration options, are then selected. The software then calculates the numerical results of the specified parameters in the last step. These numerical results and accompanying uncertainty estimation can then be used to provide a decision on conformance or non-conformance with the specified specification. Finally, the measurement results are feedback to the design stage in order to compare with the desired function which will help improve functional design.

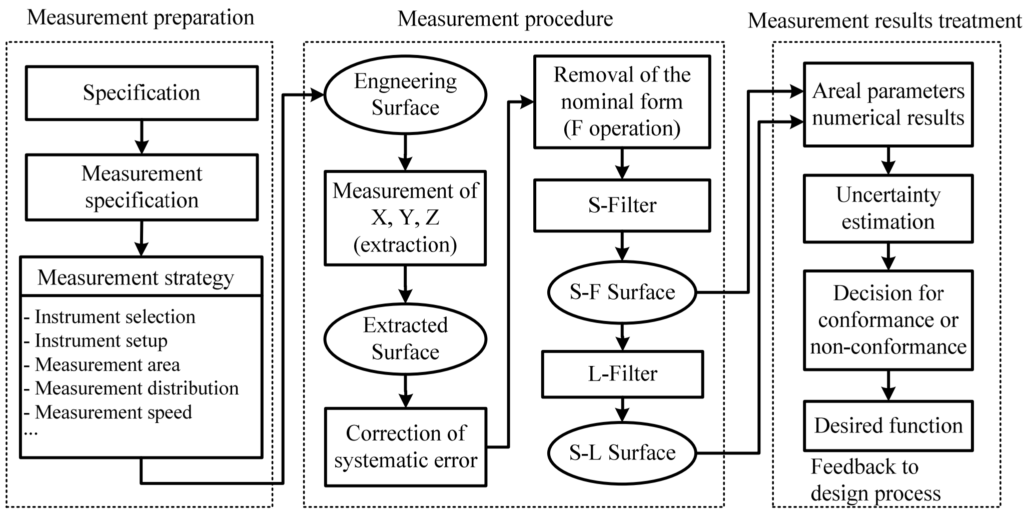


Figure 5.9 The verification process of AST

5.2.2 The categorical model for verification of AST

A series of AST verification categories are structured in accordance with the verification model. Category **AMS** (Areal Measurement Specification) as shown in figure 5.10 is mapped from the specification categorical model. It includes four objects (*tolerance_specification*, *partition*, *extraction* and *filtration*) which are inherited by five categories **ATS** (Areal Tolerance specification), **APV** (Areal Partition Verification), **AEV** (Areal Extraction Verification), **AFV** (Areal Filtration Verification), **ANIV** (Areal Nesting Indices Verification) respectively. These five categories are mapped from the categories (**ATD**, **AP**, **AE**, **AF**, and **ANI**) in specification, written as

AF_1: **ATD** → **ATS**,

AF_2: **AP** → **APV**,

AF_3: **AE** → **AEV**,

AF_4: **AF** → **AFV**,

AF_5: **ANI** → **ANIV**.

Following the explanation of the functor AF_1 which is described in section 2.2, every object and arrow in the category is mapped to the objects and arrows in another category, so are the pullbacks between different objects such as $AP_4 \rightarrow AP_{17}$, $AP_5 \rightarrow AP_{18}$, $AP_6 \rightarrow AP_{19}$, $AP_7 \rightarrow AP_{20}$, $AP_8 \rightarrow AP_{21}$.

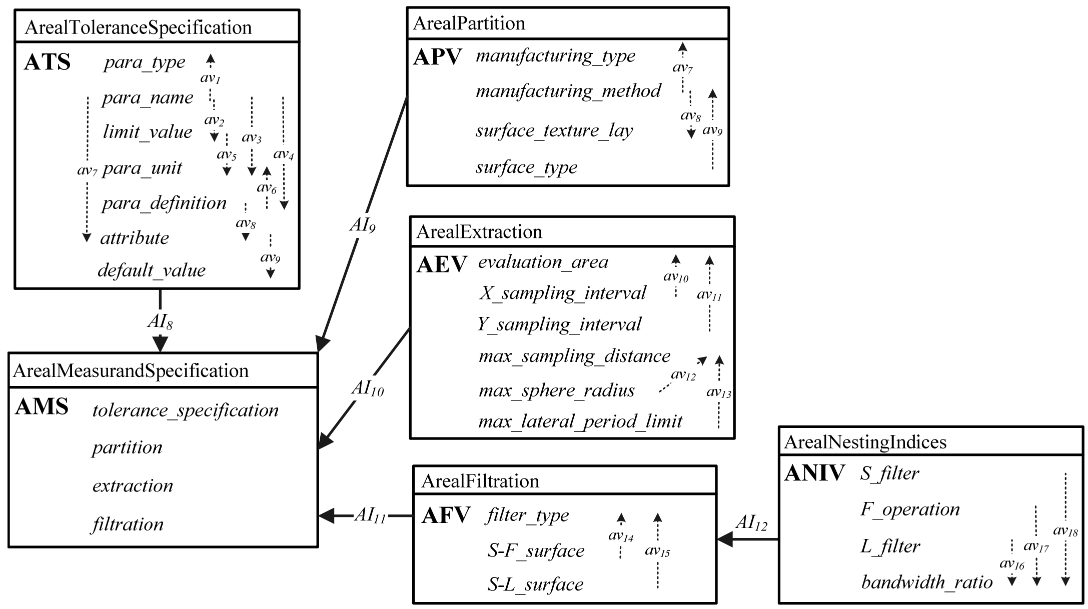


Figure 5.10 Category **AMS** and the inherited categories **ATS**, **APV**, **AEV**, **AFV** and **ANIV**

Figure 5.11 shows a category **AME** (Areal Measurement Equipment) in the verification of AST. Seven **AME**-objects are the elements presenting characteristics of measurement instrument. The arrows av_{18} - av_{23} mean that the type of instrument determines all the instrument characteristics such as repeatability, the measure range, lateral and vertical resolution, the software functions and installation conditions etc.

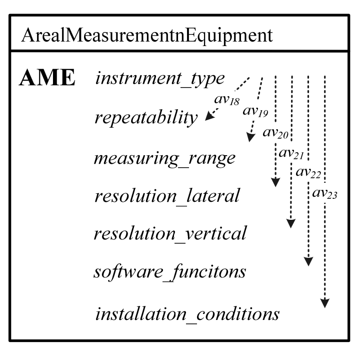


Figure 5.11 Category **AME** for areal measurement equipment in AST verification

Category **ACR** (Areal Calibration Requirement) as shown in figure 5.12 demonstrates the calibration requirements in the verification process. Six **ACR**-objects are required to characterise instrument calibration. The arrows av_{24} and av_{25} mean all kinds of measurement standards have related assessed parameters and measurement methods; the arrows av_{26} - av_{30} state that all the characteristics in calibration operation should be considered in the process of estimating the measurement uncertainty. The arrow av_{31} means that every assessed parameter has a result.

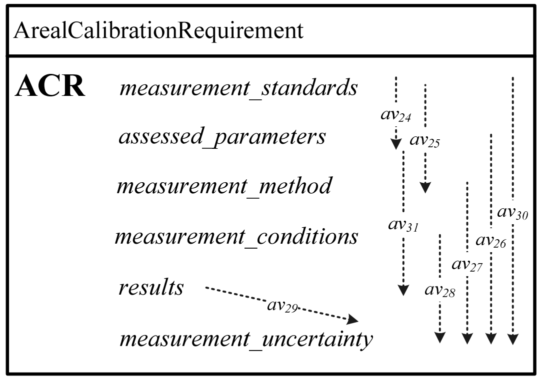


Figure 5.12 Category **ACR** for areal calibration requirement in AST verification

Category **AMR** as shown in figure 5.13 presents the measurement result in the verification process.

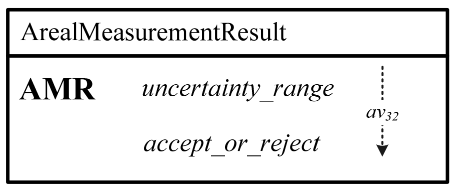


Figure 5.13 Category **AMR** for areal measurement result in AST verification

The high-level abstract categorical model for verification of AST is shown in figure 5.14. With reference to the general GPS matrix, the AST verification includes specification of the measurand, chain links 4-6, which are characteristic of the measured features, and the measurement result.

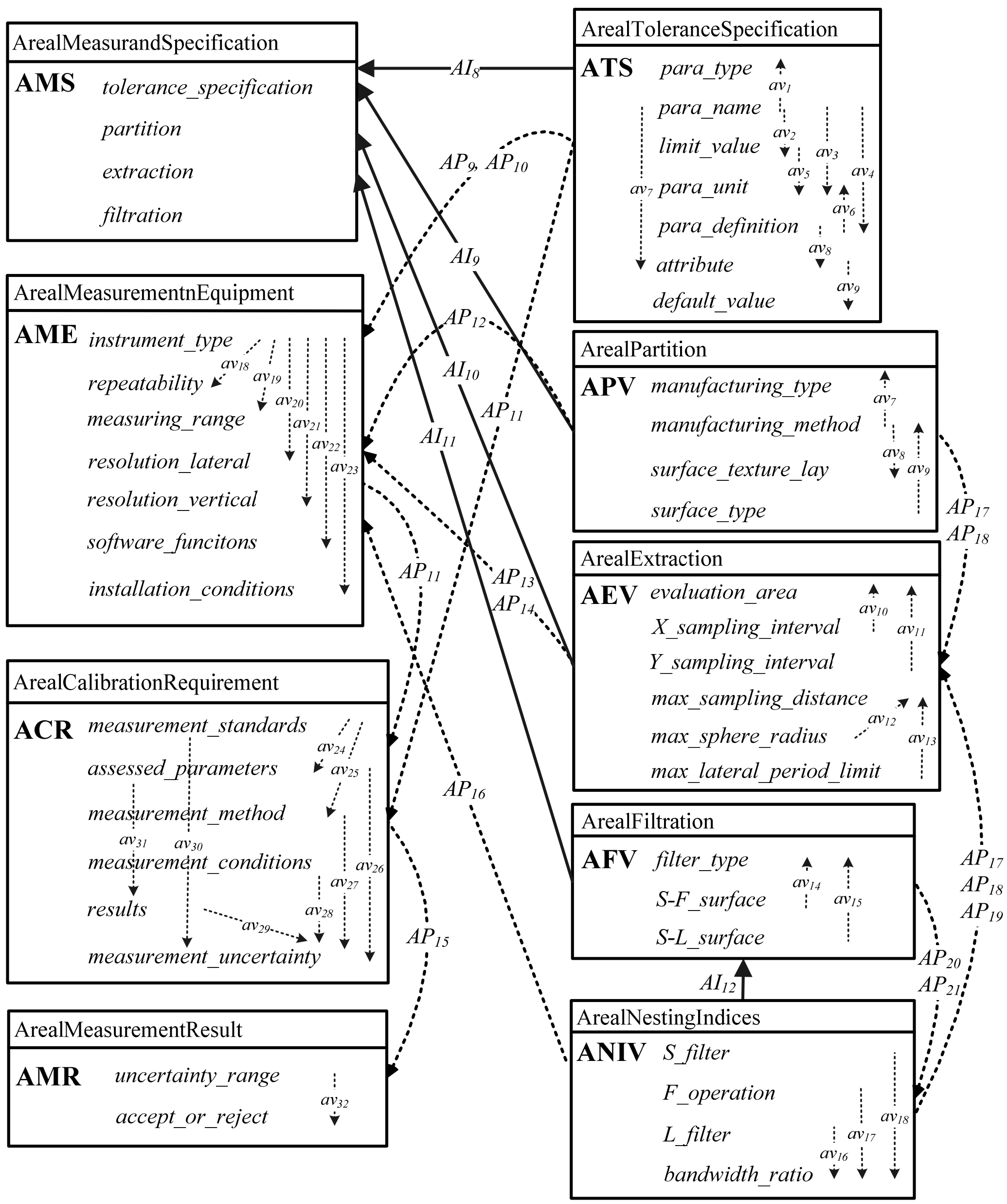


Figure 5.14 The categorical model for AST verification

5.2.3 Relationships

By the pullback inference mechanism, pullbacks AP_k can determine most of the objects in different categories in the AST verification. The details of every pullback in the verification are shown as follows:

AP_9 - the relationship between objects in the categories **ATS** and **AME**:

ATS-objects: *para_name* × *limit_value* → **AME**-objects: *resolution_lateral* × *resolution_vertical*;

AP_{10} and AP_{11}- the relationship between objects in the categories **ATS** and **AME**:

ATS-object: *para_name* × *limit_value* → **AME**-object: *software_functions*;

AP_{11}- the relationship between objects in the categories **ATS**,**AME** and **ACR**:

ATS-object: *para_type* × **AME**–objects: *instrument_type* → **ACR**-object: *measurement_standard* × *assessed_parameters*;

AP_{12} - the relationship between objects in the categories **APV** and **AME**:

APV-object: *surface_type* → **AME**-objects: *instrument_type*;

AP_{13} and AP_{14} - the relationship between objects in the categories **AEV** and **AME**:

AEV-objects: *evaluation_area* → **AME**-object: *measuring_range*;
AEV-objects: *X_sampling_interval* × *Y_sampling_interval* → **AME**-objects: *resolution_lateral* × *resolution_vertical*;

AP_{15} - the relationship between objects in the categories **ATS** and **AME**:

ACR-object: *measurement_uncertainty* → **AMR**-object: *uncertainty_range*;

AP_{16} - the relationship between objects in the categories **ANIV** and **AME**:

ANIV-object: *S_filter* × *F_operation* × *L_filter* → **AME**-object: *software_functions*;

AP_{17} and AP_{18}- the relationship between objects in the categories **APV, ANIV** and **AEV**:

APV-object: *surface_type* × **ANIV**-object: *S_filter* → **AEV**-objects: *max_sampling_distance* × *max_sphere_radius* (It is mapped from AP_4);
APV-object: *surface_type* × **ANIV**-object: *S_filter* → **AEV**-objects: *max_sampling_distance* × *max_lateral_period_limit* (It is mapped from AP_5);

AP_{19} - the relationship between objects in the categories **ANIV** and **AEV**:

ANIV-objects: *F_operation* × *L_filter* → **AEV**-object: *evaluation_area* (It is mapped from AP_6);

AP_{20} and AP_{21}- the relationship between objects in the categories **AFV** and **ANIV**:

AFV-object: *S-L_surface* → **ANIV**-objects: *S_filter* × *L_filter* (It is mapped from AP_7);

AFV-object: *S-F_surface* → **ANIV**-objects: *S_filter* × *F_operation*(It is mapped from AP_8).

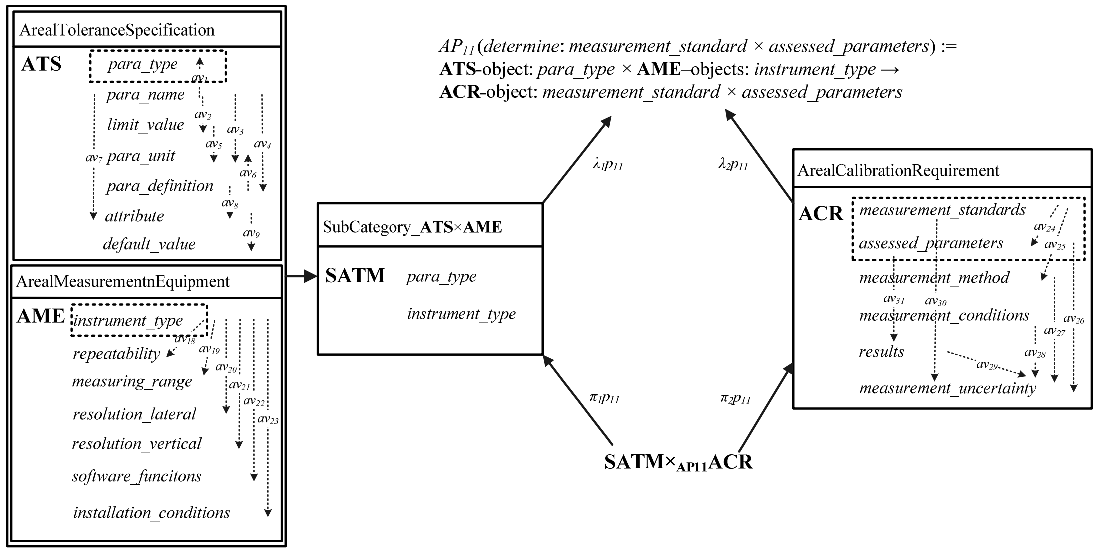


Figure 5.15 An example of pullback AP_{11} - the determination process of **ACR**-objects *measurement_standards* and *assessed_parameters*

Figure 5.15 gives an example of pullback structure AP_{11} - the deduction of **ACR**-objects *measurement_standards* and *assessed_parameters*. The product of **ATS**-object *para_type* and **AME**-object *instrument_type* determines **ACR**-objects *measurement_standards* and *assessed_parameters*. In the pullback structure, the objects *para_type* and *instrument_type* from the product of categories **ATS** and **AME** constitute a subcategory **SATM**.

The pullback structure AP_{11} means that the specified AST parameter type and related features of the measurement instrument determine the type of measurement standard and related assessed parameters in the calibration process.

Examples of AP_{11} data are shown in Table 5.3, for an areal height parameter, if the calibration applies to a measuring instrument that has a limited vertical measuring range and no accurate motion correction, the suggested standards will be types of ER2, ER3, CG1 or CG2 see (ISO 25178-701). For standard type ER2, the assessed parameters are distance l_1 and l_2 between the grooves; for type ER3, it is diameter D_f along the X- axis and the Y-axis. When the specified parameter is height or function type, if the calibration applies to the measuring instrument having a large vertical measuring range and an accurate motion correction, the suggested standard will be of type ES and the related assessed parameter is diameter D_i along X-axis and Y-axis.

Table 5.3 Data examples of pullback AP_{11}

ATS	AME	ACR	
para_type	*instrument_type*	*measurement_standards*	*assessed_parameters*
Height parameters	Instruments have a limited vertical measuring range and no accurate motion correction	Standard ER2, ER3, CG1 or CG2	For ER2: distance l_1 and l_2 between the grooves For ER3: diameters D_f along the X-axis and the Y-axis
Height and function parameters	Instruments have a large vertical measuring range and an accurate motion correction	Standard ES	Diameter D_i along X-axis and Y-axis
Spatial parameters	Instruments have a large measuring range and an accurate motion correction	Standard ER2, ER3 or ES	Δ_{PER} (see ISO 25178-601, 2010)

5.3 Conclusions

This chapter utilises category theory to model the diverse and sophisticated knowledge for specification and verification in AST. As the development of AST standards are still in progress, much modification and updating will be required as well as final publishing of AST standards. Utilisation of such a diagramming modelling approach makes it easier to update for programme designers. The knowledge model in this chapter is the foundation for developing the AST design and measurement guide system for mechanical designers and metrologists.

6. Design and development of the CatSurf system

This chapter focuses on the design and development of the CatSurf system which is a platform with knowledge generation and accessing facility based on GPS philology. The system is designed to bridge the gap between DMMs, and integrate the surface texture information and corresponding GPS realisation methodologies into an integrated CAx framework. The architecture of the CatSurf system presented in Section 6.2 includes three different modules (each composed of five components), a categorical database to provide data and information support for the modules. The development of the system is demonstrated in Section 6.3 with implementations of three different modules of the system presented in Section 6.4-6.6 respectively. Finally the implementation of the help document for the system is the subject of Section 6.7.

6.1 Introduction

The CatSurf system spans knowledge domains from surface specification, related manufacturing processes/equipment, to verification principles and calibration requirements, as well as uncertainty and measurement traceability. The envisaged potential benefits of the system can be summarised as:

- To provide a unified database for supporting engineering decisions in choosing appropriate surface texture specification elements and verification parameters according to required functional performances.
- To enable an automated querying mechanism for guiding designers with unambiguous surface texture specifications, verification and GPS-recommended information.
- To link similar functions for aiding decisions on measurement procedures and equipment.

- To provide an interface platform for facilitating CAx users access to the CatSurf system.

To achieve the desired system functions, the proposed system specifications have the following design features:

- Flexible data storage to enable data sharing, maintenance and protection through representing GPS information in the form of knowledge objects in the object-oriented style, which can be readily adopted by other platforms and tools.
- Client/Server structure for data synergy and remote collaboration between geographically dispersed designers, production engineers and metrologists.
- User-friendly system interfaces for accessing system data and functions such as cross-referencing and advanced updating.

6.2 System architecture

This section aims to demonstrate the architecture of the CatSurf system. The architecture on which the system is constructed is based on the product chain in which surface texture is defined.

The main components of the CatSurf system are presented with one database and three modules each with five components as shown in figure 6.1. The three modules '*ProfileControl*', '*SurfControl*' and '*ArealControl*' are focused on different approaches to measurement of the surface features. *ProfileControl* is a module specific to deal with design and measurement of PST. *SurfControl* is a case study of *ProfileControl* which is designed only for design and engineering specification to comply with internal standards of Rolls-Royce. *ArealControl* is developed to operate in accordance with the underdeveloped AST standards. According to the position of the product chain which involves surface texture, each module includes five components which are 'Function', 'Manufacture', 'Specification', 'Verification' and 'Help'. Here, the first three components are part of the design phase; the Verification component is designed for surface texture measurement; the Help component is developed to provide all the information for the former four components. A categorical database is developed to support all the data and information store, manipulation, querying and reasoning in the three modules and related five components. The database is based on the knowledge model presented in chapters 4 and 5.

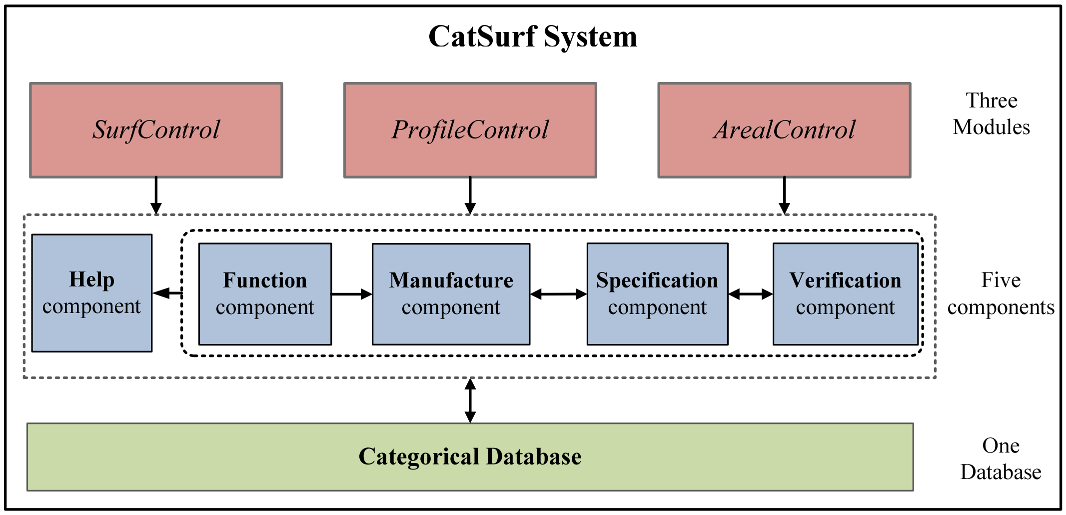


Figure 6.1 Main components of the CatSurf system

6.2.1 Five components

Five components are designed to provide both designers and metrologists with related information based on different phases in the product chain. As shown in figure 6.2, designers are involved in 'Function', 'Manufacture' and 'Specification'; metrologists are involved in the 'Verification'. All four components are expected to:

- provide databases[20] for data storage and induction;
- manipulate input and output data;
- provide a human-computer interaction interface.

Accordingly, the former four components are designed with a related database, interfaces, and input and output data processing mechanisms. Depending on the external input of function and other requirements, all output data will be transferred to the following components.

[20] The databases in each component are the sub-databases in the categorical database.

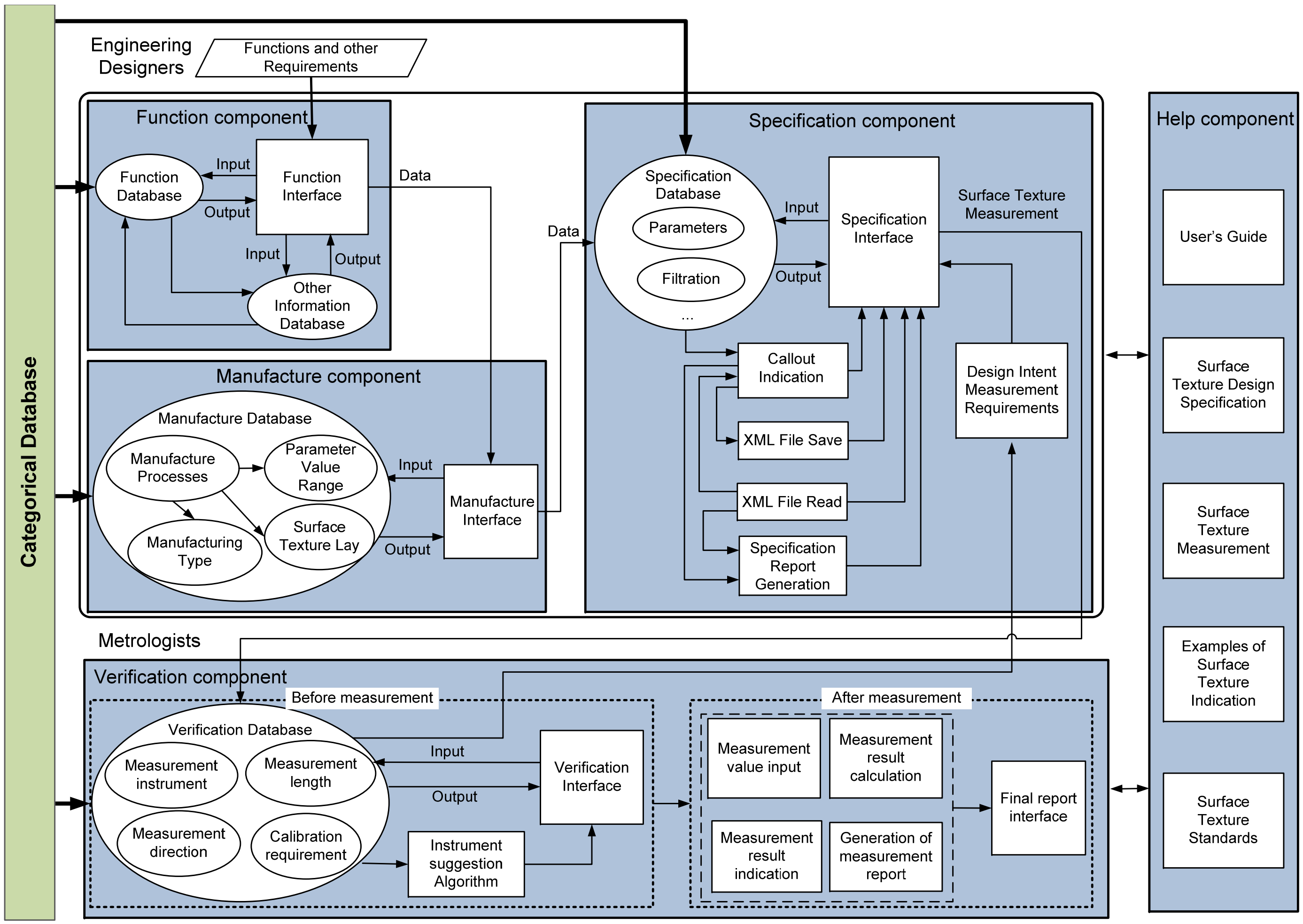


Figure 6.2 The interaction between 'Function', 'Manufacture', 'Specification', 'Verification', and 'Help' components

6.2.1.1 The Function component

Functional requirements are one of the most important considerations in assigning appropriate specification elements. The Function component aims to provide all relevant information for the engineered artefact before the assignment of a specification. This component is designed to help designers with optimal specification elements such as suggested parameters, limit values, applicable manufacturing processes etc. Besides the common objectives with other components, the design of the Function component is expected to:

- deal with different kinds of functional requirements and other information such as the dimension or tolerance of the specified surface;
- provide experimental or recommend surface texture parameters and limit values.

Accordingly, the two databases which are the function database and the other information database for storing and deducing related information are placed in the Function component. A Function interface for gaining inputted data and outputting the deduced results is expected to be developed (as shown in figure 6.2). As indicated in table 6.1, the Function interface provides various surface functions, component information, materials and other information used for selection. The designers input the requirements, then the input data will be sent to the functional database or other information database for related output information such as function related parameters, limit value or suggested manufacturing process. In many cases, the generating procedure may need to query the other information database. The function database and other information database will provide all the inputs required for relationship manipulating. The information reasoning will apply the relationship mechanism which was developed in chapter 3. For example, to assign surface texture specification for a mating bearing shaft surface, if the required function performance is fitting and wear, the suggested parameter could be *Ra* with a limit value 1.6μm depending on experimental results.

Table 6.1 Series of input information

Function Interface Component		*Examples*
Function Performance	Functional Surfaces	Convex spherical sliding surfaces for control rod ends
	Component Types	'D' bolt abutment area for turbine shafts
	Unspecified Surfaces	Centre drill holes
Other Information	Materials	Steel titanium
	Tolerance	5 (International Tolerance)
	Dimension	50mm

Finally, the output data from the Function component will be sent to the Manufacture component as the input data.

6.2.1.2 The Manufacture component

The Manufacture component is the guide for the manufacturing process involved in creation of surface texture rather than for manufacturing process planning. It is an essential link between the Function and Specification components. The design of the Manufacture component is expected to:

- provide different kinds of manufacturing process and related key information such as the capability of the manufacturing process, and the expected different surface texture lay of the manufacturing process;
- recommend the manufacturing process for certain functional surfaces;
- provide restriction rules and suggested corrective action in a situation where a designer selects the wrong manufacturing process.

Accordingly, a manufacture database which includes manufacturing processes, manufacture types, surface texture lay and parameter value range is placed in the Manufacture component. As shown in figure 6.2, transferring the function selection and output data in the Function component, the Manufacture interface will link to the manufacture database for inferring the right manufacturing process and related information such as parameter value range and surface texture lay. For example, if the specified surface is designed to be manufactured by turning, the expected range of *Ra* is 0.025-25μm (see table 4.6) and possible surface texture lay will be '=', '⊥' or 'C' if the specified surface is the end face of a cylinder.

Finally, the output data will be returned to the Manufacture interface and will be transferred to the next component.

6.2.1.3 The Specification component

The Specification component aims to provide complete surface texture specifications for designers with the least amount of input information. As stated previously, the specification of surface texture is the design step where all control elements (ten for PST and eleven for AST) are stated, accommodating the design requirements of the workpiece and it's functional surfaces corresponding to the required production capabilities and for the use in design and engineering drawings. The data from both Function and Manufacture components will generate inputs for this component to generate a complete surface texture specification. The design of the specification component is expected to:

- avoid the indiscriminate use of surface texture values that result in impractical and costly production requirements;
- generate a complete specification based on the information gained in Function and Manufacture components;
- provide the opportunity for designers to revise the specification details according to their specialised requirements;
- generate and save indications and specification data;
- provide a specification report to explain indications;
- provide basic measurement information for designers.

A specification database is designed to store and manipulate all specification data. As shown in figure 6.2, all the data from both the Function and Manufacture components will be sent to the specification database for generating the control elements, the generated results will then be produced as a callout indication which will be shown in the Specification interface. The process of generating a complete specification is carried out by the specification categorical model presented in chapters 4 and 5. In the interface, designers are allowed to change the details of certain specification elements under limited privileges. However, any revisions which are contrary to previous inputs such as functional requirements and manufacturing process, or any other input which is contrary to the relationship restriction in the specification models will not be allowed. The generated specification will be saved into an XML (Extensible Markup Language) file; every detail of the specification will be explained in a specification report. Furthermore, the measurement database in the verification process will be connected to this so that designers are provided with the required indications so that they have a straightforward understanding about the measurement requirements of the assigned specification.

6.2.1.4 The Verification component

The Verification component is split into two different sections - the measurement strategy and the final report. The measurement strategy is designed to:

- provide the metrologist with detailed measurement parameters such as the measurement environment, measurement direction and length and calibration requirements;
- provide a suggested instrument according to the specification;
- generate a measurement report.

The final report is designed to:

- record the details of the measurement environment such as measurement time, humidity and operator;
- calculate the number of measurements;
- estimate the measurement uncertainty;
- indicate the measurement result;
- provide a conformance zone to make a measurement result decision according to the specification and uncertainty.

In the measurement strategy component, a verification database which includes measurement length, measurement instrument, measurement direction and calibration requirement, is developed. To provide the recommended instrument, an instrument suggestion algorithm (Wang, 2008) is placed in the section. As shown in figure 6.2, a main Verification interface is developed to provide both the measurement strategy and the attainment of final report interfaces. The measurement report which includes all the details of measurement strategy will be generated in the verification interface.

The final report component includes the input of measurement environment and value, the calculation of the measurement result by considering the uncertainty, the indication of measurement result and the generation of the final measurement report. All these functions will be shown in the interface.

6.2.1.5 The Help component

The Help component is established to provide users with all the information they need to use and understand the CatSurf system. Users are expected to use the help document as a handbook for both the CatSurf system and surface texture design and measurement.

As shown in figure 6.2, five sections in the Help document have been designed. The user guide demonstrates how to use this system step by step. The second is the surface texture instruction in GPS which includes all definitions, terms and parameters involved in surface texture specification, the relationship between function and surface texture, the Manufacture component in surface texture etc. The third is the verification of PST and AST. The fourth is a list of all related surface texture standards. The last one gives different indication examples and related explanations.

6.2.2 Three modules - *ProfileControl*, *SurfControl* and *ArealControl*

6.2.2.1 ProfileControl

*ProfileContro*l is designed to provide designers and metrologists with suggested specification and verification information in PST. This module is composed of the five components in PST and it's structure is shown in figure 6.3. The five components are placed in three different categories according to their different users. In the structure, Profile Specification includes Function, Manufacture and Specification components; Profile Verification includes Measurement Strategy and Final Report.

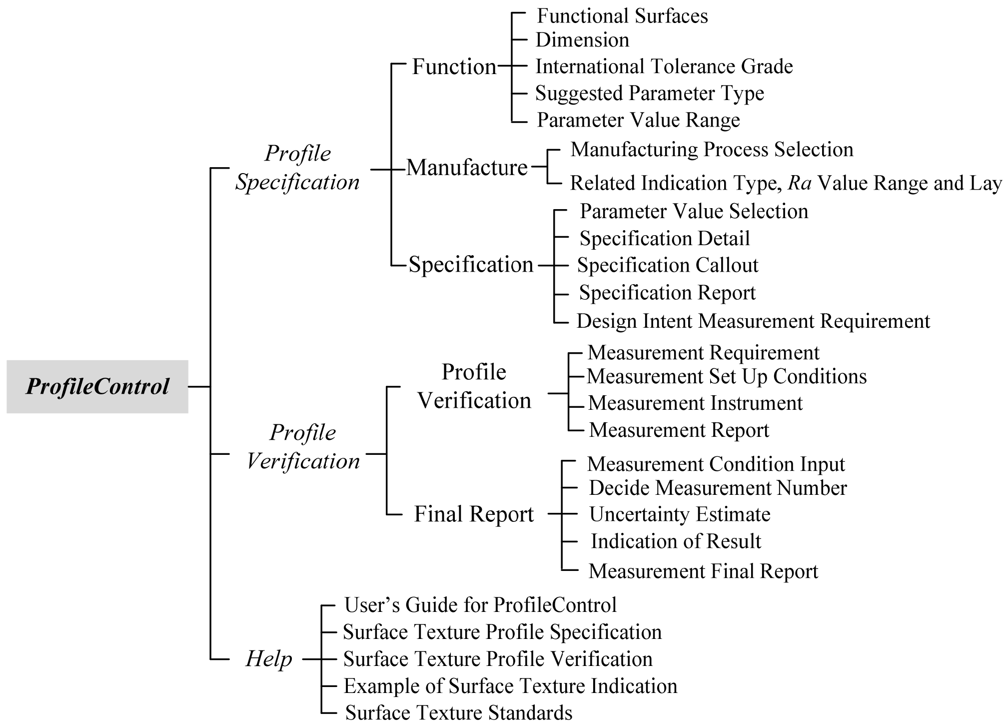


Figure 6.3 The structure of *ProfileControl*

6.2.2.2 SurfControl

SurfControl is a single case study of *ProfileControl* that is unique to Rolls-Royce. As shown in figure 6.4, there are number of differences between *SurfControl* and *ProfileControl*. Firstly, *Ra* is the only parameter in the Rolls Royce specification, whereas full selection of profile parameters is available in the *ProfileControl*. Secondly, the functional requirements in the Function component of SurfControl are mainly focus on gas washed surfaces. Thirdly, the required manufacturing processes from R-R are then mainly used for gas washed surfaces.

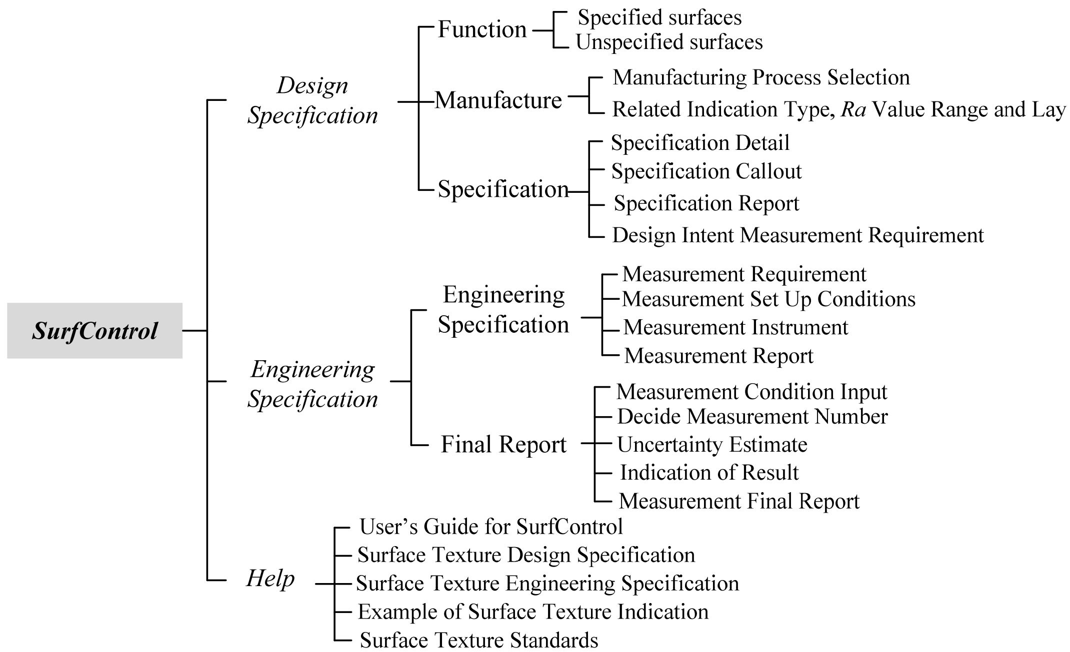


Figure 6.4 The structure of *SurfControl*

6.2.2.3 ArealControl

ArealControl is developed to provide designers and metrologists with suggested specification and verification information in AST according to the current underdeveloped standards. As was the case with *ProfileControl*, this module is composed of the five components of AST as is shown in figure 6.5.

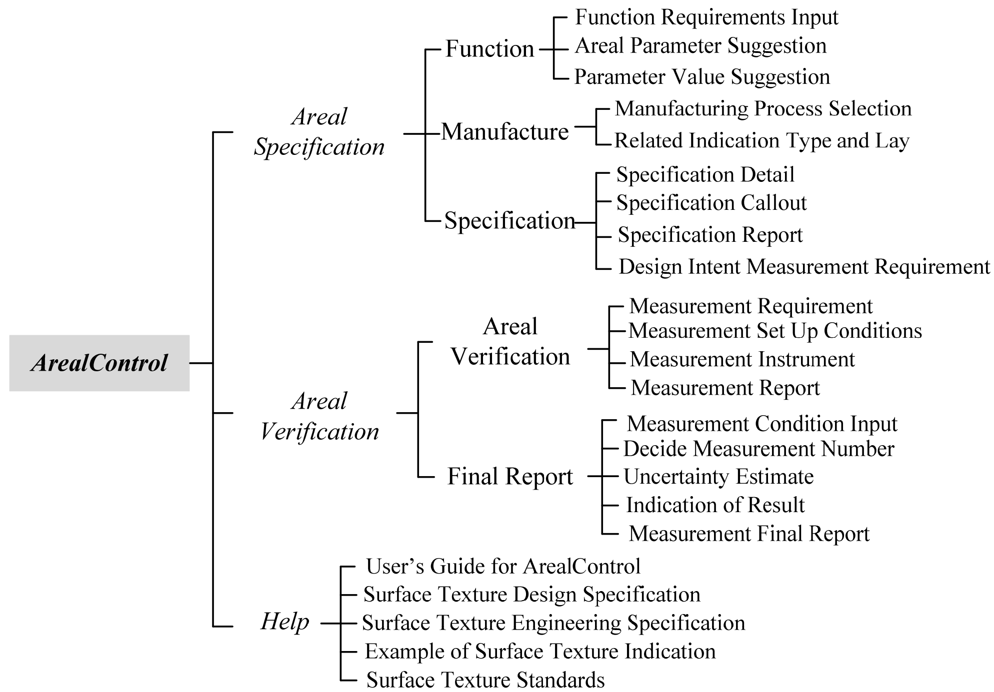


Figure 6.5 The structure of *ArealControl*

6.2.3 The categorical database

The categorical database aims to provide all the databases and relationship manipulation support for the three modules. The categorical model for profile and areal developed in chapters 4 and 5 is the foundation of the database. The design of the database is expected to:

- provide different databases for the three modules;
- provide a relationship manipulation mechanism.

The components of the categorical database are shown in figure 6.6. Module *ProfileControl* and *ArealControl* have individual databases, while *ProfileControl* and *SurfControl* share specification and verification databases (indicated with the same colour).

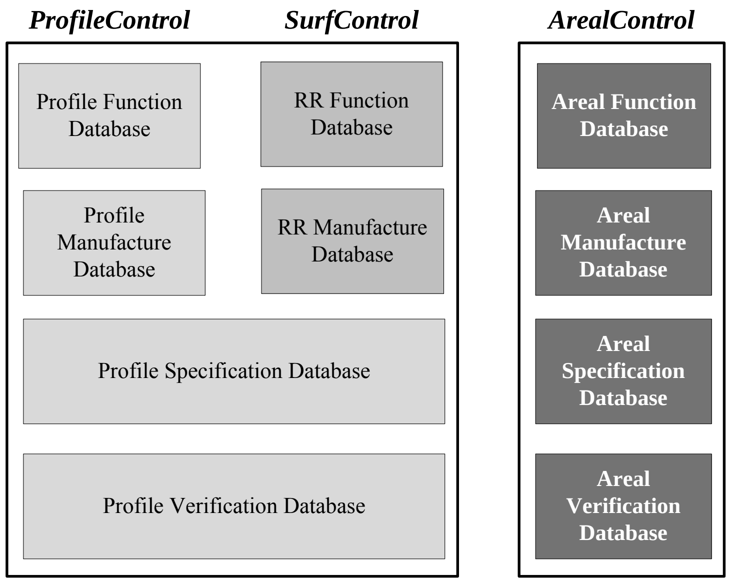


Figure 6.6 The structure of the categorical database in the CatSurf system

6.3 System Development

This section starts with a brief explanation of the tools and platform for implementation of the CatSurf system. It then moves on to demonstrate the interface of the system.

6.3.1 Tools and platform for developing the CatSurf system

With reference to the module structure design, the system is developed using Visual C++ and C#. The following tools are used in this project:

- Microsoft Visual Studio 2008 (Microsoft), Visual C# and Visual C++. The language package and platform are the main tools and platform to develop the CatSurf system;
- JfreeChart.jar plug in (JFreeChart, 2007) (Object Refinery Limited). This plug-in is used to dynamically draw various charts and diagrams for the CatSurf system;
- Microsoft Visual J# 2.0 Redistributable Package (Microsoft);
- Db4objects C# Database (Mono). This C# language database tool is used to develop the categorical database;

- Help & Manual Version 5 (EC Software GmbH). This help document tool is used to write and publish the Help document in the CatSurf system.

6.3.2 The system interface

The system interface has been developed using the development tools. As shown in figure 6.7, the interface of the CatSurf system shows three modules on the opening Menu and users can only choose one module at a time.

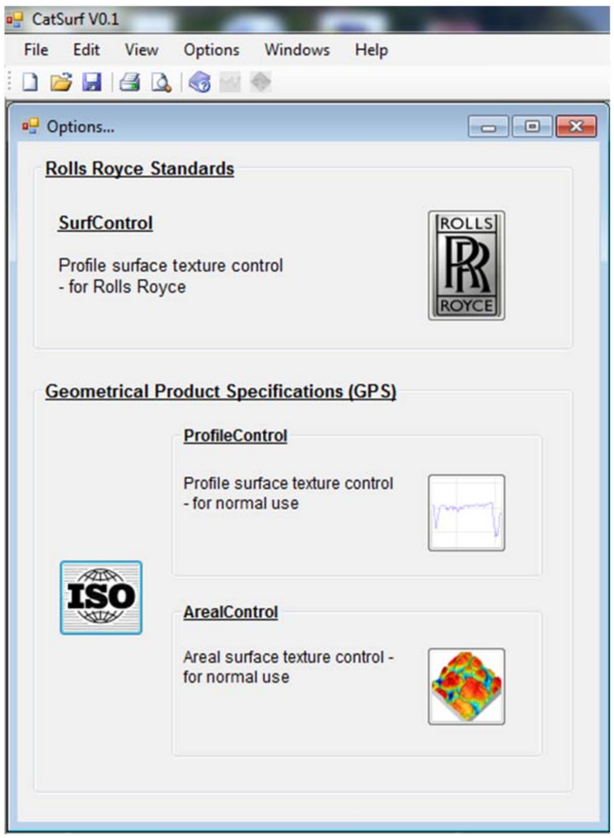


Figure 6.7 The interface of the CatSurf system

6.4 The implementation of *ProfileControl*

This section aims to demonstrate the detailed implementation of the module *ProfileControl*, the menu is shown in figure 6.8. There are three menu items in *ProfileControl*: Designers, Engineers and Help. Function, Manufacture and Specification are the sub-menu of Designers; Verification and Final Report are the sub-menus of Engineers (note that the term 'Engineers' applied here only involving with measurement tasks); the Help menu links to the Help document.

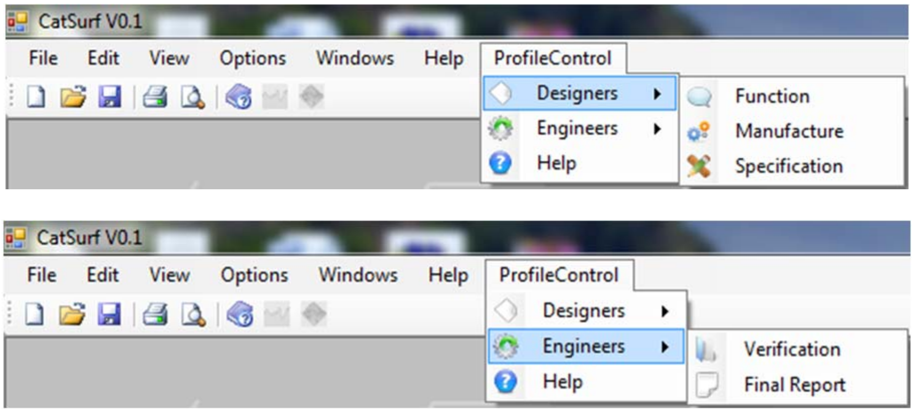


Figure 6.8 The menu of five components in *ProfileControl*

The flowchart in figure 6.9 indicates the detailed implementation processes in and between Function, Manufacture, Specification and Verification components. The program starts with the Function component.

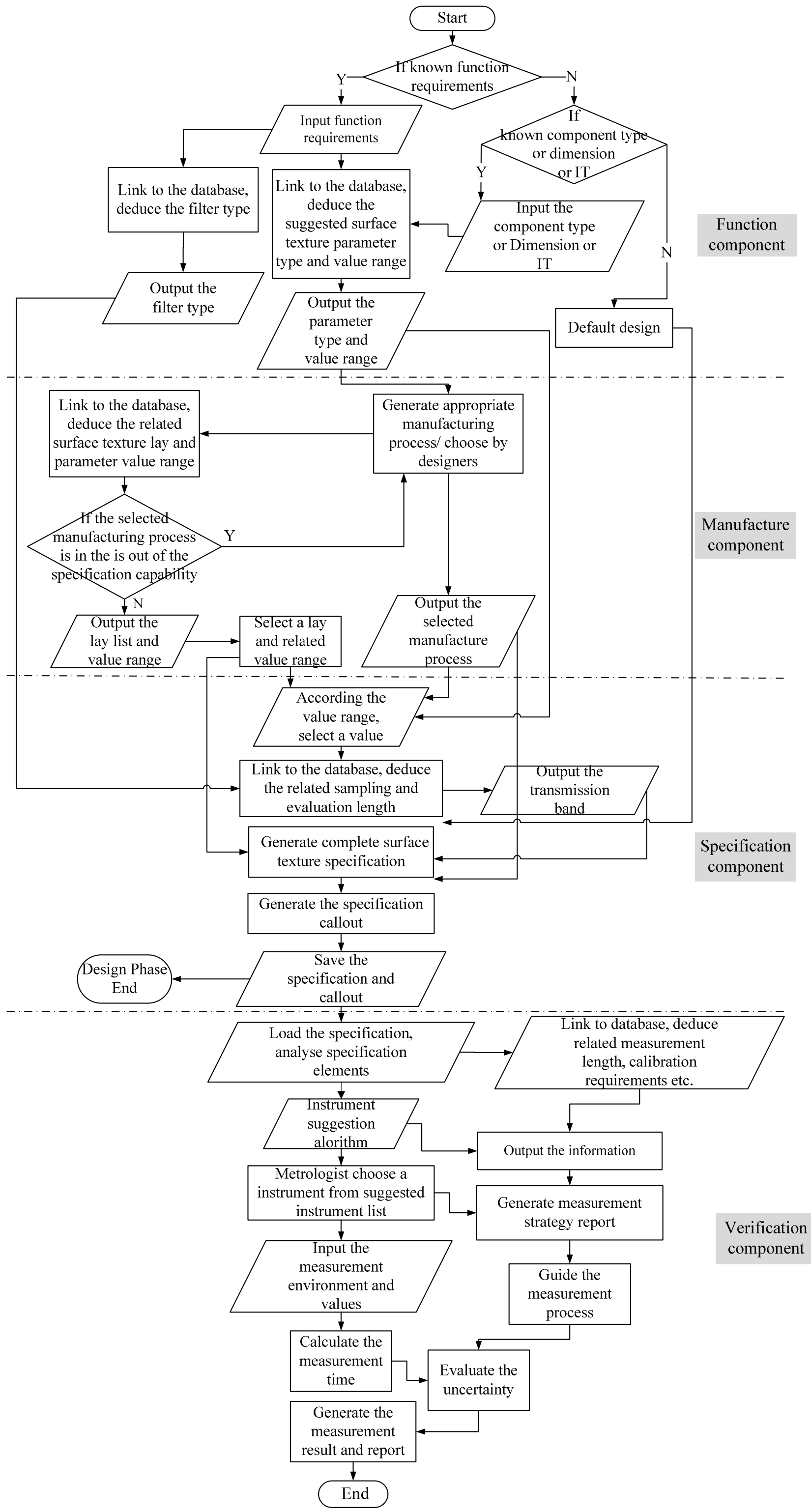


Figure 6.9 The surface texture specifications design flow chart for *ProfileControl*

6.4.1 The Function component in *ProfileControl*

The interface of the Function component in *ProfileControl* is shown in figure 6.10. The interface is composed of two groups.

- The 'Inputs' group includes the functional surfaces, the dimension of the specified surface and IT. Designers are required to select a component type such as a shaft of a cylinder from the dialog box. The dimension and IT are additional information which is non-mandatory.
- The 'Suggestion' group includes the suggested parameter type and value range. By clicking the 'Generate Suggestions' button, the system will link to the database for deducing the suggested parameter and value by utilising the relationship R_2 as shown in figure 4.13.

A part of the code for generating the relationship R_2 is shown in Appendix 1. A default parameter *Ra* is pre-indicated in the interface. By clicking the 'Next' button, the system will transfer to the Manufacture Component.

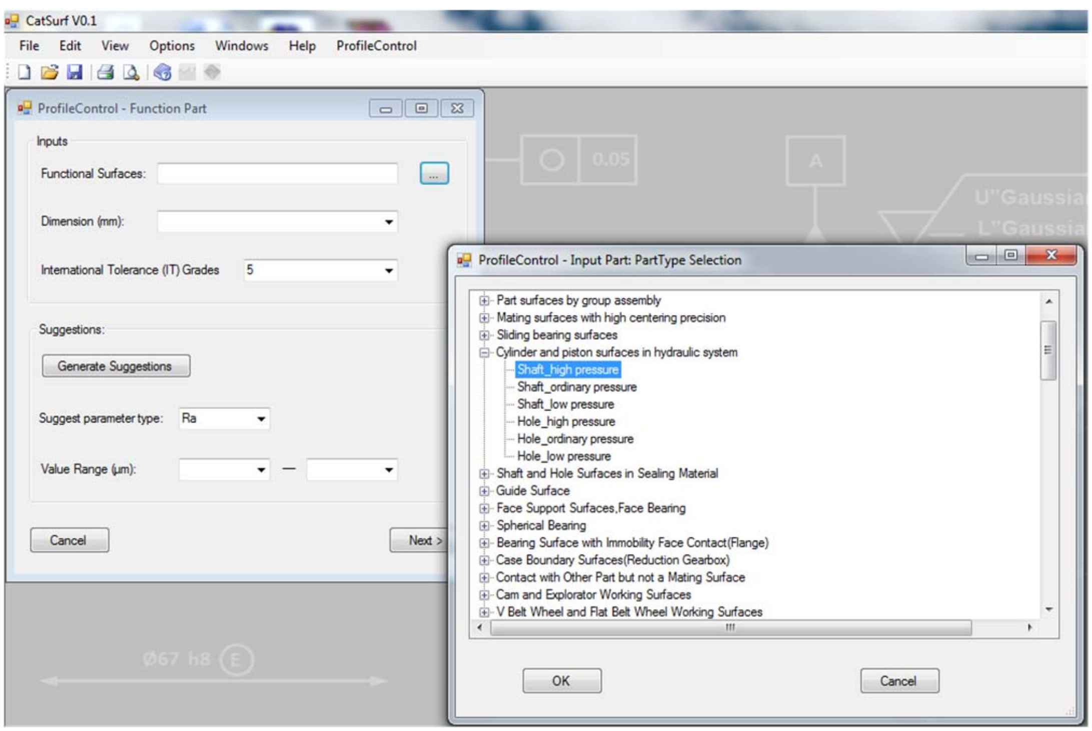


Figure 6.10 The interface of the Function component in *ProfileControl*

6.4.2 The Manufacture component in *ProfileControl*

The interface of the Manufacture component in *ProfileControl* is shown in figure 6.11. The selection of the manufacturing process is either chosen by designers or automatically generated according to the function input. After the selection of the manufacturing process, the related information such as the type of manufacturing process, related value range of the process and the possible surface texture lay of the manufacturing process will be indicated in the list, by utilising the relationship R_3 and s_5 as shown in figure 4.14. A part of the code for generating the relationship R_3 is shown in Appendix 2. Before the system transfers across to the next component, the designers are required to select one type of lay and value range from the list. When all the input and inferred results are generated, they will be sent to the Specification component for generation of a complete surface texture specification.

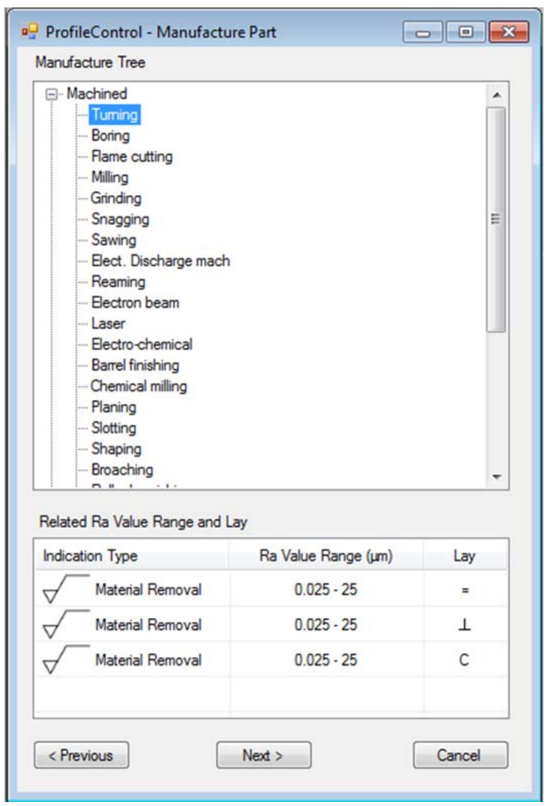


Figure 6.11 The interface of the Manufacture component in *ProfileControl*

6.4.3 The Specification component in *ProfileControl*

The interface for the Specification component in *ProfileControl* is shown in figure 6.12. It is composed of three groups: specification details, specification callout and report, and simple measurement requirements for design intent. After the designer selects a value from the suggested value range, full details of all specification elements will be generated and presented. This generation includes the utilisation of all the pullback relationships and related arrows between objects which were described in section 4.2. These specification elements can be added, deleted and modified. By clicking the 'Detail' button or double click on the specification elements, the specification details interface will be shown (see figure 6.13). In the interface for the specification details, designers can choose different profile parameters, or modify other specification elements such as limit value, filter type and transmission band. However, all modifications should be consistent with the relationship designed in the categorical database. An example is shown in figure 6.14 where the limit value has been changed from 0.2μm to 6.3μm, and a warning message is shown stating "This limit value is out of manufacturing process range, please reselecting a value."

After the modification of the specification details, clicking the 'Generate Specification Callout and Report' button will result in the specification callout and report being generated and presented in the interface. The specification can be saved or open by XML format. The specification report includes every detail of the specification. On the right side of the interface is the measurement requirement for design intent, this contains some basic measurement information for the assigned specification will be presented to give the designers basic information of measurement. It includes measurement direction, measurement length and traverse length, suggested instrument type, tip radius of a contacting stylus and sampling spacing, and calibration requirements of measurement standards. The results are obtained from the utilisation of the relationships R_9 and R_{13} in the verification model of PST. A part of the code for generating the relationship R_9 and R_{13} is described in Appendix 3.

The designer or metrologist will access the Verification component by clicking the 'Verification' button, and all of the designed specification data will be transferred to the next step.

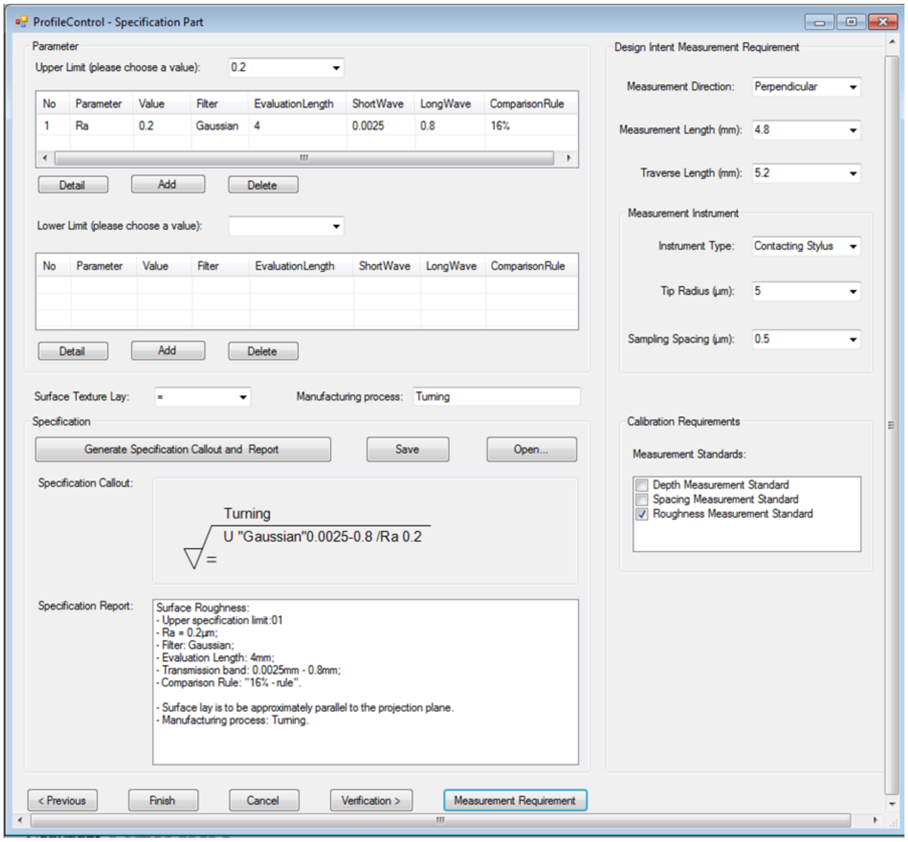


Figure 6.12 The interface of the Specification component in *ProfileControl*

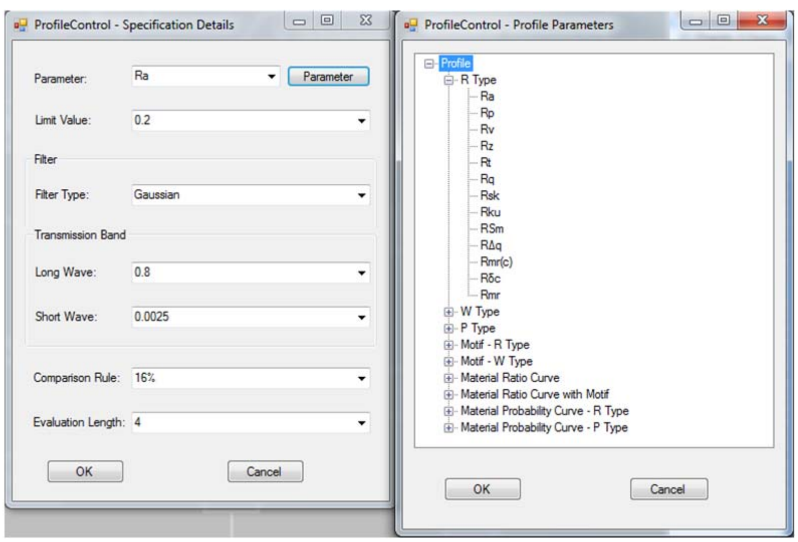


Figure 6.13 The details of the specification elements interfaces

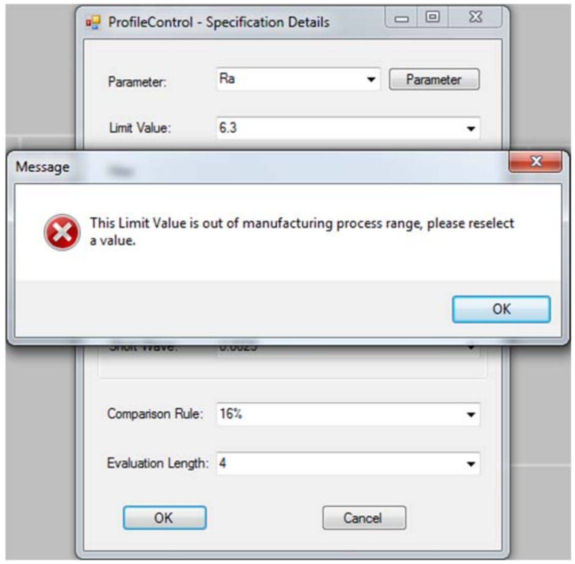


Figure 6.14 The warning message

6.4.4 The Verification component in *ProfileControl*

The interface for the Verification component is shown in figure 6.15. This component starts with the analysis of the assigned specification either by opening a saved specification XML file, or the same specification which is transferred from the Specification component. After the analysis, the measurement set up conditions, calibration requirements and measurement length will be generated and shown in the interface. Using the instrument suggestion algorithm, an amplitude-Wavelength Diagram is shown in the interface. In the diagram, the point coordinates of limit value and sampling spacing are indicated. Determined by coordinate, related instrument suggestions are given underneath the diagram. By clicking the 'Instrument Detail' button, the list of instrument suggestion types will be shown (see figure 6.16). In the interface of the instruments list, instruments can be added to the diagram which are appropriate. After the selection of the instrument, the measurement strategy report will be generated by clicking 'Generate Measurement Requirement Report'.

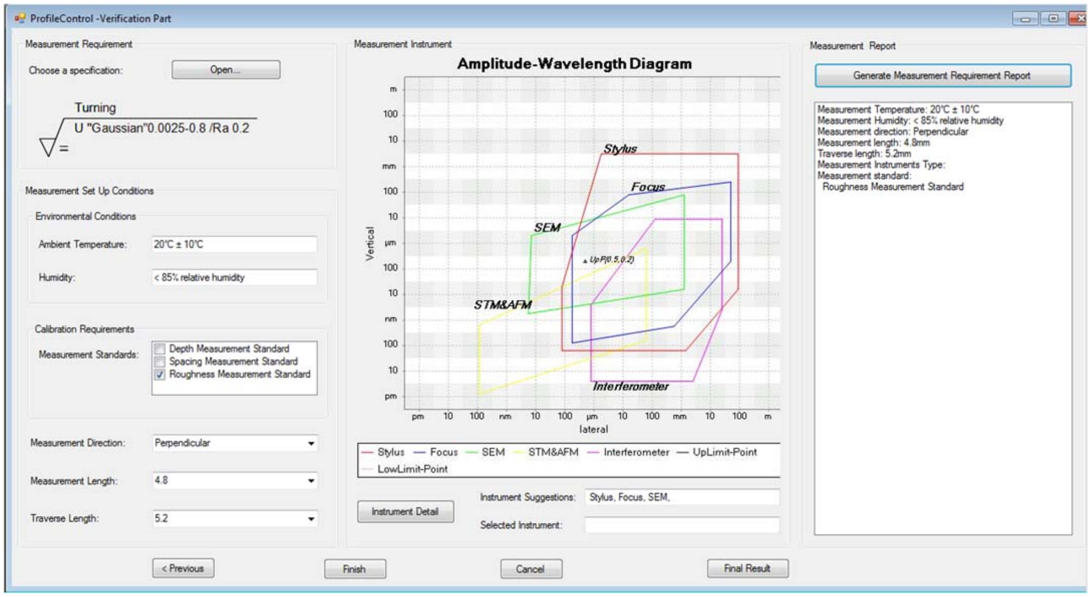
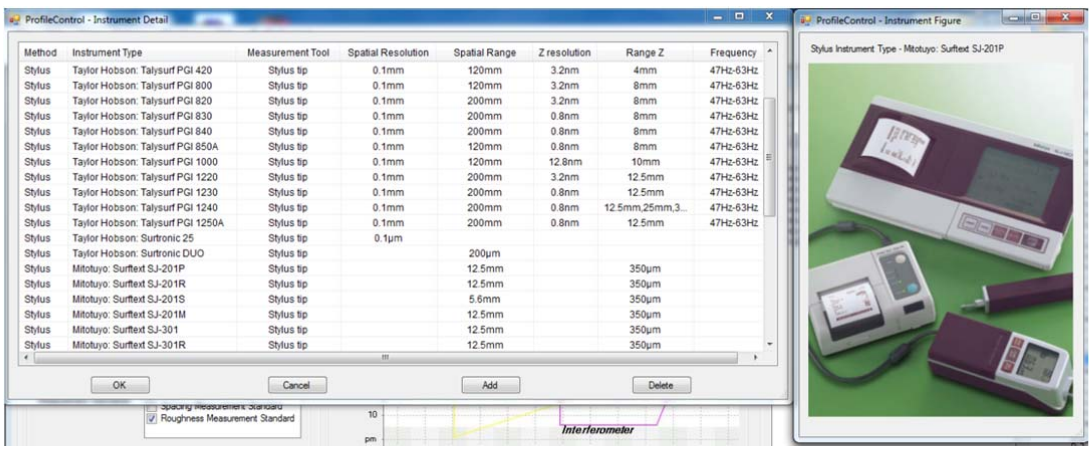


Figure 6.15 The interface of the Verification component in *ProfileControl*



Method	Instrument Type	Measurement Tool	Spatial Resolution	Spatial Range	Z resolution	Range Z	Frequency
Stylus	Taylor Hobson: Talysurf PGI 420	Stylus tip	0.1mm	120mm	3.2nm	4mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 800	Stylus tip	0.1mm	120mm	3.2nm	8mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 820	Stylus tip	0.1mm	200mm	3.2nm	8mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 830	Stylus tip	0.1mm	200mm	0.8nm	8mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 840	Stylus tip	0.1mm	200mm	0.8nm	8mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 850A	Stylus tip	0.1mm	120mm	0.8nm	8mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 1000	Stylus tip	0.1mm	120mm	12.8nm	10mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 1220	Stylus tip	0.1mm	200mm	3.2nm	12.5mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 1230	Stylus tip	0.1mm	200mm	0.8nm	12.5mm	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 1240	Stylus tip	0.1mm	200mm	0.8nm	12.5mm,25mm,3...	47Hz-63Hz
Stylus	Taylor Hobson: Talysurf PGI 1250A	Stylus tip	0.1mm	200mm	0.8nm	12.5mm	47Hz-63Hz
Stylus	Taylor Hobson: Surtronic 25	Stylus tip	0.1μm				
Stylus	Taylor Hobson: Surtronic DUO	Stylus tip		200μm			
Stylus	Mitotuyo: Surftext SJ-201P	Stylus tip		12.5mm		350μm	
Stylus	Mitotuyo: Surftext SJ-201R	Stylus tip		12.5mm		350μm	
Stylus	Mitotuyo: Surftext SJ-201S	Stylus tip		5.6mm		350μm	
Stylus	Mitotuyo: Surftext SJ-201M	Stylus tip		12.5mm		350μm	
Stylus	Mitotuyo: Surftext SJ-301	Stylus tip		12.5mm		350μm	
Stylus	Mitotuyo: Surftext SJ-301R	Stylus tip		12.5mm		350μm	

Figure 6.16 The instrument details

The measurement strategy will be used to guide the measurement. Accessing to the 'Final Report' interface, the metrologist is required to record all measurement conditions such as the tip radius, measurement speed, traverse length, temperature, humidity, instrument name, calibration type, measurement data and name of the operator. After the uncertainty is estimated, the metrologist can input the measurement values in the 'Decide Measurement Number' group; the system will generate the measurement decision using the comparison procedure which was demonstrated in section 4.3.3.4. Although the uncertainty estimate function is provided in the 'Final Report', the function is currently not available as there is currently no effective method to estimate the uncertainty of surface texture measurement. The 'Indication Result' group provides a profile view and Gaussian filtering capability for measurement data that is in SDF format (ISO 5436-1, 2001; ISO 5436-2, 2012). The 'Measurement Final Report' group provides a report containing all of the measurement information to ensure measurement traceability.

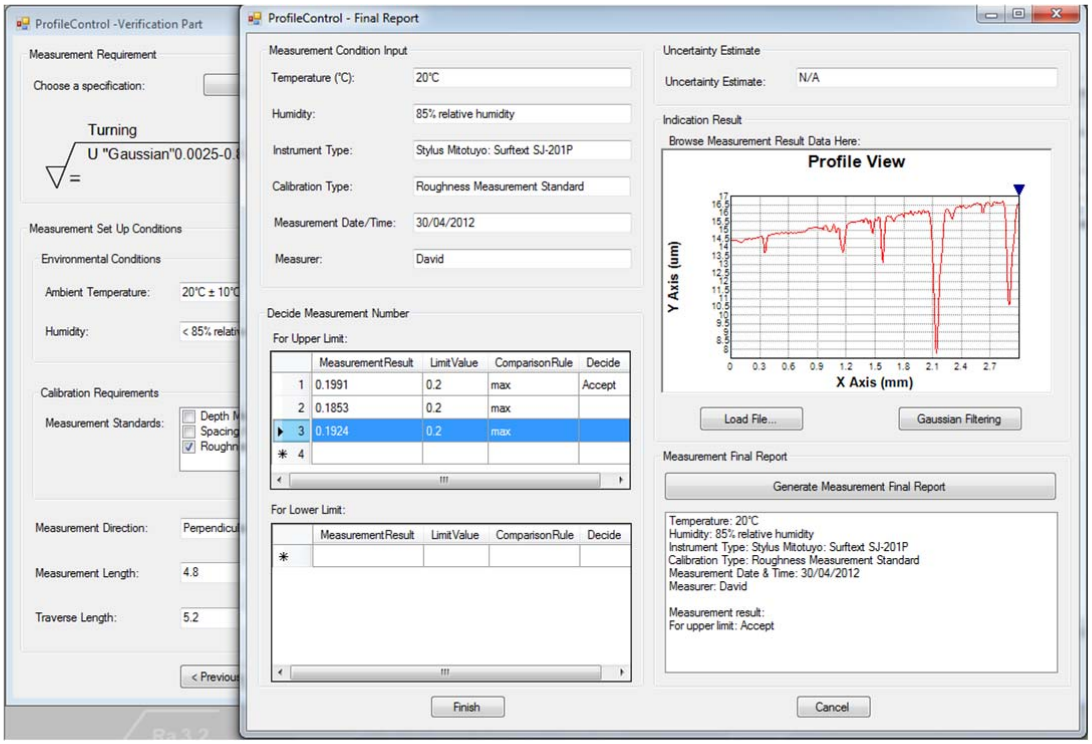


Figure 6.17 The interface of Final Report in *ProfileControl*

6.5 The implementation of *SurfControl*

This section aims to demonstrate the detailed implementation of the *SurfControl* module. As *SurfControl* is a special case of *ProfileControl*, the module inherits a major part of its methodology from *ProfileControl* but is different in several details such as the function requirement and Manufacture component (both of which will be more specific).

The flowchart shown in figure 6.18 indicates detailed implementation processes in and between the Function, Manufacture, Specification and Verification components. As the majority of functionality is shared with *ProfileControl*, and has been previously described, here details will only be given of the operational differences with *ProfileControl*.

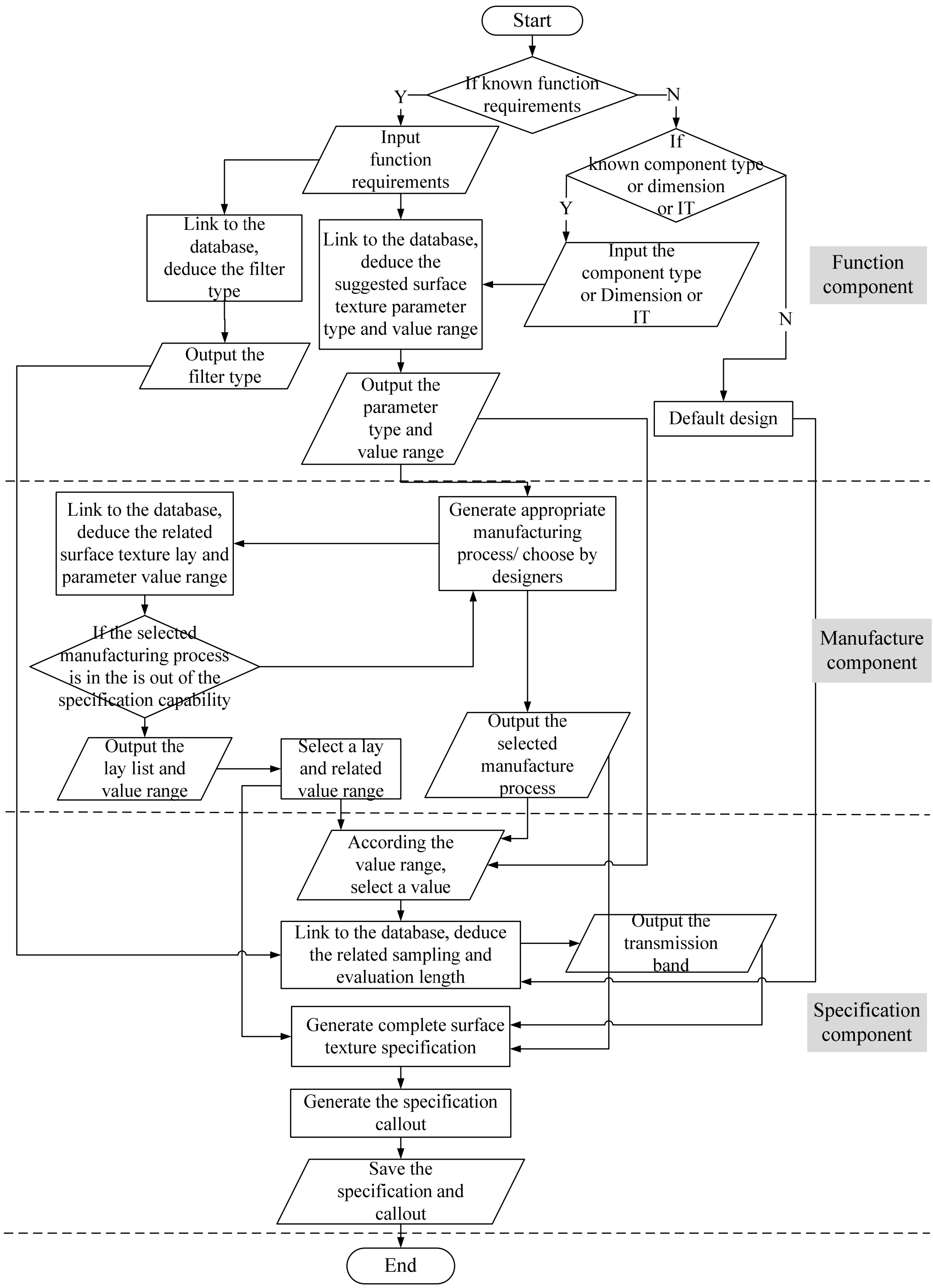


Figure 6.18 The surface texture specifications design flow chart for *SurfControl*

The programme starts with the Function component. The implementation is as follows:

- adjudge whether the functional surface is known. If yes, choose the function surface. If not, go to unspecified surfaces;
- after the selection of the functional surface, select one type of materials;
- link to the database, deduce the suggested parameter, value and manufacturing process;

The program then moves to the Manufacture component which is common with the *ProfileControl* programme. The difference between the Specification component of *SurfControl* compared to that of *Profilecontrol* is that the designers do not have to select a value from the value range as an assigned value will be given at the Function step.

6.5.1 The Function component in *SurfControl*

The interface of the Function component in *SurfControl* is shown in figure 6.19. In this interface, there are two types of inputs which are specified or unspecified surfaces. The former is for designed surfaces with specific surface texture requirements, and the latter is for general surfaces with no/low surface texture requirements.

Specified surfaces include functional surfaces such as thrust face, machined air flow surfaces or component surfaces defined as surfaces of specific engine components. Designers who select one of the specified surfaces should also choose a material such as steel or aluminium. If the designer chooses the wrong material, a warning message dialog box will activate.

Certain readily identifiable features will not normally have a surface texture value specified on the component definition. These surfaces are specified under ‘unspecified surface’ group. Unspecified surfaces include machined surfaces such as rolled screw threads and keyways, and unmachined surfaces such as cast or forged surfaces.

In this programme no matter which kind of surface the designer has elected, the specified parameter will be *Ra* only, and a related value will be assigned according to the database.

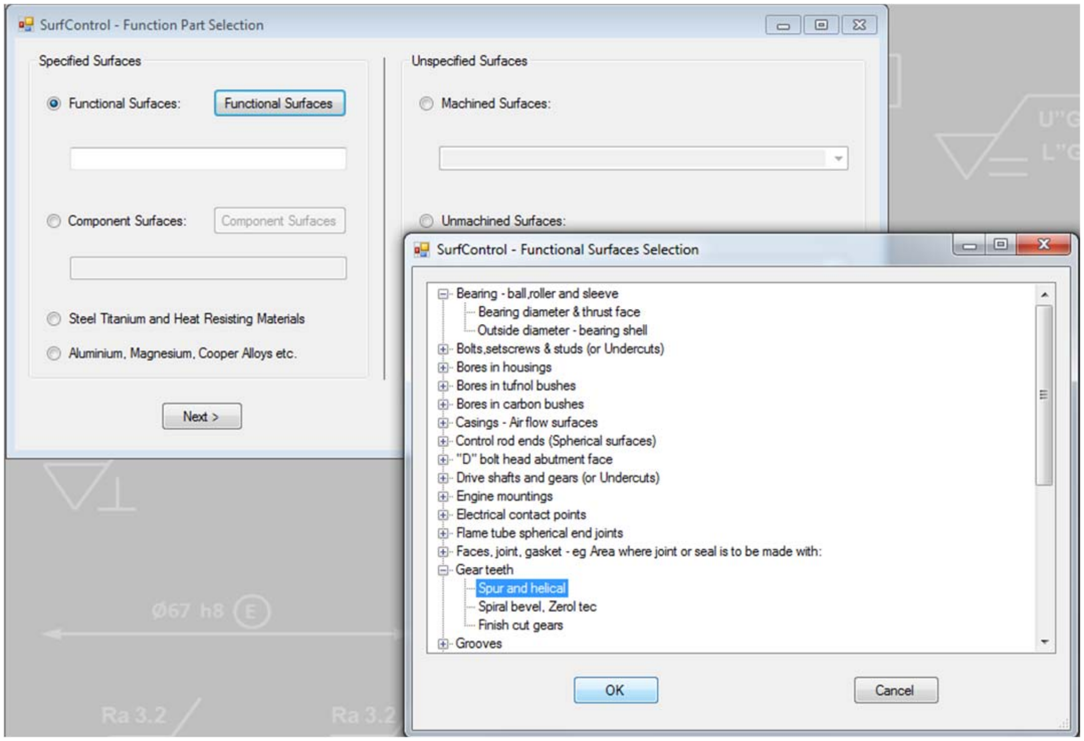


Figure 6.19 The interface of the Function component in *SurfControl*

6.5.2 The Manufacture component in *SurfControl*

The Manufacture component in *SurfControl* is in general the same as *ProfileControl* excepting the inclusion of a number of manufacturing processes as shown in figure 6.20.

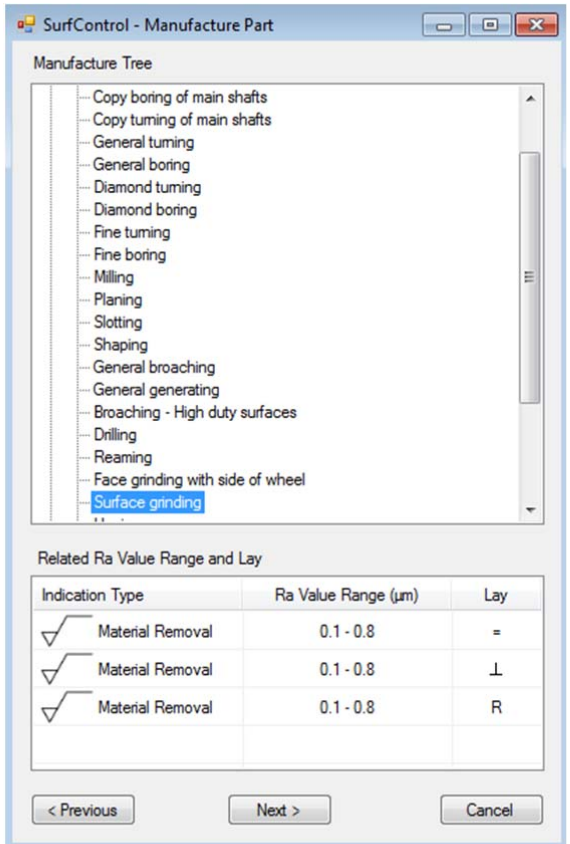


Figure 6.20 The interface of Manufacture component in *SurfControl*

6.5.3 The Specification component in *SurfControl*

The *Specification* component in the *SurfControl* programme is also generally the same as that in *ProfileControl* except that there is no utility to select a value range as shown in figure 6.21. In addition, the function for changing parameters is disabled.

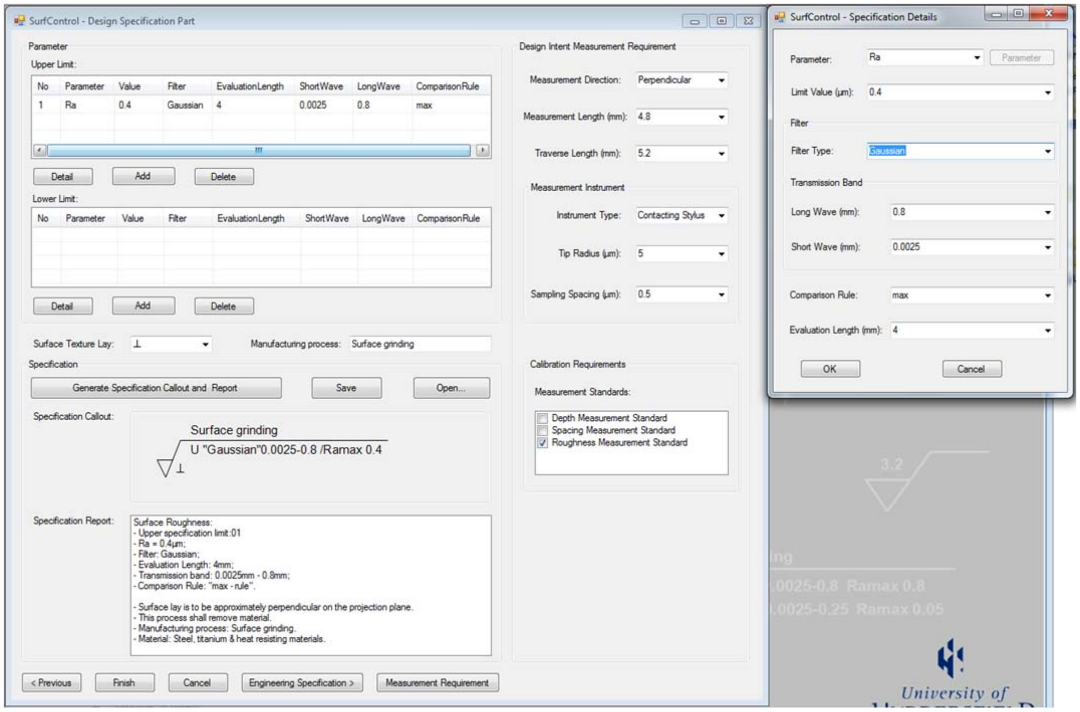


Figure 6.21 The interface of the Specification component in *SurfControl*

The Verification component in *SurfControl* is exactly the same as in *ProfileControl* (Section 6.3.4).

6.6 The implementation of *ArealControl*

This section aims to demonstrate the detailed implementation of the module *ArealControl*. The menus of five components are divided in the same way as they are *ProfileControl*.

6.6.1 The Function component in *ArealControl*

The interface of the Function component is shown in figure 6.22. The interface is composed of two groups which are 'Function requirements input' and 'Surface Texture Parameter and Value Suggestions'. The 'Function requirements input' includes sheet materials for automotive applications as a case study, and other supplementary applications. It is envisaged that more applications will be added to this as part of future work. Figure 6.22 illustrates an example where the function requirement is 'Oil retention during storage of the sheet materials'. By analysing the input, the database recalls the relationship AP_3 in the AST categorical model, the consequent suggested parameter is *Sda*(*c*) and suggested value for this is 'FC;D;Wolf:5%;Edge:50%;Area;Mean'.

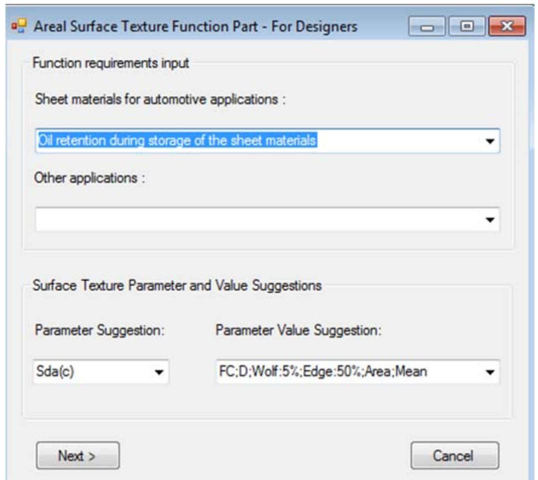


Figure 6.22 The interface of the Function component in *ArealControl*

6.6.2 The Manufacture component in *ArealControl*

The Manufacture component in *ArealControl* is generally the same as in *ProfileControl* except there is no parameter value range shown in the interface (see figure 6.23).

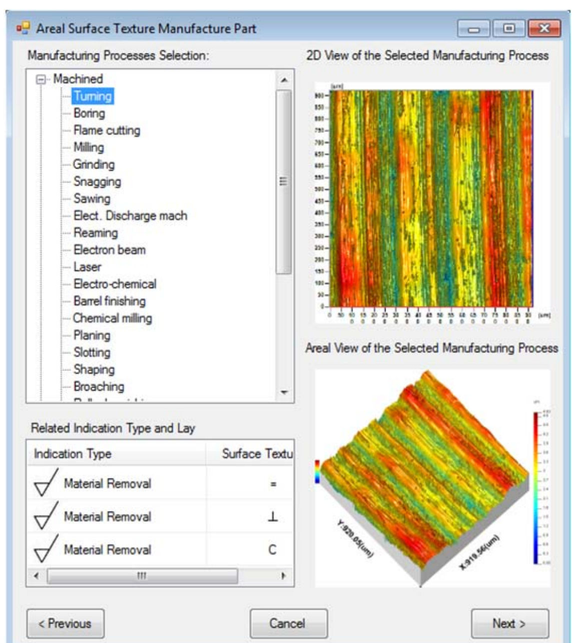


Figure 6.23 The interface of the Manufacture component in *ArealControl*

6.6.3 The Specification component in *ArealControl*

The *Specification* component in *ArealControl* is similar to that in *ProfileControl*. However, as areal specification is different from profile specification, the specification details, callout, and measurement requirements differ as shown in figure 6.24. The specification detail of *ArealControl* as shown in figure 6.25 is designed based on the areal parameters defined in ISO 25178-2:2012. Most of the elements in the ‘Parameter’ interface are designed according to the objects in the category **ATD**. If different areal parameters are chosen in place of the default/suggested parameters, the related attribute, default value and unit of the parameter will be shown in the interface to give information about the parameter. Using the same principle, it is not permitted to create specification details which are not consistent with the relationship constraints defined by the pullbacks and arrows detailed in chapter 3.

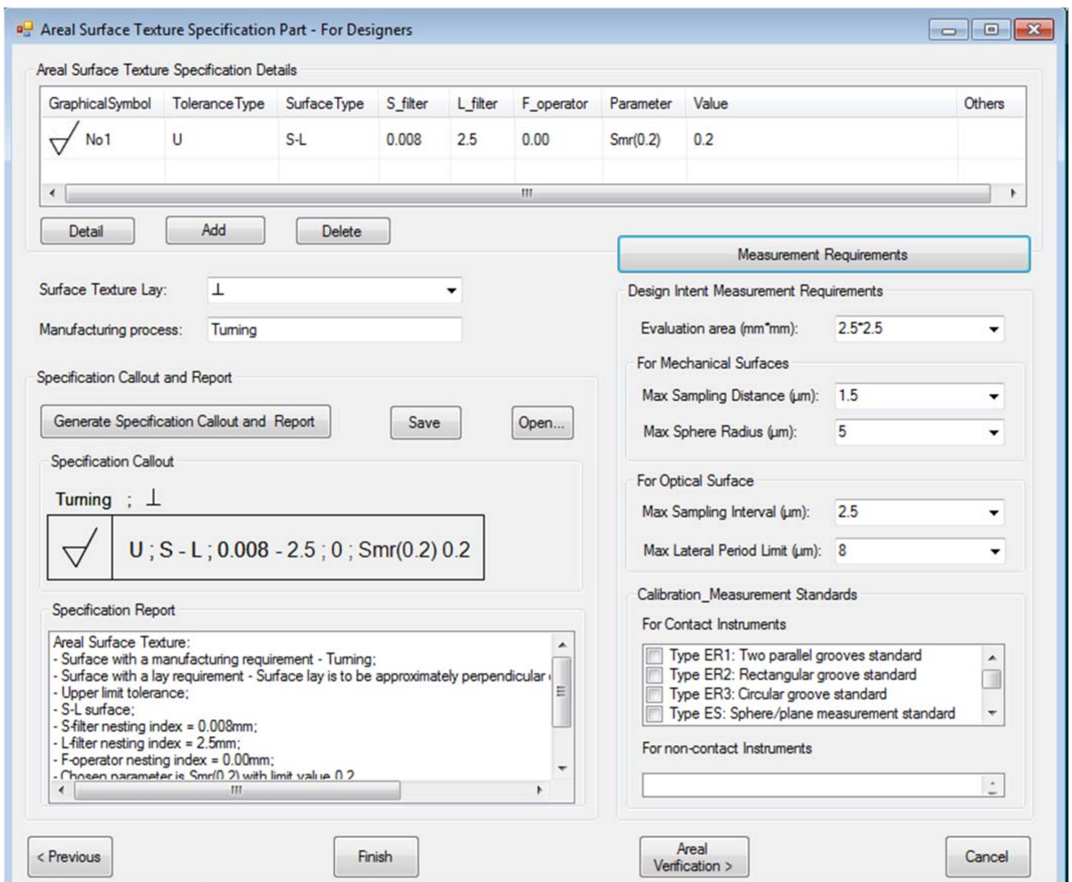


Figure 6.24 The interface of the Specification component in *ArealControl*

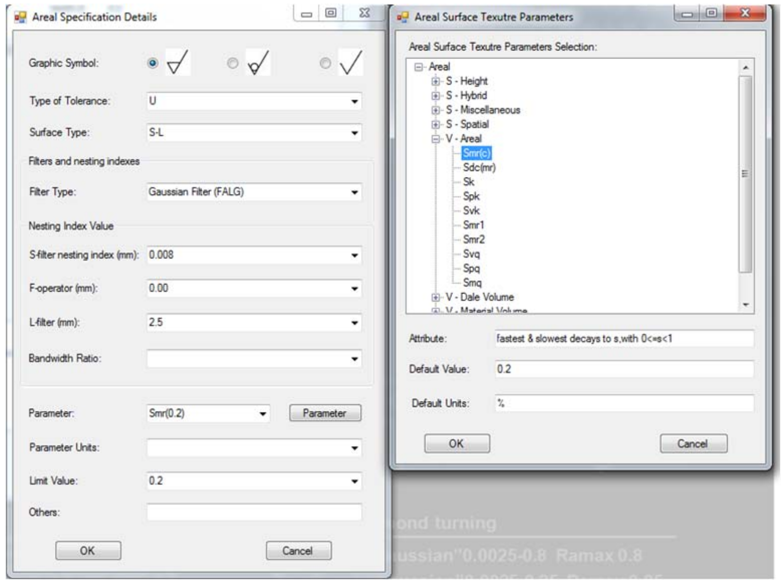


Figure 6.25 The interface of specification details in *ArealControl*

6.6.4 The Verification component in *ArealControl*

The *Verification* component in *ArealControl* is similar to that in *ProfileControl* (Section 6.3.4) apart that is from the calibration requirements and evaluation area as illustrated in figure 6.26 and 6.27.

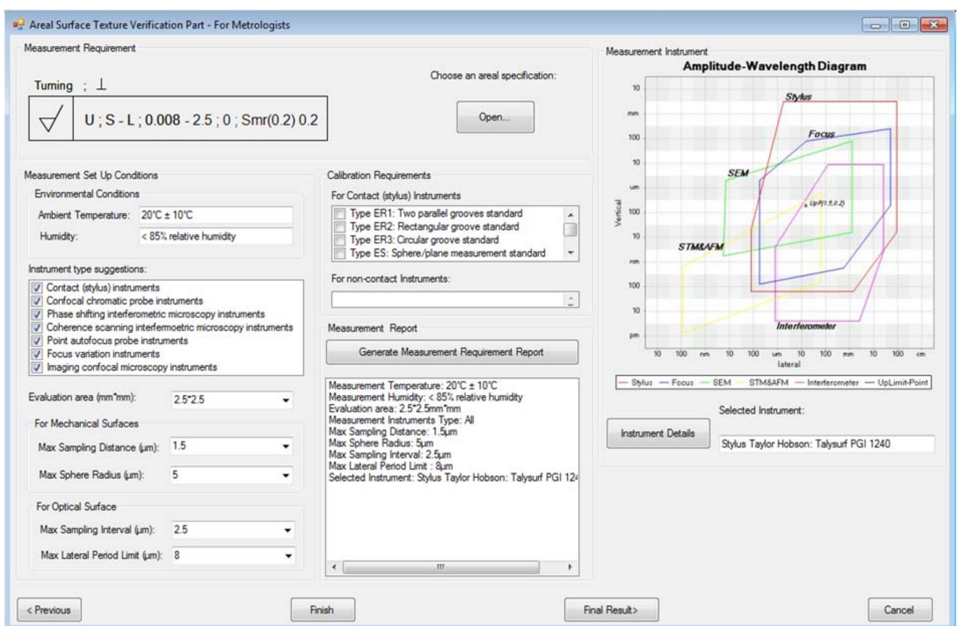


Figure 6.26 The interface of Verification component in *ArealControl*

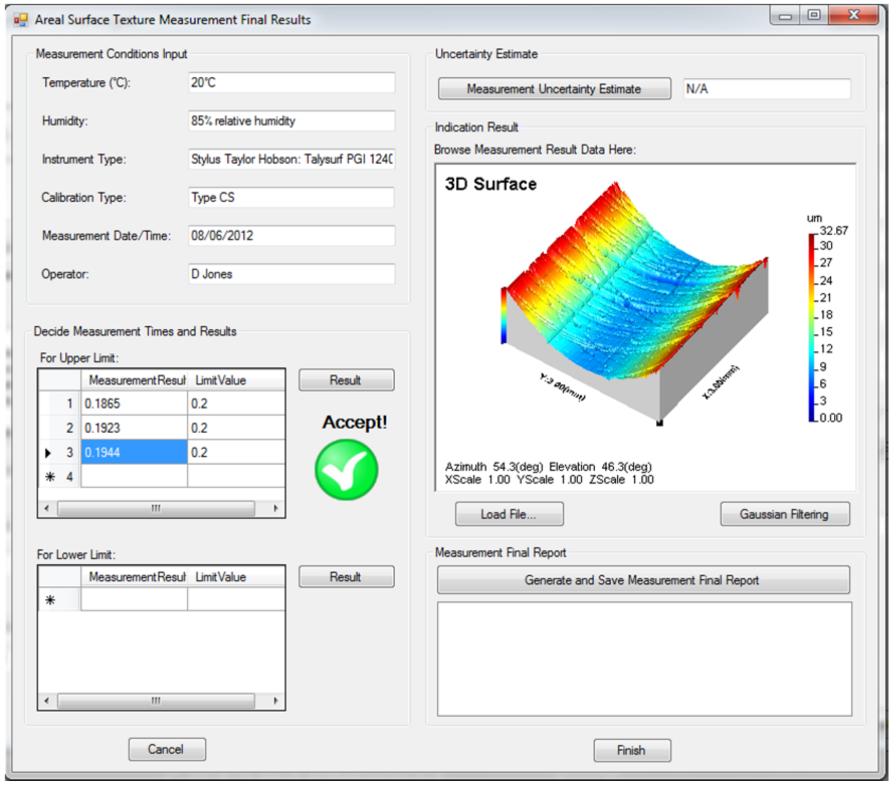


Figure 6.27 The interface of Final Report in *ArealControl*

6.7 The implementation of the Help document

This section will demonstrate the development of Help documents in the CatSurf system. The help documents were developed through use of Help & Manual Ver 5. The help document is implemented in accordance with the structure shown in figure 6.2 and one of the interfaces can be seen in figure 6.28. The 'User's' Guide includes every detail of how to use the system, as well as the explanation of every term in every interface. 'Surface Texture Design Specification' includes every detail of how to design a complete surface texture specification according to the functional requirements. 'Surface Texture Engineering Specification' includes details of how to measure surface texture according to the assigned specification. Figure 6.28 - 6.30 show the interface in different modules.

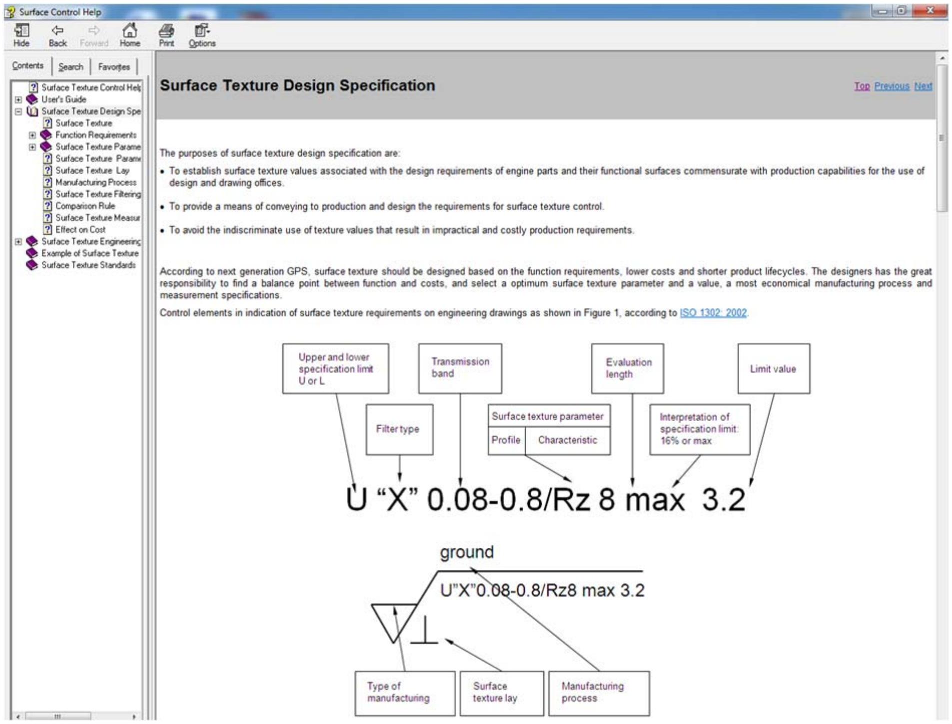


Figure 6.28 The interface of Help document in *ProfileControl*

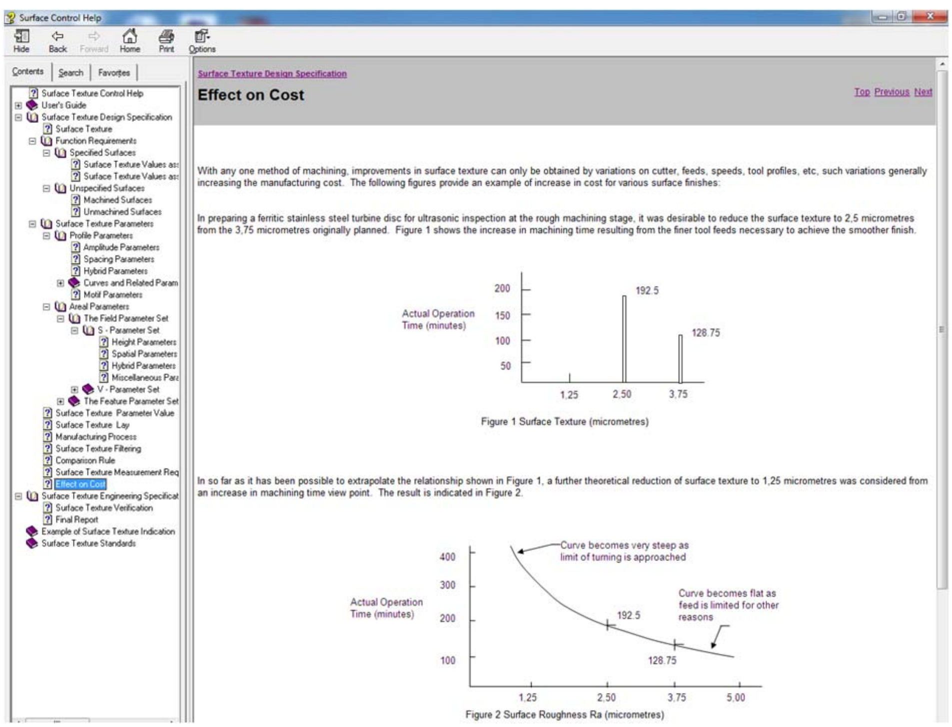


Figure 6.29 The interface of Help document in *SurfControl*

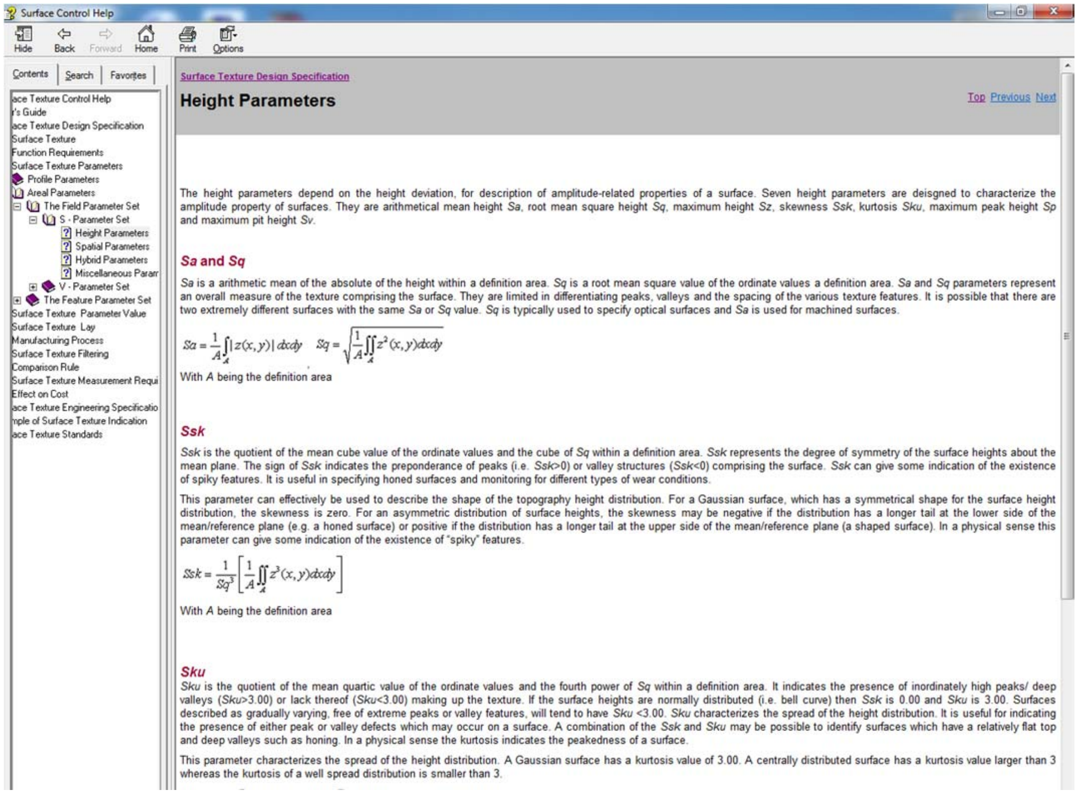


Figure 6.30 The interface of Help document in *ArealControl*

6.8 Conclusions

This chapter has designed the architecture of the CatSurf system. A prototype system has been developed and the implementation of three modules each of five components was presented. Currently it is an executable program which can be integrated with CAx systems, and the integration methodology will be introduced in the next chapter.

7. The integration between CatSurf and CAD systems

This chapter records in detail the integration between the CatSurf system and CAD systems. The methodology and implementation of the integration, as well as two test cases are demonstrated.

7.1 Integration methodology

This section aims to demonstrate the methodology of integration between CatSurf and CAD systems. A universal XML based approach for integrating CAD and CatSurf is proposed. As shown in figure 7.1, the designed specifications are saved to XML files according to a specified format (details will be described in the next section) in CatSurf. By reading the XML files, transferring the specification data to a CAD database, and executing the command from the interface in the CAD, an interface application program is developed to integrate CAD and CatSurf. As a part of the interface application program, two embedded function menus are developed. The menu ‘Surface Texture Control’ is used to open CatSurf for surface texture specification design. The menu ‘Surface Texture Drawing’ is used to read and analyse the saved XML file, translate the specification data to CAD systems, then generate the surface texture indications in the CAD drawing space. Sharing the same address space and making direct function calls, the interface application is programmed by specialised software development tools provided by different CAD systems, for example, ObjectARX (AutoCAD Runtime Extension) (Autodesk) is an API (application programming interface) for customizing and extending AutoCAD, and UG/Open is a development tool for UX.

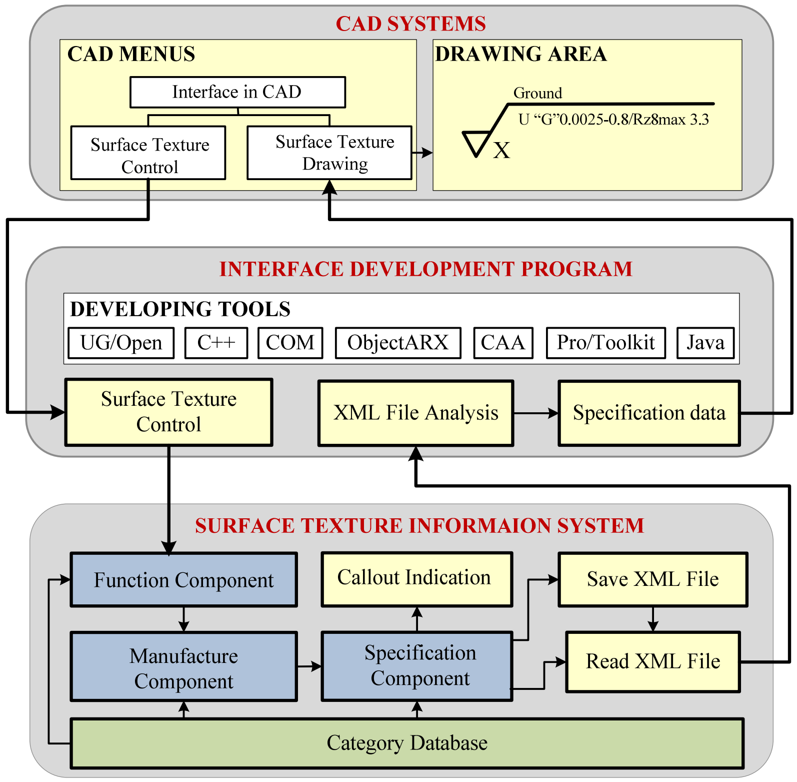


Figure 7.1 The integration methodology

7.2 XML Schema

This section aims to demonstrate in detail the XML schema of surface texture specification. While file formats that are currently in wide use such as the SDF format cover the representation of discrete data points along with some header information, they do not convey information about the measurement operation, the manufacturing process or the functional requirements of the component. In this section, we justify the choice of XML related technologies to represent surface texture information in GPS. As a markup language, XML provides the standard format for structured document/data exchange. The simplicity, generality and usability of XML makes it easy to solve interoperability problems. XML provides distributed computing with a set of well-defined standards for electronic transfer of data/documents in application-to-application, business-to-business, and application-to-human communications (Rezayat, 2000).

This section shows how to represent surface texture information using XML schema in multiple layered conformance levels to meet different application domains' requirement according to the properties of XML schema and the requirements from GPS in representing surface texture.

Based on the XML Schema file, the user can construct an appropriate XML data file to meet requirements. Such an XML file can be used in GPS and is suitable for web-based application. In addition, XML files can be imported into CAx systems, making it easier to transfer data between different stages of production, such as CAD systems in the design stage, CAM and CAPP (computer-aided process planning) in the manufacturing stage and CAT (computer-aided tolerancing) in the measurement/inspection step.

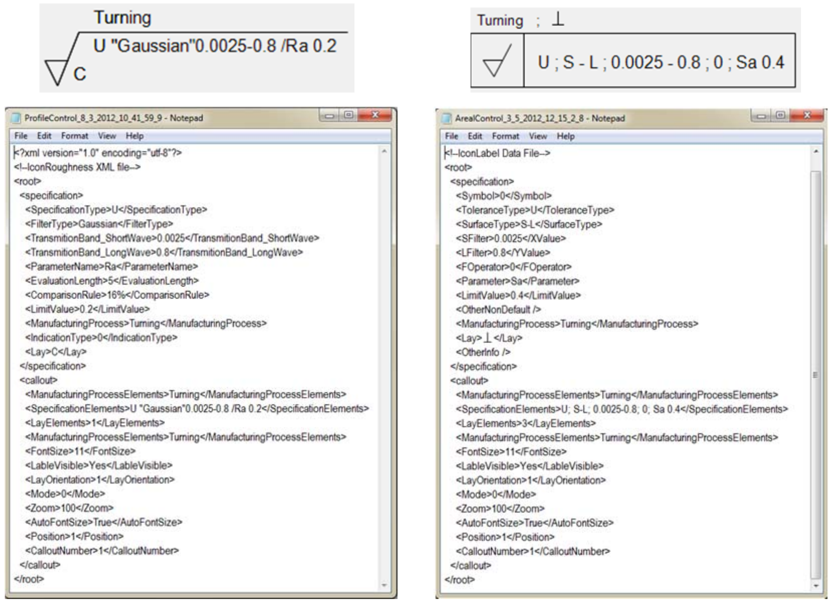
Figure 7.2 XML schema for PST

There are four levels in the XML schema. Figure 7.2 gives an example of XML schema in *ProfileControl*. From top to bottom, the first level presents specifications in different modules such as *ProfileControl* or *ArealControl*. The second level separates specification details and indication data. The ‘Specification’ includes every specification element in a specification; and the ‘Callout’ includes the elements and attributes of indication such as the font size and position. The third level is the detail of every element in 'Specification' and 'Indication'. The fourth level gives the data types of the elements in the third level. The details of every level of *ArealControl* are shown in table 7.1.

Table 7.1 Four levels of the XML schema for *ArealControl*

First Level	*Second Level*	*Third Level*	*Fourth Level*
Areal surface texture	Specification	Symbol	unsignedByte
		ToleranceType	string
		SurfaceType	string
		SFilter	decimal
		FOperation	decimal
		LFilter	decimal
		Parameter	string
		LimitValue	string
		OtherNonDefault	string
		ManufacturingProcess	string
		Lay	string
		OtherInfo	string
	Callout	ManufacturingProcessElements	string
		SpecificationElments	string
		LayElements	string
		OtherInformaiton	string
		FontSize	unsignedByte
		LabelVisible	string
		LayOrientation	unsignedByte
		Mode	unsignedByte
		Zoom	unsignedByte
		AutoFontSize	string
		Position	unsignedByte
		CalloutNumber	unsignedByte

Once the designers click the 'Save' button in the Specification component of every module in CatSurf, the system extracts the specification details and converts them into a XML file following the proposed schema. Figure 7.3 shows two example XML files for indications *ProfileControl* and *ArealControl* respectively.



(a) (b)

Figure 7.3 The XML files for indications of *ProfileControl* (a) and *ArealControl* (b)

7.3 Integration programming and interface

This section aims to demonstrate the implementation of integration. It will start with the tools and platform which have been used for implementation. It then moves to programming achievements which includes programming the integration with AutoCAD and SolidWorks.

7.3.1 Platform and tools

As discussed in chapter 2, most current commercial CAD systems such as AutoCAD, SolidWorks, Pro/Engineer employ the surface texture model only as an indication tool. This thesis mainly focuses on the integration of CatSurf with AutoCAD and SolidWorks. The integration with other CAx systems will be implemented in future work. The tools and platforms used are as follows:

- AutoCAD 2011;
- SolidWorks 2009;
- Microsoft Visual Studio 2008, Visual C++ and Visual C#. Visual C++ language package and platform are the main tools and platform to develop the interface program in AutoCAD 2011. Visual C# language package and

platform are the main tools and platform to develop the interface program in SolidWorks 2009.

- ObjectARX 2011.

7.3.2 Programming achievement in AutoCAD

Two sections have been developed in the interface programme for AutoCAD 2011. The first of which is the interface to connect to the CatSurf system. AutoCAD users using this section have access to the CatSurf system to assign the surface texture specification. When users finish the specification design, the saved XML file will be sent back to the interface program. The menu of the two parts is shown in figure 7.4. The menu in AutoCAD is developed using COM (Component Object Model) and is used to enable interprocess communication and dynamic object creation in a large range of programming languages.

Using COM component to build a menu in AutoCAD system, the following is part of the programming code.

```
…
CAcadApplication IAcad(acedGetAcadWinApp()->GetIDispatch(TRUE));
CAcadMenuBar IMenuBar(IAcad.get_MenuBar());
long numberOfMenus;
numberOfMenus = IMenuBar.get_Count();
CAcadMenuGroups IMenuGroups(IAcad.get_MenuGroups());
VARIANT index;
VariantInit(&index);
V_VT(&index) = VT_I4;
V_I4(&index) = 0;
CAcadMenuGroup IMenuGroup(IMenuGroups.Item(index));
CAcadPopupMenus IPopUpMenus(IMenuGroup.get_Menus());
CString cstrMenuName = _T("Surface Texture");
VariantInit(&index);
V_VT(&index) = VT_BSTR;
V_BSTR(&index) = cstrMenuName.AllocSysString();
…
```

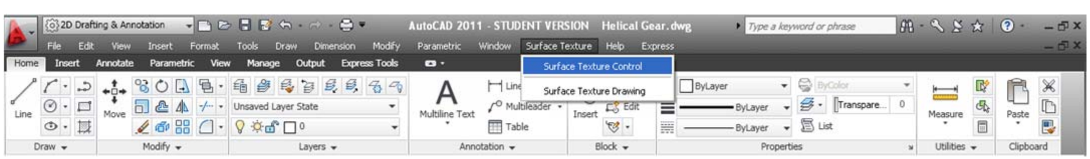


Figure 7.4 The embedded menu interface in AutoCAD 2011

The second part is programmed using ObjectARX 2011 which is built into AutoCAD 2011. The flow chart of the interface programme is shown in figure 7.5. The program first reads the XML file, changes the data format to the format of the AutoCAD program, generates the specification data into a surface texture indication block, inserts the indication block onto the engineering drawing with a certain angle, position and scale according to the users selection. The indication block is saved in the database of AutoCAD. The interface for reading the specification is shown in figure 7.6. When designers are dealing with similar requirements for the same or different surfaces, the saved indication can be accessed and inserted again as shown in figure 7.7.

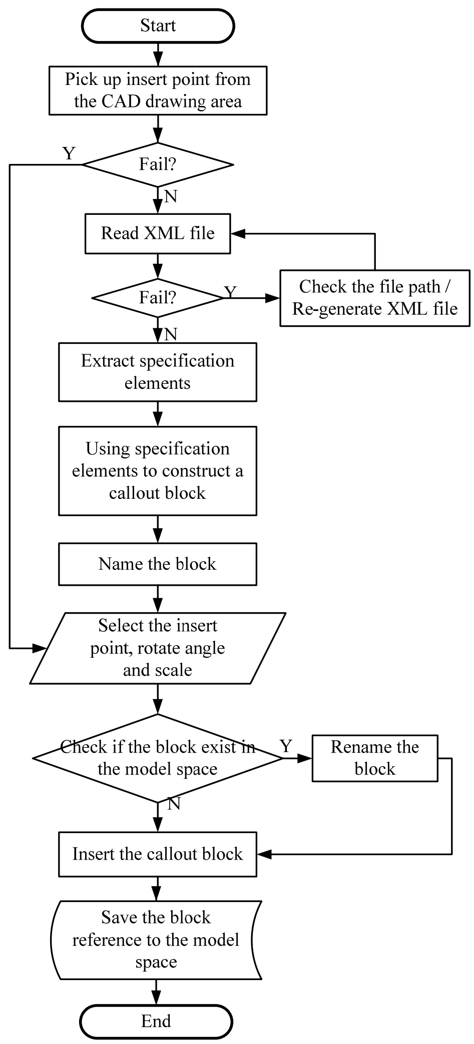


Figure 7.5 The flow chart of the integration process

A part of the code for reading XML file data is as follows:

```
…
CComPtr<MSXML::IXMLDOMDocument> spDoc;
HRESULT hr = spDoc.CoCreateInstance(__uuidof(MSXML::DOMDocument)); //Create document
object
VARIANT_BOOL bFlag;
hr = spDoc->load(CComVariant(csFileName), &bFlag);                  //Load the xml file
CComPtr<MSXML::IXMLDOMElement> spElement;
```

```
hr = spDoc->get_documentElement(&spElement);                    //Get root node
CComBSTR strTagName;
hr = spElement->get_tagName(&strTagName);
CComPtr<MSXML::IXMLDOMNodeList> spNodeList;
hr = spElement->get_childNodes(&spNodeList);                    //Get child node list
long lCount;
hr = spNodeList->get_length(&lCount);
for (long i=0; i<lCount; ++i)
        {
                CComVariant varNodeValue;
                CComPtr<MSXML::IXMLDOMNode> spNode;
                MSXML::DOMNodeType NodeType;
                CComPtr<MSXML::IXMLDOMNodeList> spChildNodeList;
                hr = spNodeList->get_item(i, &spNode);          //Get node
                hr = spNode->get_nodeType(&NodeType);           //Get node type
                if (NODE_ELEMENT == NodeType)
                {
                        hr = spNode->get_childNodes(&spChildNodeList);
                        long childLen;
                        hr = spChildNodeList->get_length(&childLen);
                        for (int j=0; j<childLen; ++j)
                        {
                                CComPtr<MSXML::IXMLDOMNode> spChildNode;
…
```

A part of the code for generating and saving of the indication block is as follows:

```
        …
        AcGePoint3d minPt1,minPt2,maxPt2,maxPt1,minPt3,maxPt3;
        …
        double len1 = maxPt1.x-minPt1.x;
        …
        pt4[X]=basePoint[X]+6.33+len;
  pt4[Y]=basePoint[Y]+7.5;
        AcDbLine *pLine4 =new AcDbLine (asPnt3d(pt3),asPnt3d(pt4));

  AcDbBlockTable *pBlockTable ;
        Acad::ErrorStatus es ;
        AcDbBlockTableRecord *pBlockTableRecord =new AcDbBlockTableRecord;
        …
  //Append the block reference to the model space
    //Block table record

  AcDbObjectId newEntId;
  pBlockTableRecord->appendAcDbEntity(newEntId, pBlkRef);
  pBlockTableRecord->close();

  AcDbBlockTableRecord *pBlockDef;
  acdbOpenObject(pBlockDef, blockId, AcDb::kForRead);
  AcDbBlockTableRecordIterator *pIterator;
  pBlockDef->newIterator(pIterator);
  AcDbEntity *pEnt;

  for (pIterator->start(); !pIterator->done();
     pIterator->step())
  {
     // Get the next entity.

     pIterator->getEntity(pEnt, AcDb::kForRead);
     pEnt->close(); // use pEnt... pAttdef might be NULL
  }
  delete pIterator;
  pBlockDef->close();
     pBlkRef->close();
```

…

The insertion program of the indication block：

```
…
m_ctrlBlcokName.GetWindowText(blockname);
AcDbBlockTable *pBlockTable ;
acdbHostApplicationServices()->workingDatabase()->getBlockTable (pBlockTable, AcDb::kForRead);

if ( pBlockTable->has (blockname) == Adesk::kTrue || blockname == "")
{
        AfxMessageBox(_T("Please reselect the name of the block!"));
        pBlockTable->close ();
        return;
}
pBlockTable->close ();
CAcUiDialog::OnOK();
addBlock();
…
```

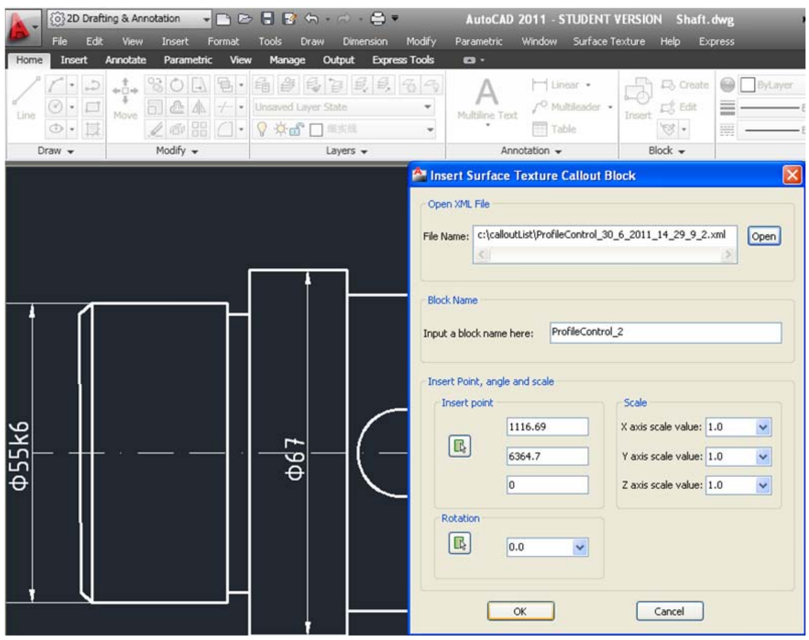


Figure 7.6 The interface for surface texture specification indication block insertion

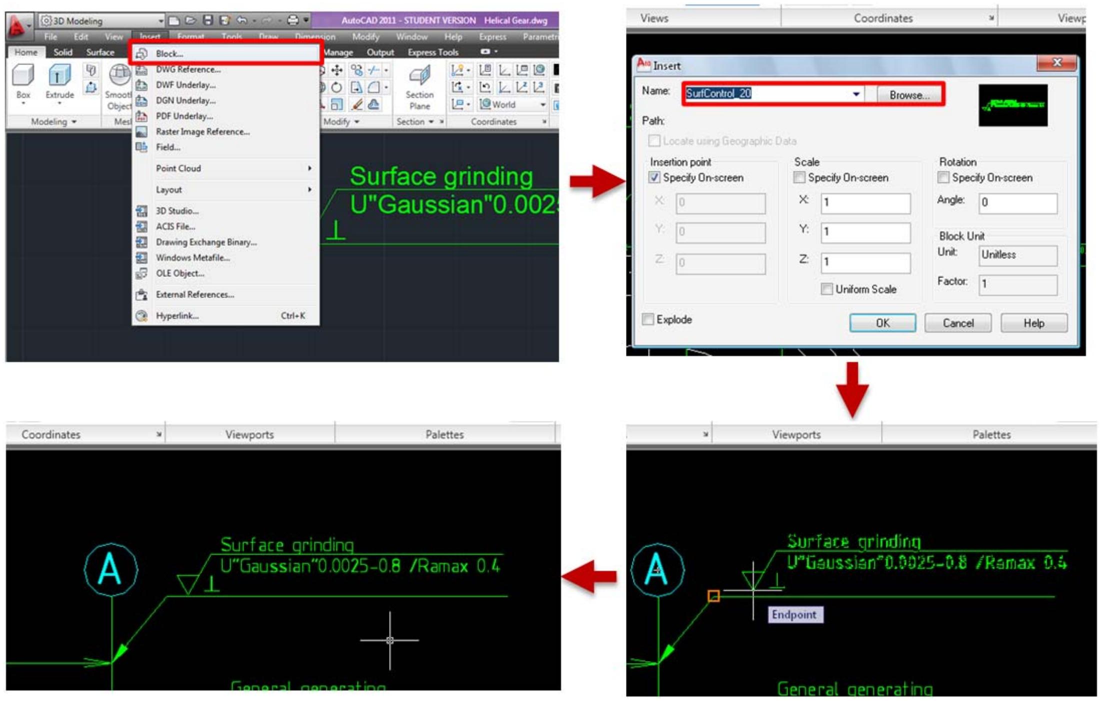


Figure 7.7 The generated specification indication block and saving in AutoCAD block database

7.3.3 Programming the interface for SolidWorks

A 'Surface Texture Addin' with two sections has been developed and is similar to the integration of AutoCAD. The menus for the two sections are shown in figure 7.8. The menu 'Surface Texture Design' is the interface that connects to the CatSurf system. Menu 'Insert Block' is the interface to open the saved XML file and generate the indication block.

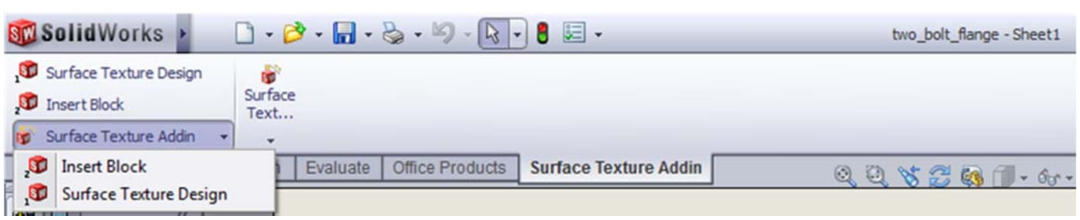


Figure 7.8 The Addin menus in SolidWorks 2009

Using Visual C# to build Addin in SolidWorks, following is part of the programming code:

```
#region UI Methods
public void AddCommandMgr()
{
   ICommandGroup cmdGroup;
```

```
BitmapHandler iBmp = new BitmapHandler();
Assembly thisAssembly;
int cmdIndex0,cmdIndex1;
string Title = "Surface Texture Addin", ToolTip = "Surface Texture Addin";
int[] docTypes = new int[]{(int)swDocumentTypes_e.swDocASSEMBLY,
                (int)swDocumentTypes_e.swDocDRAWING,
                (int)swDocumentTypes_e.swDocPART};
thisAssembly = System.Reflection.Assembly.GetAssembly(this.GetType());
cmdGroup = iCmdMgr.CreateCommandGroup(1, Title, ToolTip, "", -1);
cmdGroup.LargeIconList =
iBmp.CreateFileFromResourceBitmap("SurfaceTextureAddin.ToolbarLarge.bmp", thisAssembly);
cmdGroup.SmallIconList =
iBmp.CreateFileFromResourceBitmap("SurfaceTextureAddin.ToolbarSmall.bmp", thisAssembly);
cmdGroup.LargeMainIcon =
iBmp.CreateFileFromResourceBitmap("SurfaceTextureAddin.MainIconLarge.bmp", thisAssembly);
cmdGroup.SmallMainIcon =
iBmp.CreateFileFromResourceBitmap("SurfaceTextureAddin.MainIconSmall.bmp", thisAssembly);

cmdIndex1 = cmdGroup.AddCommandItem("Insert Block", -1, "Insert Surface Texture Symbols", "Insert
Block", 1, "InsertInterface", "", 1);
cmdIndex0 = cmdGroup.AddCommandItem("Surface Texture Design", -1, "Go to CATSURF system",
"Surface Texture Design", 0, "StartCatSurf", "", 0);
cmdGroup.HasToolbar = true;
cmdGroup.HasMenu = true;
cmdGroup.Activate();
bool bResult;
foreach (int type in docTypes)
{
  ICommandTab cmdTab;
  cmdTab = iCmdMgr.GetCommandTab(type, Title);
  if (cmdTab == null)
  {
    cmdTab = (ICommandTab)iCmdMgr.AddCommandTab(type, Title);
    CommandTabBox cmdBox = cmdTab.AddCommandTabBox();
    int[] cmdIDs = new int;
    int[] TextType = new int;
    cmdIDs[0] = cmdGroup.get_CommandID(cmdIndex0);
    System.Diagnostics.Debug.Print(cmdGroup.get_CommandID(cmdIndex0).ToString());
    TextType[0] = (int)swCommandTabButtonTextDisplay_e.swCommandTabButton_TextHorizontal;
    cmdIDs[1]= cmdGroup.get_CommandID(cmdIndex1);
    System.Diagnostics.Debug.Print(cmdGroup.get_CommandID(cmdIndex1).ToString());
    TextType[1]= (int)swCommandTabButtonTextDisplay_e.swCommandTabButton_TextHorizontal;
    cmdIDs = cmdGroup.ToolbarId;
    System.Diagnostics.Debug.Print(cmdIDs.ToString());
    TextType = (int)swCommandTabButtonTextDisplay_e.swCommandTabButton_TextHorizontal |
(int)swCommandTabButtonFlyoutStyle_e.swCommandTabButton_ActionFlyout;
    bResult = cmdBox.AddCommands(cmdIDs, TextType);
    CommandTabBox cmdBox1 = cmdTab.AddCommandTabBox();
...
```

The interface of the second part is shown in figure 7.9.

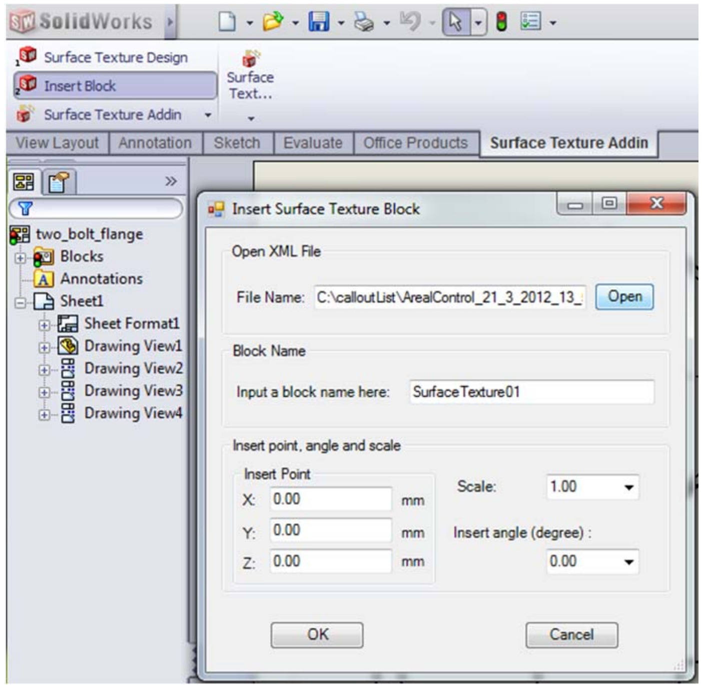


Figure 7.9 The interface for surface texture specification block insertion

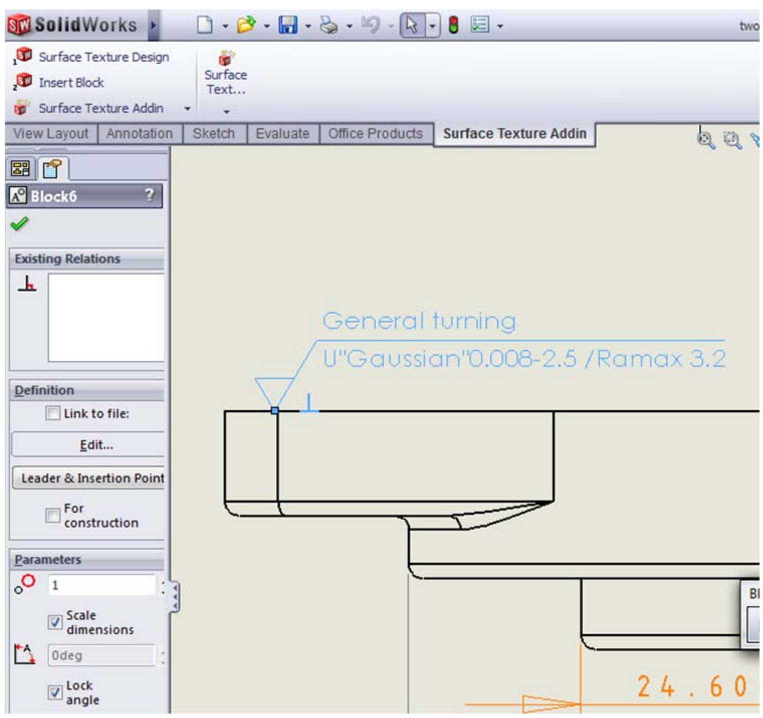


Figure 7.10 The indication block in SolidWorks 2009

Figure 7.10 shows the generated indication block in engineering drawing. Parts of the generation indication block is shown as follows:

```
…
        surfaceForm.ShowDialog();
        double[] basePoint = new double[];
        basePoint[]= Convert.ToDouble(surfaceForm.InsertPointX.Text)/1000;
        basePoint[]= Convert.ToDouble(surfaceForm.InsertPointY.Text)/1000;
        basePoint[]= Convert.ToDouble(surfaceForm.InsertPointZ.Text)/1000;
        string blockName;
        blockName = surfaceForm.BlockNmaetextBox.Text;

        ModelDoc2 swModel = default(ModelDoc2);
        DrawingDoc swDraw;
        SketchSegment[] swSkSeg = new SketchSegment[8];
        Note[] swSkNote = new Note[4];
        Object vSkSeg;
        Object vSkNote;

        SketchBlockDefinition swSketchBlockDef;
        SketchManager swSketchMgr;
        ModelDocExtension swModelDocExt;
        MathUtility swMathUtil;

        double[] nPt = new double;

        long nbrSelObjects;

        iSwApp                                                                         =
(ISldWorks)System.Runtime.InteropServices.Marshal.GetActiveObject("SldWorks.Application");
        swModel = (ModelDoc2)iSwApp.ActiveDoc;
        swDraw = (DrawingDoc)swModel;

        //Make a copy of the open drawing
        //Use the path and name of your drawing
        string CopyName = "C:\\Samples\\Copy.SLDDRW";
        swModel.SaveAsSilent(CopyName, true);

       //Interfaces needed for block APIs
       swSketchMgr = swModel.SketchManager;
       swModelDocExt = swModel.Extension;
       swMathUtil = (MathUtility)iSwApp.IGetMathUtility();
…
```

7.4 Validation of the CatSurf and interface programs

This section aims to validate the robustness and functionality of the CatSurf and interface programs by providing two case studies of surface texture specification design in AutoCAD and SolidWorks respectively. The first test case is the design of the PST specifications in AutoCAD for a helical gear. The second test case is design of the AST specifications in SolidWorks for a stepped shaft.

7.4.1 PST specifications design for a helical gear in AutoCAD

The first case study aims to assign PST specifications for a helical gear which is shown in figure 7.11.

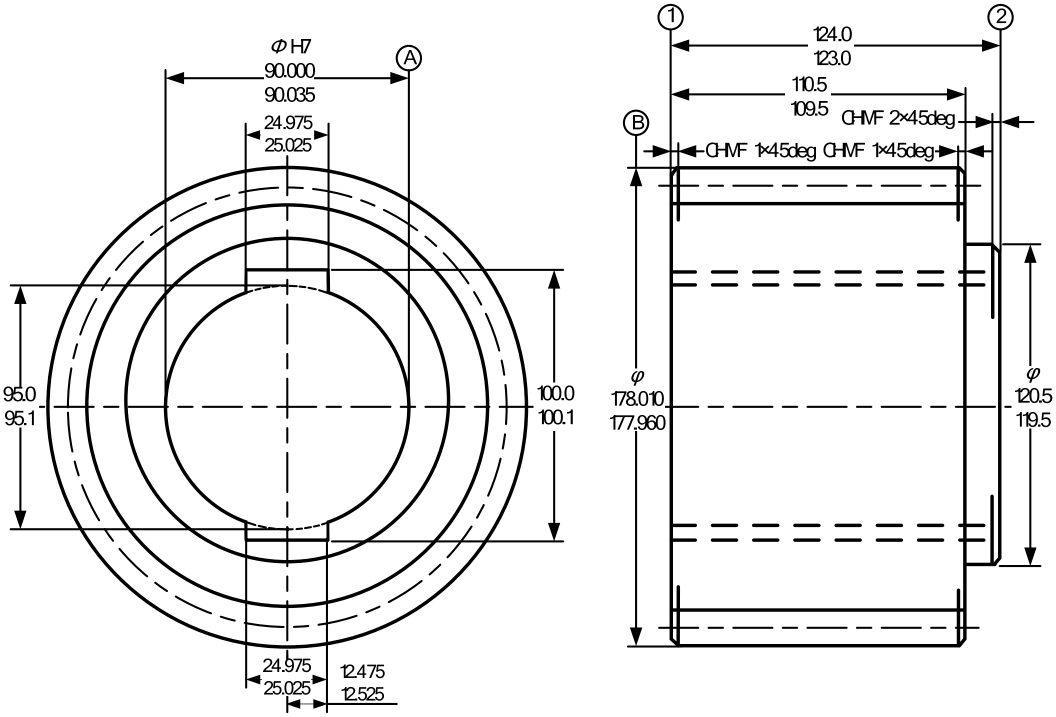


Figure 7.11 The design of a helical gear

The case study is held in the *SurfControl* module and AutoCAD 2011. There are three steps in CatSurf to assign a specification.

Step 1: In the Function component, select the correct functional surface type and material. As shown in figure 7.12, the selected functional surface is 'Spur and helical' for 'Gear teeth'; and the selected material is 'Steel Titanium and Heat Resisting Materials'.

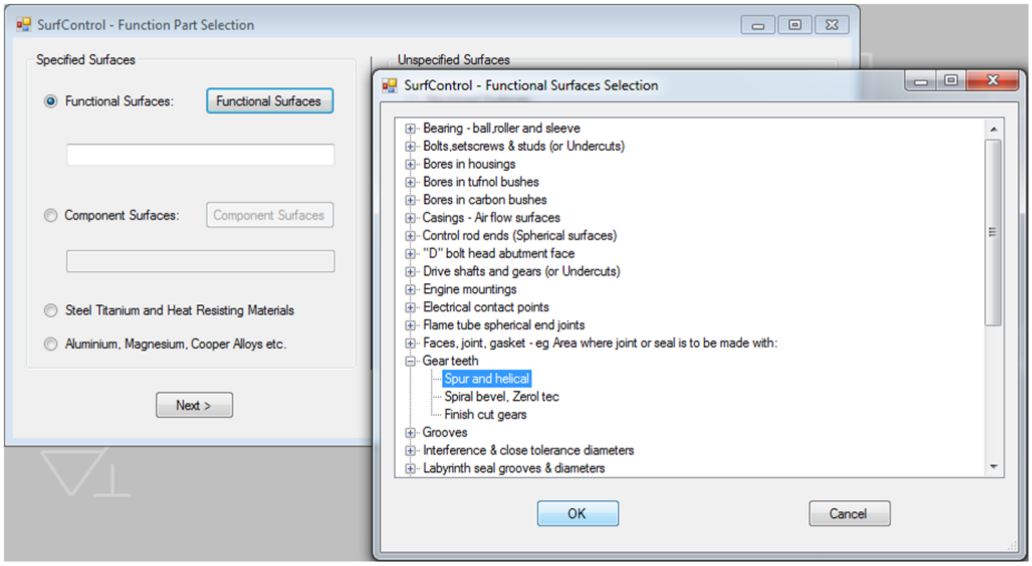


Figure 7.12 The selection of function requirement in the Function component

Step 2: In the Manufacture component, the manufacturing process of 'Surface grinding' is selected automatically as the default manufacturing process for helical gear teeth. Accordingly, the related *Ra* value range is 0.1-0.8μm and lay are '=', '⊥' and 'R'. The lay '⊥' is selected.

Step 3: In the Specification component, the details of the specification are generated automatically. The indication and XML file are saved and the XML file is named 'SurfControl_2_5_2012_11_50_59_41.xml' as shown in figure 7.13.

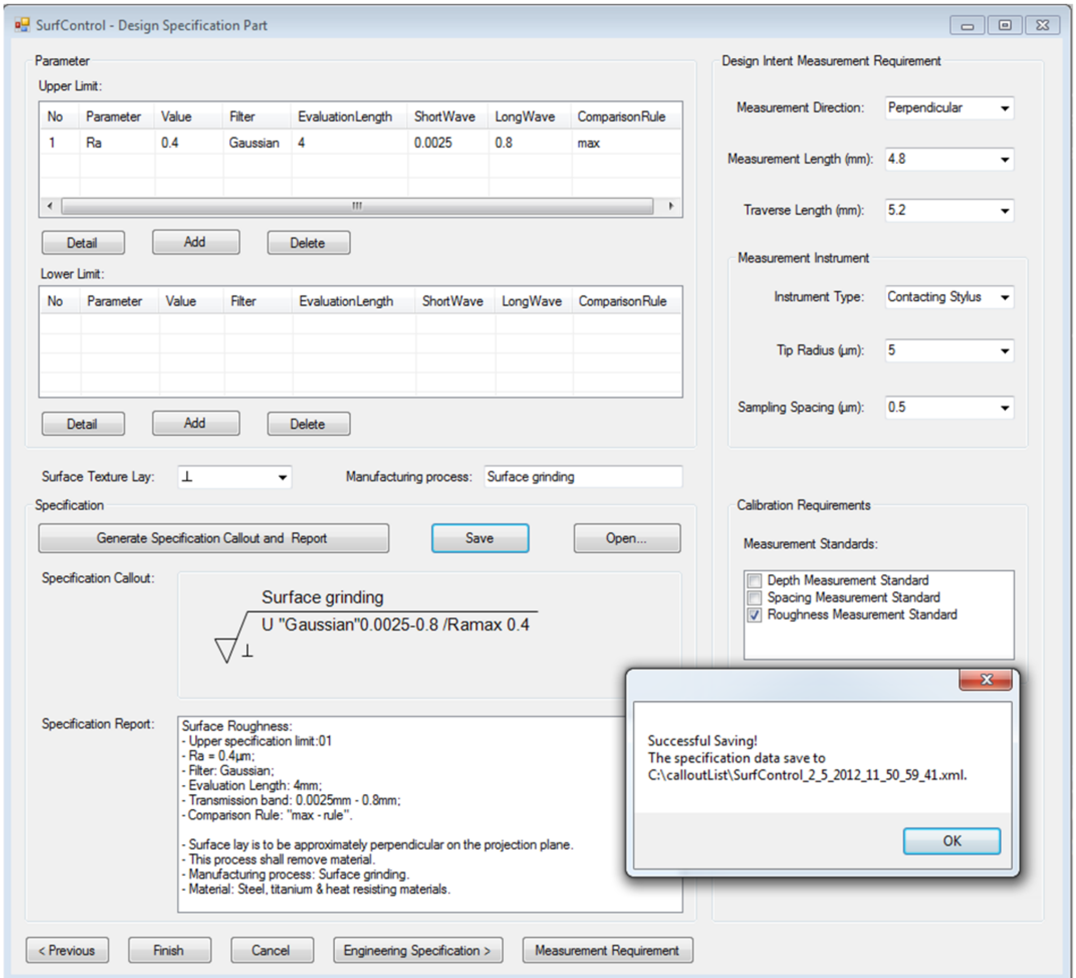


Figure 7.13 The generation of the specification in the Specification component

Returning to the AutoCAD 2011 environment, there are three steps to insert the designed specification.

Step 4: Click 'Surface Texture Drawing' menu, open the 'Insert Surface Texture Callout Block' interface as shown in figure 7.14. In the interface, open the saved XML file 'SurfControl_2_5_2012_11_50_59_41.xml'.

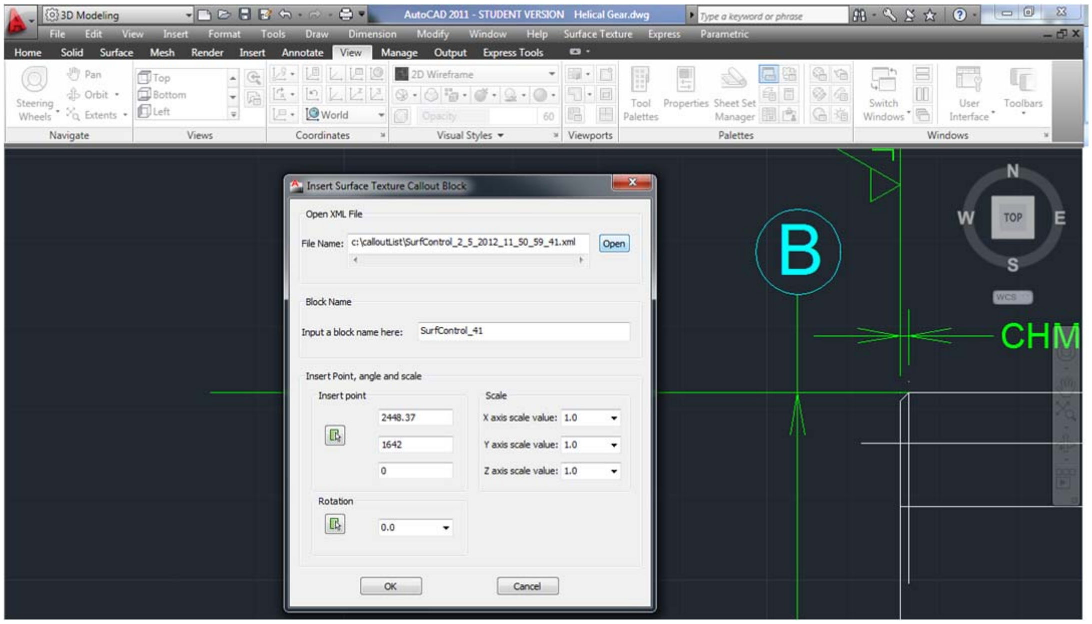


Figure 7.14 Open XML file and insert the indication block

Step 5: Change the name of the block; select the insertion point, scale and rotation. Insert the block in the drawing (as shown in figure 7.15).

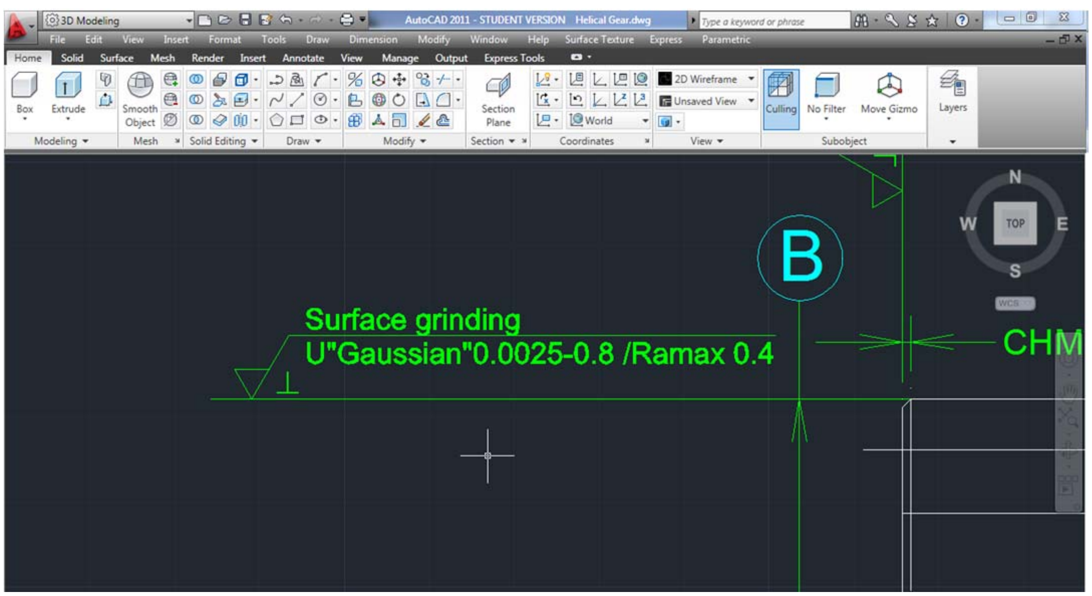


Figure 7.15 Insert the saved specification in the AutoCAD drawing

Step 6: Repeat steps 1-5 to design more specifications for a different surface in the helical gear. Alternatively it is possible to insert the saved blocks for the surfaces with the same requirements. The finished surface texture specifications are shown in figure 7.16.

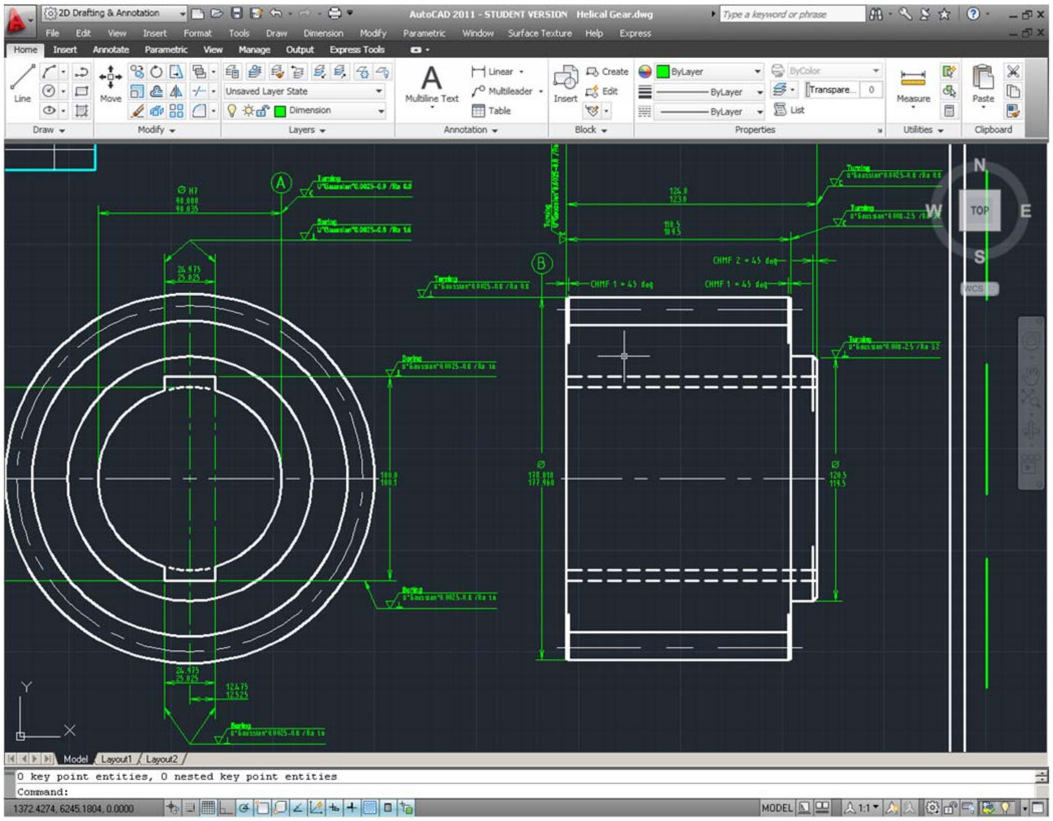


Figure 7.16 The completed PST specifications design for a helical gear

7.4.2 Areal specifications design for a stepped shaft in SolidWorks

The second case study aims to assign areal specifications for a stepped shaft which is shown in figure 7.17. According to the functional requirements, the shaft is divided into six segments.

- The shaft segment 1 of 55mm diameter is manufactured by fine turning and is an interference fit with a roller bearing.
- The shaft segment 2 of 58mm diameter with IT grade 7 is interference fitted with a helical gear.
- The shaft segment 3 of 55mm diameter is manufactured by fine turning and is an interference fit with a sleeve.
- The shaft segment 4 shares the same shaft with segment 3, and is an interference fit with a roller bearing.
- The shaft segment 5 of 55mm is manufactured by turning and is a sealing fit with an end plate.
- The segment 6 with IT grade 7 is an interference fit with a flat key.

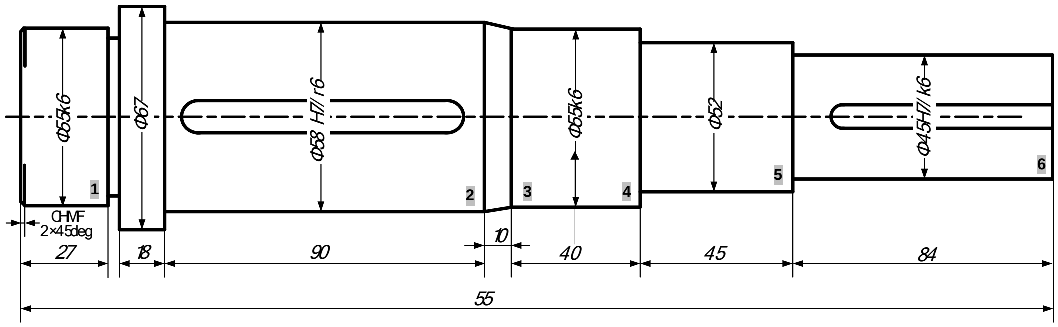


Figure 7.17 The design of a stepped shaft

By accessing the CatSurf system in SolidWorks, the *ArealControl* module is applied to carry out the specification assignment. Taking the shaft segment 1 as an example, there are three steps in the specification assignment in CatSurf.

Step 1: In the Function component, select functional surfaces 'shaft fit with rolling bearing'; Although the normal chosen parameter for turning surfaces is *Ra*, for the purpose of functionality testing, the *Sa* of 0.4µm will be chosen here as a substitute of *Ra*. Figure 7.18 shows the selection interface of Function component.

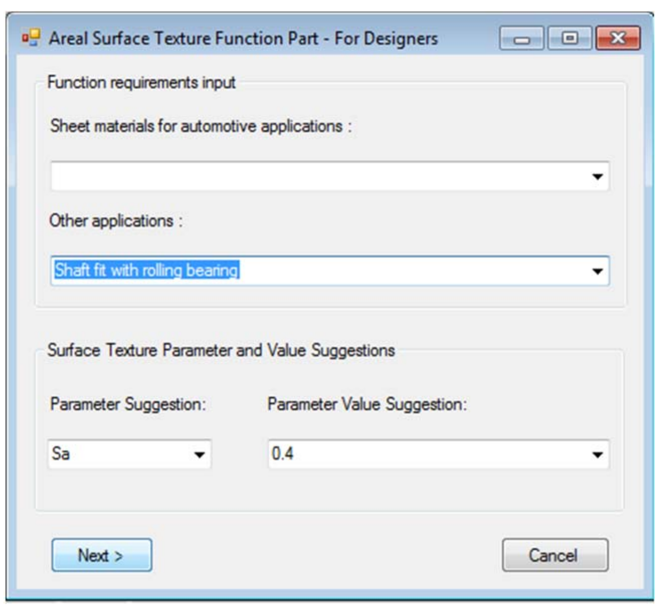


Figure 7.18 The selection of function requirements in the Function component

Step 2: In the Manufacture component, fine turning is selected (with lay '⊥').

Step 3: In the Specification component, the details of areal specification are generated automatically. The indication and XML file is saved and named 'ArealControl_3_5_2012_12_15_2_8.xml' as shown in figure 7.19.

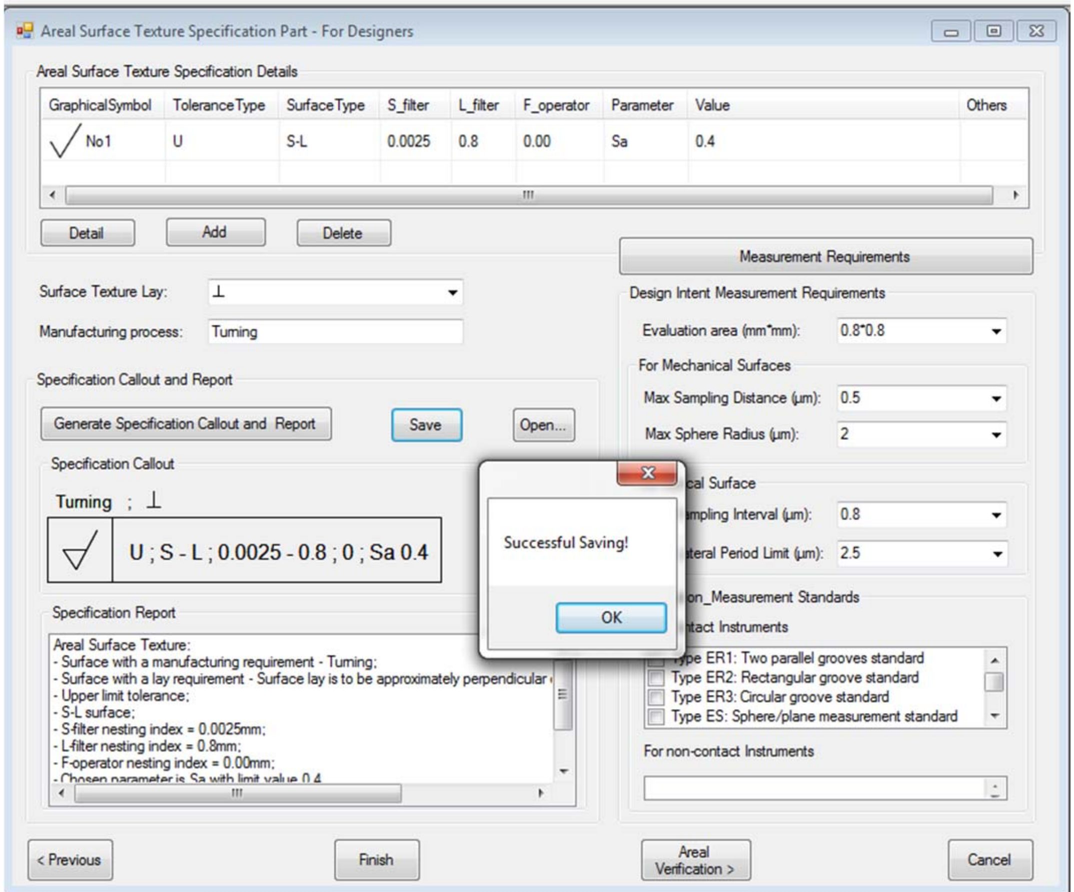


Figure 7.19 The generation of specification in Specification component

Returning to the SolidWorks 2009 environment, there are three steps to insert the saved specification in the drawing.

Step 4: Click 'Insert Block' menu, open the 'Insert Surface Texture Callout Block' interface as shown in figure 7.20. In the interface, open the saved XML file 'ArealControl_3_5_2012_12_15_2_8.xml'.

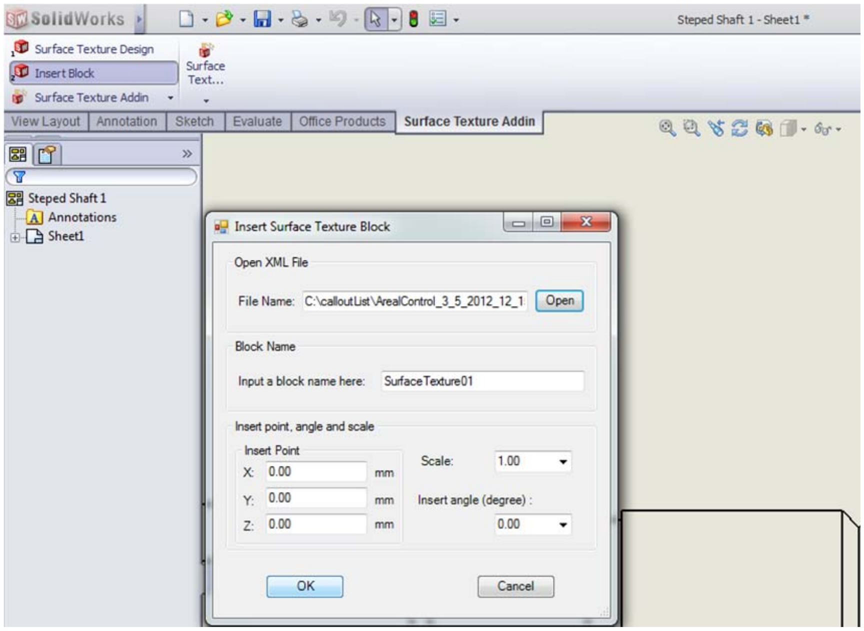


Figure 7.20 Open the saved XML file

Step 5: Change the name of the block; select insert point, scale and rotation. Insert the block in the drawing (as shown in figure 7.21).

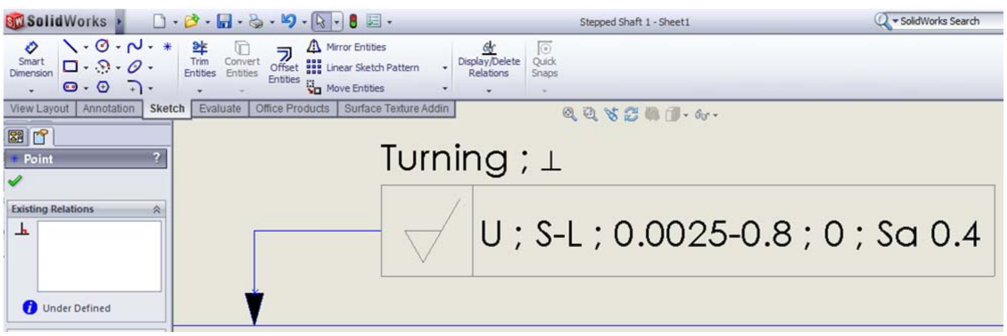


Figure 7.21 Insert the saved specification in SolidWorks

Step 6: Repeat steps 1-5 to design specifications for segment 2-6. The suggested parameter for segment 2 is *Sa* of 0.8µm, for segment 3 is *Sa* of 0.8µm, for segment 4 is *Sa* of 0.4µm, for segment 5 is *Sa* of 0.6µm and for segment 6 is *Sa* of 1.6µm. The finished surface texture specifications are shown in figure 7.22.

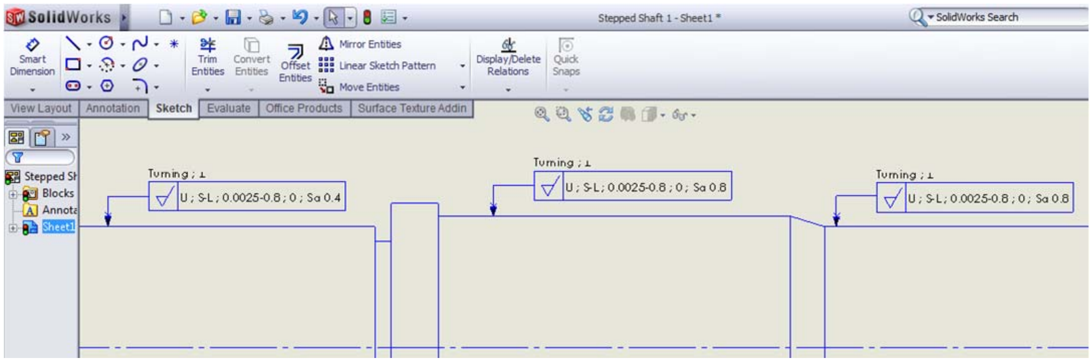


Figure 7.22 The completed AST specifications design for the stepped shaft

7.5 Conclusions

This chapter represented the integration between the CatSurf system and two different CAD systems. The XML schema based methodology is successfully carried out. Two test cases using *ProfileControl* and *ArealControl* in AutoCAD and SolidWorks were represented respectively. The integrations with other CAD systems will be introduced in future work.

8. Conclusions and Future Work

This chapter summarises the outcomes of this PhD project and highlights the contribution to knowledge in relevant research domains, by focusing on the comparison with the work that has been carried out by Wang (2008) and Xu (2009) (as mentioned in section 2.3.1). Recommendations for further work can be found in the concluding sections of this chapter.

8.1 Conclusions

The first contribution of this PhD project is the unambiguous knowledge modelling for areal and profile surface texture by utilising a more rigorous categorical model. This route includes the knowledge modelling for specification and verification of AST and PST. The knowledge model has some distinctive advantages over the other conventional data models for surface texture:

- The categorical model proposed in this project is comprehensively updated comparing with the model proposed by Wang. It redefines the families of categories, the relationships such as pullbacks and categories pullbacks, and functors, provides a more flexible, clear and easy to update model for surface texture.
- The knowledge model for AST provides foremost and latest knowledge for engineers with the underdeveloped areal standards, as similar work has not been carried out by other parties.
- The knowledge model for PST in this project is completely reconstructed comparing with the PST model proposed by Wang, provides unambiguous and complete specification and verification by utilising the updated categorical model.

The implementation method of the categorical model in the database, i.e. utilising the Db4objects C# Database, is inherited from the method proposed by Xu.

The second contribution of this project is the design and development of a new CatSurf system. This route also includes the integration methodology and implementation between CatSurf and CAD systems. This system has some distinctive advantages over the other conventional systems for surface texture:

- The first independent surface texture information system that can be integrated and which provides designers and engineers with the latest areal and profile surface texture information.
- An XML and COM based integration method which is tested in both AutoCAD and SolidWorks proves a unified integration methodology.

8.2 Future work

Detailed work in the development of CatSurf system reported in this thesis revealed more interesting issues each of which needs to be further investigated, since many of these are outside the scope of this thesis and need to be consigned to further work. These are outlined below:

1) An interesting issue that arises out of Chapter 2 is the difficulty of discovering the correlation between functional requirements and surface texture specifications. It would be desirable to incorporate more examples into the categorical database. Incorporating with more industrial users may also helpful to elaborate the function database.
2) The implementation of the categorical model discussed in Chapter 3 requires further development. It would be desirable to develop a specialised database to rigorously support category theory in the future project.
3) The knowledge model for areal surface texture developed in Chapter 5 requires continuous updating with the development of areal surface texture standards, such as updating the AST indication in accordance with the publication of ISO 25178-1 in the near future; utilisation of the new physical measurement standards defined in the ISO 25178-70; utilisation of Softgauge defined in ISO 25178-71 etc.
4) The functions for the Verification component in both profile and areal modules require further implementation. For example, a support tool for the estimation of measurement uncertainty in the Verification component is

required. The indication of areal measurement data and filtration requires further update as well.

5) Advanced industrial users need to be assigned administrational privileges for the functional database in the future. This function will provide the ability to update, create and modify the correlation cases between functional requirements and surface texture specifications, particularly for cases such as associating PST or AST parameters with specialised functional requirements, selecting a most suitable areal filter for the specified surfaces etc.
6) A simplified surface texture specification module for beginner users may be required in future work. This simplified module could be designed as a simple indication support tool which still provides simplified measurement parameters such as tip radius and traverse length, and could be integrated in the CAD systems such as AutoCAD which provides no surface texture support tool.
7) It would be desirable to develop a web-based CatSurf system for web users. The web-based system aims to provide users not only with the same information as the desktop version, but also consultative analysis and measurement results validation.

   Expected consultations may be involving explanations and implementations for GPS and national standards, such as the decision rules for proving conformity or nonconformity with specification, including the utilisation of the estimated measurement uncertainty in the Verification component.

   Case studies of analysis for the specification uncertainty of the assigned surface texture specifications from the users will also be useful to reduce the ambiguous of the specification, thus to reduce cost and any further disruption in manufacture and measurement phases.

References

Abbott, E. J., & Fireston, F. A. (1933). Specifying surface quality: a method based on accurate measurement and comparison. *Mechanical Engineering, 55*, 569-572.

ASME. (1996). *ASME Y14.36M-1996 Surface Texture Symbols*.

ASME. (2002). *ASME B46.1-2002 Surface Texture (Surface Roughness, Waviness, and Lay)*.

Autodesk. http://usa.autodesk.com/adsk/servlet/index?id=773204&siteID=123112.

Autodesk. http://www.autodesk.com/.

Awodey, S. (2006). *Category Theory*: Clarendon Press, Oxford.

Barr, M., & Wells, C. (1995). *Category theory for computing science*: Prentice Hall International.

Bennich, P. (2003). *Geometrical measurements*. Paper presented at the MetroTrade Workshop - Traceability and Measurement Uncertainty in Testing, Berlin.

Bennich, P., & Nielsen, H. S. (2005). An Overview of GPS, A Cost Saving Tool. http://www.ifgps.com/.

Blunt, L., Bills, P., Jiang, X., Hardaker, C., & Chakrabarty, G. (2009). The role of tribology and metrology in the latest development of bio-materials. *Wear, 266*(3-4), 424-431.

Blunt, L., & Jiang, X. (2003). *Advanced techniques for assessment surface topography*: Kogan Page Science.

Blunt, L., Jiang, X., & Stout, K. J. (1999). Developments in 3D surface metrology. *Laser metrology and machine performance IV*, 255-266.

British Standard. (1988). *BS 1134-1: 1988 Assessment of surface texture - Part 1: Methods and instrumentation*. British Standard.

Bui, S. H., Gopalan, V., & Raja, J. (2001). An internet based surface texture information system. *International Journal of Machine Tools and Manufacture, 41*(13-14), 2171-2177.

Bui, S. H., Muralikrishnan, B., & Raja, J. (2005). A framework for Internet-based surface texture analysis and information system. *Precison Engineering, 29*, 298-306.

Bui, S. H., & Vorburger, T. V. (2007). Surface metrology algorithm texting system. *Precison Engineering, 31*, 218-225.

Christiansen, S. & De Chiffre, L. (1997). Topographic Characterization of Progressive Wear on Deep Drawing Dies. *Tribology Transactions*, *40*:2, 346-352.

Curtis, M. A., & Farago, F. T. (2007). *Handbook of Dimensional Measurement* (4 ed.). New York: Industrial Press, Inc.

Dantan, J. Y., Vincent, J. P., Goch, G., & Mathieu, L. (2010). Correlation uncertainty--Application to gear conformity. *CIRP Annals - Manufacturing Technology, 59*(1), 509-512.

Dassault Systemes. http://www.3ds.com/products/catia/.

Dassault Systemes. http://www.solidworks.com/.

De Chiffre, L., & Nielsen, H. S. (1987). A digital system surface roughness analysis of plane and cylindrical parts. *Precision Engineering, 9*, 59-64.

De Chiffre, L., et al. (2000). Quantitative characterisation of surface texture. *CIRP Annals – Manufacturing Technology*, *49(2)*, 635-642, 644-652.

EC Software GmbH. http://www.helpandmanual.com/.

Ellerman, D. (2005). A Theory of Adjoint Functors - with some Thoughts about their Philosophical Significance. In G. Sica (Ed.), *What is Category Theory?* (pp. 127-183). Milan: Polimatrica: Polimetrica International Scientific Publisher.

Evans, H. P., Snidle, R. W., & Sharif, K. J. (2009). Deterministic mixed lubrication modelling using roughness measurements in gear applications. *Tribology International, 42*(10), 1406-1417.

Gåhlin, R., & Jacobson, S. (1998). A novel method to map and quantify wear on a micro-scale. *Wear, 222*(2), 93-102.

Hoffman, E.G., McCauley, C.J., Hussain, M.I. (2000). *Shop reference for students and apprentices (from machinery handbook)*: Industrial Press Inc., New York.

ISO 1302. (2002). *Geometrical Product Specifications (GPS) -Indication of surface texture in technical product documentation*. International Organization for Standardization.

ISO 4287. (1997). *Geometrical Product Specifications (GPS) - Surface texture: Profile method -- Terms, definitions and surface texture parameters*. International Organization for Standardization.

ISO 4288. (1996). *Geometrical Product Specifications (GPS) - Surface texture: Profile method -- Metrological characteristics of phase correct filters*. International Organization for Standardization.

ISO 5436-1. (2001). *Geometrical Product Specifications (GPS) - Surface texture: Profile method; Measurement standards - Part 1: Material measures*. International Organization for Standardization.

ISO 5436-2. (2012). *Geometrical Product Specifications (GPS) - Surface texture: Profile method; Measurement standards - Part 2: Software measurement standards*. International Organization for Standardization.

ISO 12085. (1996). *Geometrical Product Specifications (GPS) - Surface texture: Profile method -- Motif parameters*. International Organization for Standardization.

ISO 13565-1. (1996). *Geometrical Product Specifications (GPS) - Surface texture: Profile method; Surfaces having stratified functional properties - Part 1: Filtering and general measurement conditions*. International Organization for Standardization.

ISO 13565-2. (1996). *Geometrical Product Specifications (GPS) - Surface texture: Profile method; Surfaces having stratified functional properties - Part 2: Height characterization using the linear material ratio curve*. International Organization for Standardization.

ISO 13565-3. (1998). *Geometrical Product Specifications (GPS) - Surface texture: Profile method; Surfaces having stratified functional properties - Part 3: Height characterization using the material probability curve*.

ISO 14253-1. (1999). *Geometrical Product Specifications (GPS) - Inspection by measurement of workpieces and measuring equipment - Part 1: Decision rules for proving conformance or non-conformance with specifications*. International Organization for Standardization.

ISO 14253-2. (2001). *Geometrical Product Specifications (GPS) - Inspection by measurement of workpieces and measuring equipment - Part 2: Guide to the estimation of uncertainty in GPS measurement, in calibration of measuring equipment and in product verification*. International Organization for Standardization.

ISO 14253-3. (2001). *Geometrical product specifications (GPS) - Inspection by measurement of workpieces and measuring equipment - Part 3: Guidelines for achieving agreements on measurement uncertainty statements*. International Organization for Standardization.

ISO/TR 14638. (1996). *Geometrical Product Specifications (GPS) - Masterplan*. International Organization for Standardization.

ISO/DIS 16610-41. (2012). *Geometrical product specifications (GPS) - Filtration - Part 41: Morphological profile filters: Disk and horizontal line-segment filters*. International Organization for Standardization.

ISO/DIS 16610-49. (2012*Geometrical product specifications (GPS) - Filtration - Part 49: Morphological profile filters: Scale space techniques*. International Organization for Standardization.

ISO 17450-1. (2011). *Geometrical Product Specifications (GPS) - General concepts - Part 1: Model for geometrical specification and verification*. International Organization for Standardization.

ISO 17450-2. (2012). *ISO Geometrical product specifications (GPS) - General concepts - Part 2: Basic tenets, specifications, operators, uncertainties and ambiguities*. International Organization for Standardization.

ISO/CD 25178-1. (2009). *Geometrical Product Specification (GPS) - Surface Texture: Areal - Part 1: Indication of surface texture*. International Organization for Standardization.

ISO/DIS 25178-1. (2013). *Geometrical Product Specification (GPS) - Surface Texture: Areal - Part 1: Indication of surface texture*. International Organization for Standardization.

ISO 25178-3. (2012). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 3: Specification operators*. International Organization for Standardization.

ISO 25178-6. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 6: Classification of methods for measuring surface texture* International Organization for Standardization.

ISO/CD 25178-70. (2011). *Geometrical product specification (GPS) - Surface texture: Areal - Part 70: Physical measurement standards* International Organization for Standardization.

ISO/DIS 25178-71. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 71: Software measurement standards*. International Organization for Standardization.

ISO 25178-601. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 601: Nominal characteristics of contact (stylus) instruments*. International Organization for Standardization.

ISO 25178-602. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 602: Nominal characteristics of non-contact (confocal chromatic probe) instruments*. International Organization for Standardization.

ISO/DIS 25178-603. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 603: Nominal characteristics of non-contact (phase shifting interferometric microscopy) instruments*. International Organization for Standardization.

ISO/DIS 25178-604. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 604: Nominal characteristics of non-contact (coherence scanning interferometry) instruments*. International Organization for Standardization.

ISO/CD 25178-605. (2009). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 605: Nominal characteristics of non-contact (point autofocusing) instruments.* International Organization for Standardization.

ISO 25178-701. (2010). *Geometrical product specifications (GPS) - Surface texture: Areal - Part 701: Calibration and measurement standards for contact (stylus) instruments.* International Organization for Standardization.

JCGM 100. (2008). *Evaluation of measurement data - Guide to the expression of uncertainty in measurement*. International Organization for Standardization.

JCGM 101. (2008). *Evaluation of measurement data - Supplement 1 to the 'Guide to the expression of uncertainty in measurement' - Propagation of distributions using a Monte Carlo method*. International Organization for Standardization.

Jeng, Y., Lin, Z., & Shyu, S. (2002). A Microscopic Wear Measurement Method for General Surfaces. *Journal of Tribology, 124*(4), 829-833.

Jiang, X., Scott, P. J., Whitehouse, D. J., & Blunt, L. (2007a). Paradigm shifts in surface metrology. Part I. Historical philosophy. *Proceedings of the Royacl Society A*(463), 2049-2070.

Jiang, X., Scott, P. J., Whitehouse, D. J., & Blunt, L. (2007b). Paradigm shifts in surface metrology. Part II. The current shift. *Proceedings of the Royacl Society A*(463), 2071-2099.

Jung, L., Kruger-Sehm, R., Sparanger, B., & Koenders, L. (2001). Reference software for roughness analysis. *Proceedings of the EUSPEN*, S500-503.
Krupka, I., Svoboda, P., & Hartl, M. (2010). Effect of surface topography on mixed lubrication film formation during start up under rolling/sliding conditions. *Tribology International, 43*(5-6), 1035-1042.
Krystek, M. (1996). A fast algorithm for roughness measurement. *Precision Engineering, 19*, 198-200.
Krzyzak, Z., & Pawlus, P. (2006). 'Zero-wear'of piston skirt surface topography. *Wear, 260*(4-5), 554-561.
Kumar, R., Kumar, S., Prakash, B., & Sethuramiah, A. (2000). Assessment of engine liner wear from bearing area curves. [Research article]. *Wear, 239*(2), 282-286.
Lakshminarayanan, P. A., Nayak, N., & Dani, A. D. (2008). Prediction of liner wear by extending the Kragelskii model for wear of roughness peaks in sparse contact. *Part J:Journal of Engineering Tribology 216*, 327-342.
Lane, S. M. (1971). *Categories for the working mathematician*: Springer-Verlag, New York.
Leach, R. K., & Harris, P. M. (2002). Ambiguities in the definition of spacing parameters for surface-texture characterization. *Measurement Science and Technology, 13*(12), 1924-1930.
Li, T. (2011). *Softgauges for surface texture.* The degree of Doctor of Philosophy PhD thesis, The University of Huddersfield, Huddersfield.
Liu, X., Liewald, M., & Becker, D. (2009). Effects of Rolling Direction and Lubricant on Friction in Sheet Metal Forming. *Journal of Tribology, 131*, 042101-042108.
Lu, W. (2012). *Modelling the Integration between the Design and Inspection Process of Geometrical Specifications for Digital Manufacturing.* The degree of Doctor of Philosophy PhD thesis, The University of Huddersfield, Huddersfield.
Machpro. http://www.machpro.fr.
Microsoft. http://msdn.microsoft.com/en-us/vjsharp/bb188593.aspx.
Microsoft. http://www.microsoft.com/visualstudio/en-gb/products/2008-editions.
Military Standard. (1949). *MIL-STD-10A - Surface Roughness, Waviness and Lay*. Washington D.C.
Mono. http://www.mono-project.com/DB4O.
National Physical Laboratory. (2012). Metrology for the 2020s.
Nayak, P. R. (1971). Random process model of rough surface. *Tran. ASME, J. Lubric. Technol., 39*, 398-407.
Nielsen, H. S. (2006). New concepts in specifications, operators and uncertainties and their impact on measurement and instrumentation. *Measurement Science and Technology, 17*(3), 541-544.
Nielsen, H. S. (2012). *Recent developments in ISO-GPS Standards and strategic planes for future work*. Paper presented at the 12th CIRP Conference on Coumpter Aided Tolerancing, Huddersfield, UK.
Object Refinery Limited. http://www.jfree.org/jfreechart/.
Pawlus, P., & Grabon, W. (2008). The method of truncation parameters measurement from material ratio curve. *Precision Engineering, 32*(4), 342-347.
Pierce, B. C. (1991). *Basic Category Theory for Computer Scientists*: Massachusetts Institute of Technology.
PTC Inc. http://www.ptc.com/product/creo/.
Qi, Q., Jiang, X., Liu, X., & Scott, P. J. (2010). An unambiguous expression method of the surface texture. *Measurement, 43*(10), 1398-1403.
Qi, Q., Jiang, X., Scott, P.J. & Lu, W. (2013). Surface texture specification, the more complete the better? *Procedia CIRP*, *10*, 233-237.
Raja, J., & Radhakrishnan, V. (1979). Digital filtering of surface profiles. *Wear, 57*(147-155).
Raja, J., & Radhakrishnan, V. (1979). Filtering of surface profiles using fast Fourier transform. *Int J. Mach. Toll Des. Res, 19*, 133-141.

Reason, R. E. (1944a). *Surface Finish and Its Measurement*: The institution of production engineers.
Reason, R. E., Hopkins, M. R., Garrod, R. I. (1944b). *Report on the measurement of surface finish by stylus methods*: The Rank Organisation Rank Taylor Hobson Division, Leicester, England.
Rezayat, M. (2000). Knowledge-based product development using XML and KCs. *Computer-Aided Design, 32*, 299-309.
Rosén, B. G., Ohlsson, R., & Westberg, J. (1995). Interactive surface modelling, an implementation of an expert system for specification of surface roughness and topography. *International Journal of Machine Tools and Manufacture, 35*(2), 317-324.
Rossiter, B. N., Nelson, D. A., & Heather, M. A. (1994). The Categorical Product Data Model as a Formalism for Object-Relational Databases: www.cs.ncl.ac.uk/publications/trs/papers/505.pdf.
Sacerdotti, F., Porrino, A., Butler, C., Brinkmann, S., & Vermeulen, M. (2003). SCOUT - Surface Characterization Open-Source Universal Toolbox. *Meas. Sci. Technol., 13*(2), N21-N26.
Sayles, R. S., & Thomas, T. R. (1977). *Measurement of the statistical microgeometry of Engineering surfaces.* Paper presented at the Proc. 1st Joint Polytechnic Symp. manufacturing engineering, Leicester.
Schlesinger, G. (1942). *Surface Finish*. London: Institute of Production Engineers.
Schorsch, H. (1958). *Gutebestimung an technischen Oberflachen*. Stuttgart: Wissenschaftliche.
Scott, P. J. (2006). The case of surface texture parameter RSm. *Measurement Science and Technology, 17*(3), 559-564.
Scott, P. J. (2009). Feature Parameters. *Wear, 266*, 548-551.
Scott, P. J. (2013). Latest development of GPS (In private communication).
Siemens. http://www.plm.automation.siemens.com/en_us/products/nx/index.shtml.
Machpro. http://www.machpro.fr/default.html.
Stout, K. J., & Blunt, L. (1994). *Three dimensional surface topography*. London: Penton Press.
Stout, K. J., Sullivan, P. J., Dong, W. P., Mainsah, E., Luo, N., Mathis, T., & Zahyouani, H. (1993). *The development of methods for the characterization of roughness in three dimensions*: Commission of the European Communities.
Suh, A. Y., Polycarpou, A. A., & Conry, T. F. (2003). Detailed surface roughness characterization of engineering surfaces undergoing tribological testing leading to scuffing. *Wear, 255*(1-6), 556-568.
Taylor Hobson. http://www.taylor-hobson.com/.
Teague, E. C., Scire, F. E., Baker, S. M., & Jensen, S. W. (1982). Three-dimensional stylus profilometry. *Wear, 83*, 1-12.
Veeco. http://www.veeco.com/.
Walters, R. F. C. (1991). *Categories and Computer Science*: Carslaw Publications.
Wang, Y. (2008). *A Knowledge-based Intelligent System for Surface Texture (VIRTUALSURF)* The degree of Doctor of Philosophy PhD thesis, The University of Huddersfield, Huddersfield.
Whitehouse, D. J. (1967-68). Improved type of wavefilter for use in surface-finish measurement. *Proc. Inst. Mech. Eng., 182*, 306-318.
Whitehouse, D. J. (1982). The parameter rash -- is there a cure? *Wear, 83*(1), 75-78.
Whitehouse, D. J. (1994). *Handbook of Surface Metrology*: Institute of Physics Pub.
Whitehouse, D. J. (2001). Function maps and the role of surfaces. *International Journal of Machine Tools and Manufacture, 41*(13), 1847-1861.
Whitehouse, D. J. (2002). *Surface and Their Measurement*: Hermes Penton Ltd.

Whitehouse, D. J., Bowen, D. K., Venkatesh, V. C., Lonardo, P., & Brown, C. A. (1994). Gloss and Surface Topography. *CIRP Annals - Manufacturing Technology, 43*(2), 541-549.

Whitehouse, D. J., & Jiang, X. (2012). *General Metrology in Manufacture*. Paper presented at the 12th CIRP Conference on Computer Aided Tolerancing, Huddersfield, UK.

Williamson, J. P. B. (1967-1968). Microtopography of surfaces. *Proc. Inst. Mech. Eng., 182*, 21-30.

Wolf, G. W. (1991). A FORTRAN subroutine for cartographic generalization. *Computers & Geosciences, 17*(10), 1359-1381.

Xu, Y. (2009). *The exploration of a category theory-based VirtualGPS system for design and manufacturing.* The degree of Doctor of Philosophy PhD thesis, The University of Huddersfield, Huddersfield.

Yuan, Y. B. (2000). A fast algorithm for determining the Gaussian filtered mean line in surface metrology. *Precision Engineering, 24*, 62-69.

Appendix - Partial code for pullbacks *R_2*, *R_3*, *R_9* and *R_{13}* implementation in *ProfileControl*

1. Partial C# Code of Pullback *R_2* for the implementation of Function component in *ProfileControl*

```
private void suggestionButton_Click(object sender, EventArgs e)
{
   IObjectContainer db = Db4oFactory.OpenFile("ProfileControlSchema");
   try
   {
      IQuery query = db.Query();
      query.Constrain(typeof(databaseInit.ProfileInput));
      IConstraint constr = query.Descend("_partType").Constrain(partType);
      IConstraint constr1 = query.Descend("_surface").Constrain(surface);
      IConstraint constr2 = query.Descend("_it").Constrain(itComboBox.Text);
      IConstraint constr3 = query.Descend("_dimension").Constrain(dimensionComboBox.Text);
      IObjectSet Result = query.Execute();

      while (Result.HasNext())
      {
         databaseInit.ProfileInput func = (databaseInit.ProfileInput)Result.Next();
         paraTypeComboBox.Text = func.parameter();
         valueLowComboBox.Text = Convert.ToString(func.valueLow());
         valueUpComboBox.Text = Convert.ToString(func.valueUp());
      }
   }
   finally
   {
      db.Close();
   }

   if (valueLowComboBox.Text != "")
      valueLow = Convert.ToDouble(valueLowComboBox.Text);
   else
   {
      MessageBox.Show("Please re-select one type of dimension or IT!", "ProfileControl",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      return;
   }
   if(valueUpComboBox.Text != "")
      valueUp = Convert.ToDouble(valueUpComboBox.Text);
   else
   {
      MessageBox.Show("Please re-select one type of dimension or IT!", "ProfileControl",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      return;
```

```
        }
    }
```

Here, 'ProfileInput' is a class in the database for *ProfileControl*

```
  public class ProfileInput
  {
    string _partType;
    string _surface;
    int _it;
    string _dimension;
    string _parameter;
    double _valueLow;
    double _valueUp;

    public ProfileInput(string partType, string surface, int it, string dimension, string parameter,
double valueLow, double valueUp)
    {
        _partType = partType;
        _surface = surface;
        _it = it;
        _dimension = dimension;
        _parameter = parameter;
        _valueLow = valueLow;
        _valueUp = valueUp;
    }

    public string partType()
    {
        return _partType;
    }

    public string surface()
    {
        return _surface;
    }

    public int it()
    {
        return _it;
    }

    public string dimension()
    {
        return _dimension;
    }

    public string parameter()
    {
        return _parameter;
    }

    public double valueLow()
    {
        return _valueLow;
    }
```

```
        public double valueUp()
        {
            return _valueUp;
        }

    }
```

2. Partial C# Code of Pullback *R₃* for the implementation of Manufacture component in *ProfileControl*

```
        private void manufactureTreeView_AfterSelect(object sender, TreeViewEventArgs e)
        {
            this.manuListView.Items.Clear();
            manuProcess = this.manufactureTreeView.SelectedNode.Text;
            IObjectContainer db = Db4oFactory.OpenFile("ProfileControlSchema");
            if (e.Node != null)
            {
                try
                {
                    IQuery query = db.Query();
                    query.Constrain(typeof(databaseInit.ProfileManufacture));
                    query.Descend("_processName").Constrain(manuProcess);
                    IObjectSet Result = query.Execute();
                    while (Result.HasNext())
                    {
                        databaseInit.ProfileManufacture manu = (databaseInit.ProfileManufacture)Result.Next();
                        ListViewItem li = new ListViewItem();
                        li.SubItems.Clear();
                        li.SubItems[0].Text = manu.indicationType();
                        ImageList imageList = new ImageList();
                        imageList.ImageSize = new Size(40, 26);
                        if (manu.indicationType() == "Material Removal")
                            imageList.Images.Add(new    Bitmap(System.IO.Directory.GetCurrentDirectory()    +
"/image/Remove.bmp"));
                        if (manu.indicationType() == "Non-Material Removal")
                            imageList.Images.Add(new    Bitmap(System.IO.Directory.GetCurrentDirectory()    +
"/image/NonRemove.bmp"));
                        else
                            imageList.Images.Add(new    Bitmap(System.IO.Directory.GetCurrentDirectory()    +
"/image/Any.bmp"));
                        this.manuListView.SmallImageList = imageList;
                        li.SubItems.Add(manu.RaValueLow().ToString()    +    "    -    "    +
manu.RaValueUp().ToString());
                        li.SubItems.Add(manu.lay());
                        li.ImageIndex = 0;
                        this.manuListView.Items.Add(li);
                        RaUp = manu.RaValueUp();
                        RaLow = manu.RaValueLow();
                    }
                }
                finally
                {
                    db.Close();
                }
            }
```

Here, ‘ProfileManufacture’ is a class in the database for *ProfileControl*

```
public class ProfileManufacture
  {
    string _manufactureType;
    string _processName;
    double _RaValueUp;
    double _RaValueLow;
    string _lay;
    string _indicationType;
    string _layInterpretation;
    public ProfileManufacture(string manufactureType, string processName, double RaValueUp, double RaValueLow, string lay, string indicationType, string layInterpretation)
    {
      _manufactureType = manufactureType;
      _processName = processName;
      _RaValueUp = RaValueUp;
      _RaValueLow = RaValueLow;
      _lay = lay;
      _indicationType = indicationType;
      _layInterpretation = layInterpretation;
    }
    public string manufactureType()
    {
      return _manufactureType;
    }
    public string processName()
    {
      return _processName;
    }
    public double RaValueUp()
    {
      return _RaValueUp;
    }
    public double RaValueLow()
    {
      return _RaValueLow;
    }
    public string lay()
    {
      return _lay;
    }
    public string indicationType()
    {
      return _indicationType;
    }
    public string layInterpretation()
    {
      return _layInterpretation;
    }
  }
```

3. Partial C# Code of Pullback R_9 and R_{13} for the implementation of Specification component in *ProfileControl*

```
…
            if (this.lowerListView.Items.Count > 0)
            {
              IQuery query2 = db.Query();
              query2.Constrain(typeof(databaseInit.MeasureParameter));
              IQuery constr3 = query2.Descend("_RaValueUp");

constr3.Constrain(Convert.ToDouble(this.lowerListView.Items[0].SubItems[2].Text)).Greater().Equal(
);
              IQuery constr4 = query2.Descend("_RaValueLow");

constr4.Constrain(Convert.ToDouble(this.lowerListView.Items[0].SubItems[2].Text)).Smaller();
              IObjectSet meaLowResult = query2.Execute();
              while (meaLowResult.HasNext())
              {
                databaseInit.MeasureParameter meaLowPara =
(databaseInit.MeasureParameter)meaLowResult.Next();
                samplingSpacingLow = meaLowPara.samplingSpacing();
              }
            }
            IObjectSet meaResult = query1.Execute();
            while (meaResult.HasNext())
            {
              databaseInit.MeasureParameter                    meaPara                    =
(databaseInit.MeasureParameter)meaResult.Next();
              Rtip = meaPara.tipRadius();
              travelLength = meaPara.stylusTravel();
              samplingSpacingUp = meaPara.samplingSpacing();
            }
          }
          finally
          {
            db.Close();
          }
…
```

Here, 'MeasureParameter' is a class in the database for *ProfileControl*

```
    public class MeasureParameter
    {
      double _RaValueUp;
      double _RaValueLow;
      int _tipRadius;
      double _sampleLength;
      double _shortWave;
      double _evaluLength;
      double _stylusTravel;
      double _samplingSpacing;
      public MeasureParameter(double RaValueUp, double RaValueLow, int tipRadius, double
sampleLength, double shortWave, double evaluLength, double stylusTravel, double samplingSpacing)
      {
        _RaValueUp = RaValueUp;
```

```
        _RaValueLow = RaValueLow;
        _tipRadius = tipRadius;
        _sampleLength = sampleLength;
        _shortWave = shortWave;
        _evaluLength = evaluLength;
        _stylusTravel = stylusTravel;
        _samplingSpacing = samplingSpacing;
    }
    public double RaValueUp()
    {
        return _RaValueUp;
    }
    public double RaValueLow()
    {
        return _RaValueLow;
    }
    public int tipRadius()
    {
        return _tipRadius;
    }
    public double sampleLength()
    {
        return _sampleLength;
    }
    public double shortWave()
    {
        return _shortWave;
    }
    public double evaluLength()
    {
        return _evaluLength;
    }
    public double stylusTravel()
    {
        return _stylusTravel;
    }
    public double samplingSpacing()
    {
        return _samplingSpacing;
    }
}
```